Derivative free optimization

Prof. Cesar de Prada
ISA –UVA
prada@autom.uva.es

Outline

- Derivative free methods
- Evolutionary algorithms
  - Genetic algorithms
  - Simulated Annealing
  - Tabu Search

$$\left.\begin{array}{l}\min\limits_{x} \mathrm{J}(\mathbf{x}) \\ \mathbf{m} \leq \mathbf{x} \leq \mathbf{M}\end{array}\right\}$$

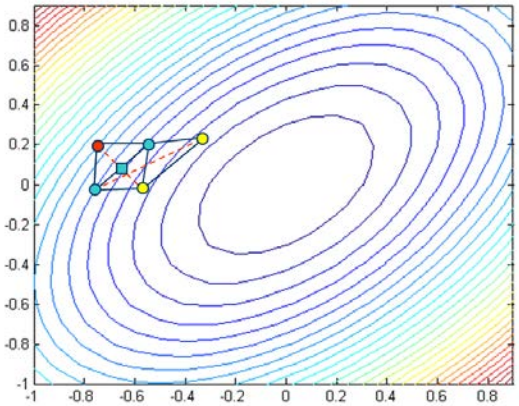
Other constraints are added as penalty functions

Derivative-free optimization

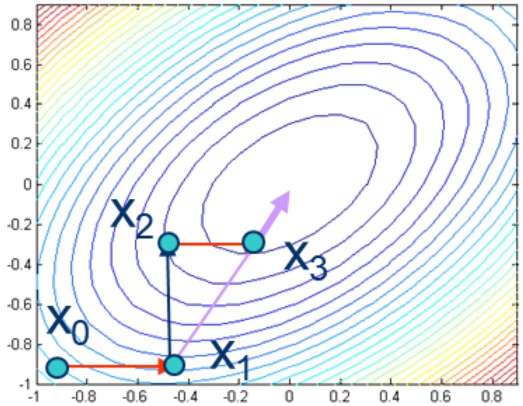
- Sometimes, the derivatives of J are difficult, costly or impossible to obtain.
- Methods oriented to solve this type of problems are known as “derivative-free”
- They can be classified as:
  - Direct methods: They use on values of J to compute the search direction
  - Model based methods: They build a model of J(x) using only values of J, and then they use this subrrogate model to compute the search direction.
  - Local : They use local information around the current iterate
  - Global: They explore simultaneously the full feasible space to iterate or use approximations of J(x) over the full feasible space.
  - Evolutionary or deterministic: If they use in the space exploration random or deterministic values

Direct local methods

- Nelder-Mead , simplex



- Powell’s conjugate directions



Trust-region methods

- An approximate model is built around the current iterate x_k which can be a sensible approximation in a region of radious r (Trust region). For instance a quadratic model:

$$J(x) \approx J_a(x) = J(x_k) + \nabla_x J(\mathbf{x_k})\Delta\mathbf{x} + \frac{1}{2}\Delta\mathbf{x}' H_k \Delta\mathbf{x}$$

- But instead of building the approximation computing the derivatives, the model coefficients are estimated by interpolation using a set of points x_{ki} inside the trust region and their $J(x_{ki})$ values.

- The minimum of $J_a(x)$ in the region of radious r, x_k^*, is computed as well as the ratio $(J(x_k) – J(x_k^*))/((J_a(x_k) – J_a(x_k^*))$. According to its value, the region is enlarged or reduced and the test points are changed in order to maintain a good shape and facilitate the estimation of $J_a(x)$, iterating with the new point until $r < \varepsilon$

Response surface methods (RSM)

- These methods approximate globally the function J(x) by a mathematically simpler surface $J_a(x)$, and use the subrogate model instead of J(x) for the optimization. Typically, $J_a(x)$ is a linear combination of known polynomials $P_i(x)$ and radial base functions $\varphi(.)$ (linear, cubic, splines, Kriging,…)

$$J(x) \approx J_a(x,p) = \sum_i \alpha_i P_i(x) + \sum_i \beta_i \varphi(x - x_i)$$

- Parameters α_i y β_i are estimated by least squares using a set of sampling points over the whole feasible domain and their corresponding J values.
- $J_a(x)$ is used in the optimization over the domain facilitating the computation of derivatives. As such, RSM are global methods.

