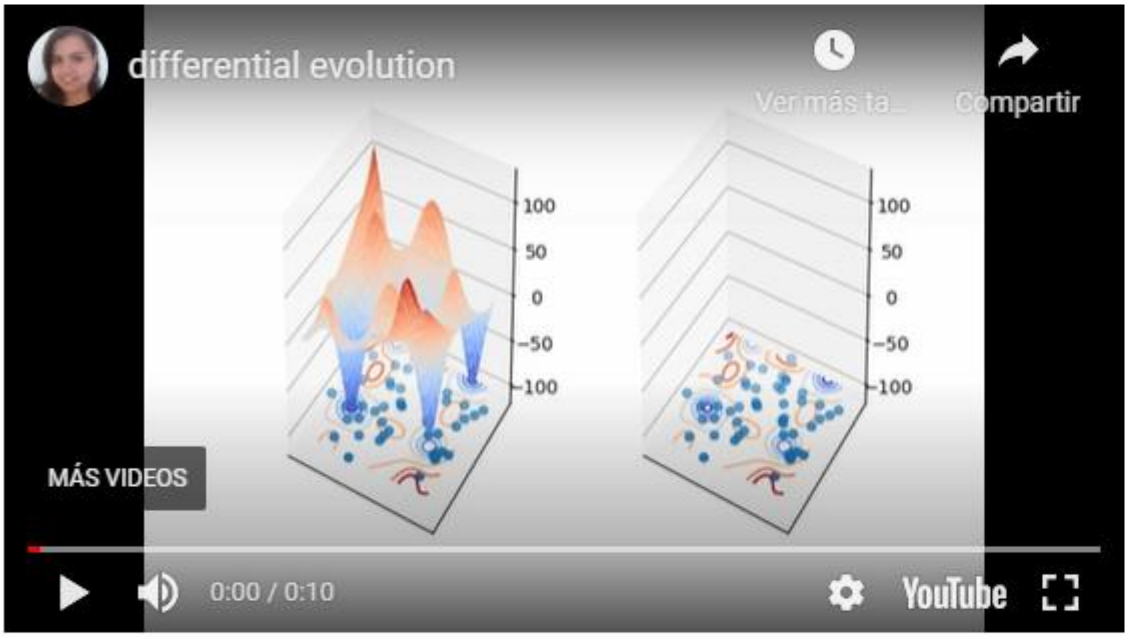
Diferencial Evolution (DE)

Differential Evolution was proposed by Storn and Price in 1995. It is a robust algorithm for solving continuous multidimensional optimization problems. In this algorithm, individuals as seen as vectors. The novelty is the mutation operator, that uses three individuals for mutate another one, and the mutation depends on the distance

It is possible to see that Evolution Differential's individuals converge to the local minimums, and eventually, all converge to the global minimum. In the next video you can see the main idea:



https://youtu.be/BsfJDg0a0Z4

Representation

The individuals are represented as vectors whose entries are the variables values. In this case, the *i-th* individual is represented with the vector $x^i \in \mathbb{R}^d$.

$$x^i =< x_1^i, x_2^i, \dots, x_d^i >$$

The first generation of individuals must be initialized using a range of values. Each variable has a minimum x_k^{min} and a maximum value x_k^{max}. Formally, each variable *k* of *i-th* individual can be calculated as follows:

$$x_k^i = x_k^{min} + r(x_k^{max} - x_k^{min})$$

Where *r* is a random number from 0 to 1.