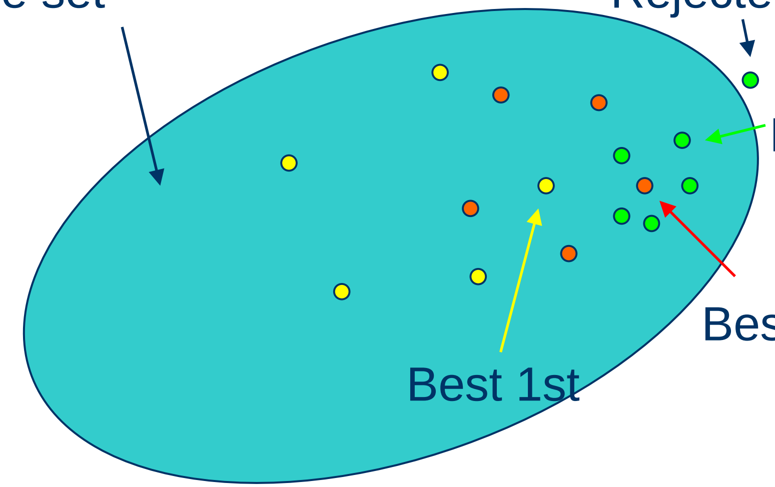
Evolutionary methods

- The optimal solution advances as the result of more or less random decisions inspired by nature.
- Some methods are oriented to deal with populations. Other look for search directions inspired by physical phenomena.
- The population methods work simultaneously with several individuals (candidate solutions) that are selected over several stages as a result of some kind of selection rule.
- In each stage (generation) the population keeps the best individuals and generate new ones with other characteristics in order to avoid local minimums.
- In this sense, they claim to be global methods.
- The selection of the individuals that continues in a new stage, as well as the generation of new individuals is made using several methods most of them of random nature.
- Several families of methods: GA, TS, SA,…

Evolutionary methods

- Usually, they employ values of the cost function, but not of its gradient, which facilitate its use in non smooth problems.
- Using these methods, it is difficult to deal with constraints and quite often these are formulated as penalty functions.
- They can be slower than the deterministic methods but have a higher chance of providing the global optimum.
- They may not be adequate if the number of decision variables is large

Montecarlo’s method



Feasible set

Rejected

Best 3rd

Best 2nd

Best 1st

● First generation

● Second generation

● Third generation

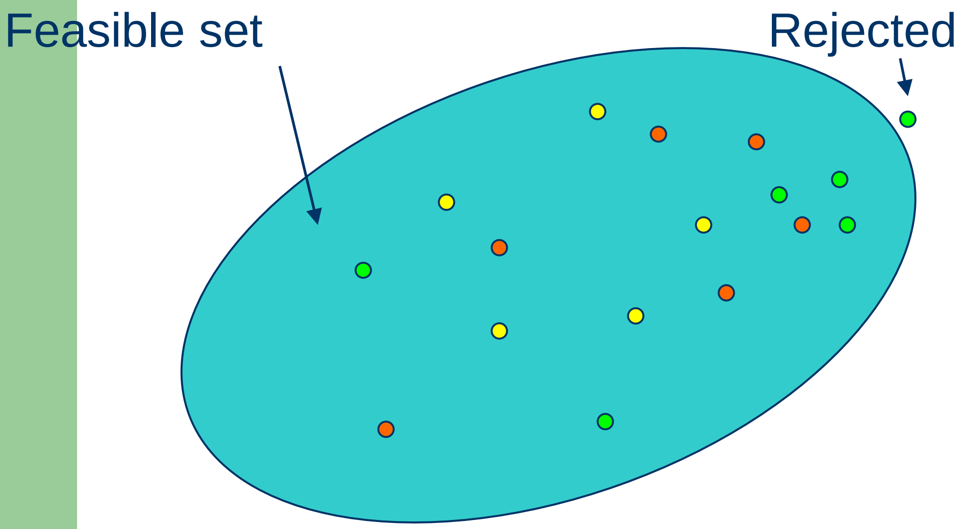
Local
optima

1 Generate randomly N candidates x_i and evaluate $J(x_i)$

2 Choose the best of them. Generate new candidates in a smaller set around the best candidate of the previous generation.