Mutation

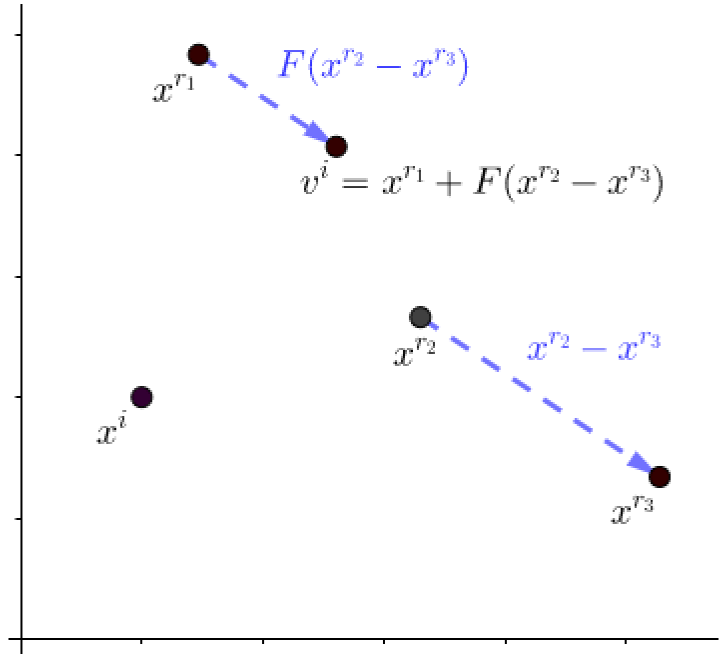
Mutation is the main operation in Differential Evolution. In this operator, for each individual x^i, another one called v^i is generated. The first step consists of randomly selecting three individuals from the population: x^{r1}, x^{r2}, x^{r3}. The new individual v^i is calculated as follows:

$$v^i = x^{r1} + F(x^{r2} - x^{r3})$$

Where F is a random number between 0 and 2.

The difference between x^{r2} and x^{r3} defines the direction and the magnitude of mutation. F slightly changes the magnitude. x^{r1} Represents the initial point.

At the beginning, all the individuals are dispersed and $x^{r2} - x^{r3}$ is big, but when the algorithm is converging, the individuals are concentrated in some local minimums and the value of $x^{r2} - x^{r3}$ is smaller. It is magic!



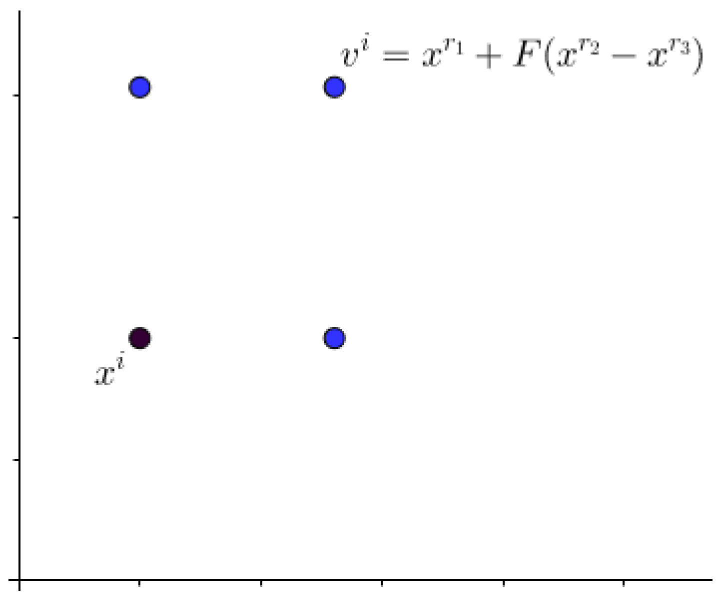
Crossover

Crossover's objective is to combine the original vector x^i with the new one v^i for creating another one u^i.

$$u_k^i = \begin{cases} v_k^i & If\ rand(0,1) \leq Cr\ or\ k = l \\ x_k^i & otherwise \end{cases}$$

For each variable *k* of u^i, the value is selected randomly between v^i or u^i. If a random number between 0 and 1 is less than Cr, the value is taken from v^i. For guarantying that at less one value is taken from v^i, the value of a variable randomly selected *l*, is assigned with the value of v^i.

The new vector u^i is positioned in one of the corners of the hyperrectangle generated with the positions of x^i and v^i.



Selection

The selection is performed by tournament. The best individual of u^i and x^i is selected to be part of the next generation.

Differential Evolution Algorithm

Parameters:
    N, population size
    G, Maximum number of generations
    Cr, crossover probability
 	
Return: the best individual
 	
Begin
  Create the initial population of N individuals
  Calculate the population fitness
  While the number of generations is less than G or we haven't found a good solution
    For each individual x^i in the population
      Mutation: Create a new individual (v^i)
      Crossover: combine the individuals $x^i, v^i \longrightarrow u^i$
      Calculate the fitness of u^i
      Selection: select the best individual between $x^i = best(x^i, u^i)$
    End for
  End while
End
Return the best individual