3 Keep on repeating the procedure until an ending criterion is met

Evolutionary algorithms



🟡 First generation

🟠 Second generaton

🟢 Third generation

Tend to avoid local optima

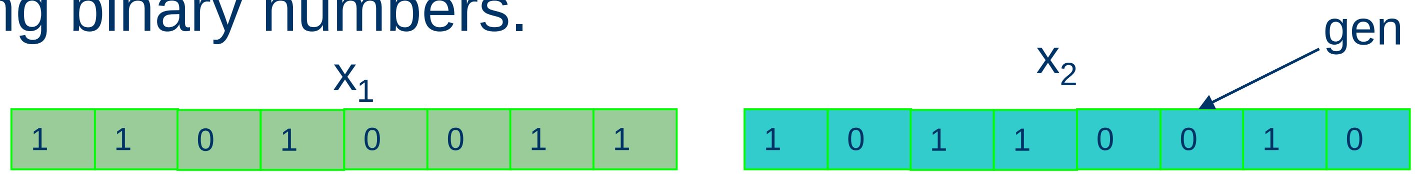
1 Generate randomly N candidates x_i and evaluate $J(x_i)$

2 Apply a selection rule. Generate new candidates so that some of them are far away from the best ones.

3 Keep on repeating the procedure until an ending criterion is met

Genetic algorithms (GA), Holland 1975

- The individuals (possible solutions x_i) are coded using binary numbers.



x_1

1	1	0	1	0	0	1	1

x_2

1	0	1	1	0	0	1	0

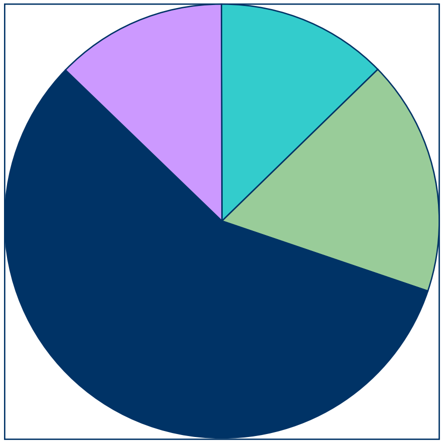
gen

- Each individual is assigned a value related to the cost function (Fitness cost) (and penalty if required)
- The generation of new individuals is based on the same principles as the genetic reproduction
- Mechanisms of selection, crossing and mutations

Selection

Those individuals that remain for the next generation are selected using a random procedure with probabilities depending on the value of the cost function for each individual

$$P(x_i) = \frac{J(x_i)}{\sum_{i=1}^{N} J(x_i)}$$

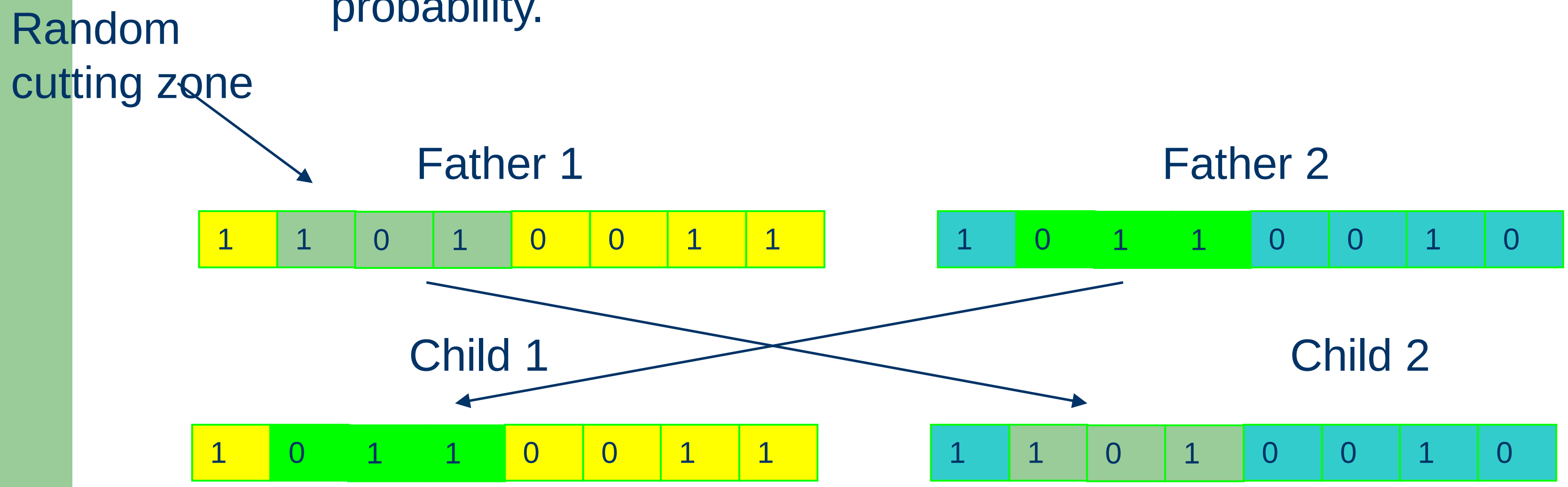


E.g. Dart game with areas of each individual proportional to the value of the cost function

It is common practice to keep also the best individual if it was not selected

Crossing

From the selected individuals, others are generated crossing them genetically at points chosen at random. Two individuals of the population are selected to be crossed with a certain probability.



Random cutting zone

Father 1

1	1	0	1	0	0	1	1

Father 2

1	0	1	1	0	0	1	0

Child 1

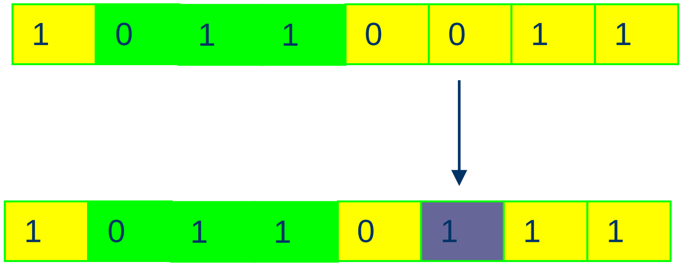
1	0	1	1	0	0	1	1

Child 2

1	1	0	1	0	0	1	0

Mutations

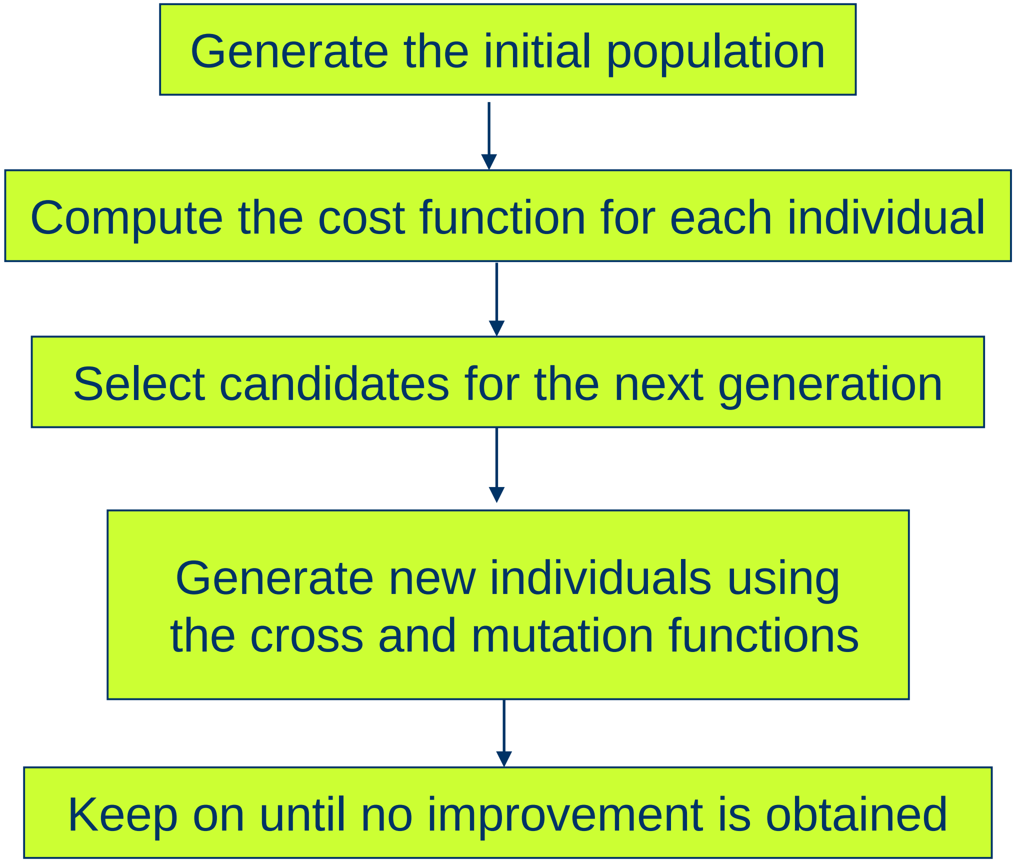
In each son, a gen is changed randomly from 0 to 1 or viceversa with a likelihood inverse to the number of genes of the individual



1 0 1 1 0 0 1 1

1 0 1 1 0 1 1 1

Population based evolutionary methods



Generate the initial population

Compute the cost function for each individual

Select candidates for the next generation

Generate new individuals using the cross and mutation functions

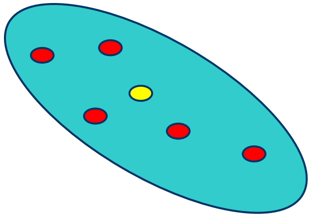
Keep on until no improvement is obtained

Simulated Annealing (SA)

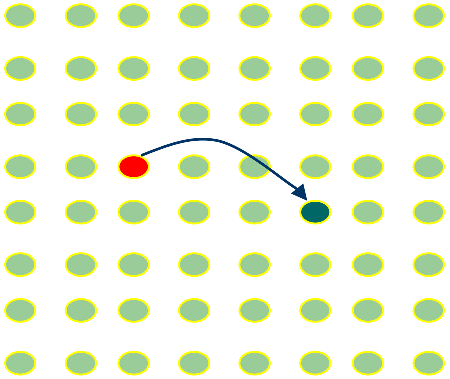
- Unlike the genetic algorithms, the simulated annealing method does not use a population of individuals but a single one that evolves according to rules that imitates the annealing process, a heat treatment used in metallurgy. In this process, first a material is heated and then cooled slowly in a controlled way, evolving to a state of lower energy, to increase the size of its crystals and improve its properties.
- A neighbourhood of the current solution point is considered as the set of points where the solution could evolve from the current point.

Simulated Annealing

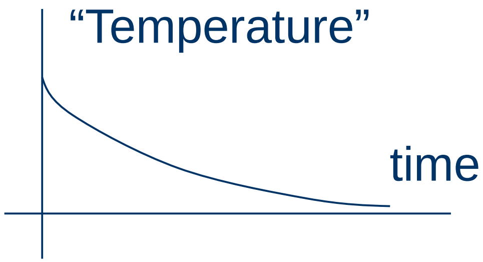
- Starting with an initial guess, the next point is chosen at random among its neighbours according to a certain probabilistic rule.
- An "energy" function E is assigned to a point and the probability of moving from x_i to x_j is defined as a function of the difference of energies $E(x_i) – E(x_j)$. E can be the cost function J
- The probability is always positive, even is the new point has more energy that the previous one, avoiding local minima
- The probability depends also on a global parameter called temperature, which decreases over time, decreasing when T goes to zero and allowing then only transitions to lower energy points.



Simulated Annealing

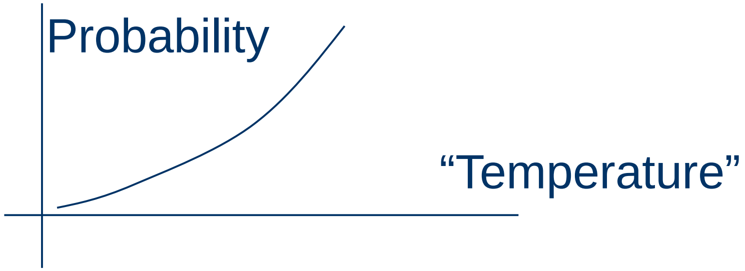


The temperature decreases with time



The probability of moving from one point x_i to another x_j depends on its difference of “energies” and on the parameter “temperature”

Prob([$J(x_i) – J(x_j)$], T) > 0 even if the new point is worst, in order to avoid local minima



Tabu search (TS)

- In order to improve the efficiency of the exploration process, one needs to keep track not only of local information (like the current value of the objective function) but also of some information related to the exploration process. This systematic use of *memory* is an essential feature of tabu search (TS).
- Such information will be used to guide the move from x_i to the next solution x_j to be chosen in the neighbourhood of x_i (*N(i))* The role of the memory will be to restrict the choice to some subset of *N(i)* by forbidding for instance moves to some neighbour solutions.
- TS can be characterized as a *metaheuristic*. Its role will most often be to guide and to orient the search of another (more local) search procedure.