Evolution Strategies (ES)

Evolution Strategies were proposed by Rechenberg y Schwefel (Technical University of Berlin) in 1960's. The goal is to solve continuous multidimensional optimization problems. The main characteristic is the **self-adaptation** of parameters. It means that some evolutive algorithm parameters change during the execution. Those parameters are included in the individual representation and evolve at the same time that the solution.

Individuals' representation

Given the goal is solving continuous multidimensional optimization problems, the individuals' solutions are represented as vectors whose inputs are the values of the variables, the individual i's solution is represented as the vector $\vec{x_i} \in \mathbb{R}^d$, where d represents the number of features. One of the main characteristics of Evolution Strategies is self-adaptation of parameters, where each individual contains the mutation parameters σ_i, in addition to the values of the variables that are stored in the vector $\vec{x_i}$. The individual's representation is as follows:

$$< \vec{x_i}, \sigma_i >$$

Mutation

The individuals can be seen as points in a d –multidimensional space, where the mutation's goal is to move those points so that the position of the mutated individual is close to the position of the individual before mutation.

The individual's position $\vec{x_i}$ is modified by adding a random number, noise, to each entry. The noise follows a normal distribution zero-centered and with standard deviation σ_i. The value σ_i is known as *mutation step size*, where the bigger the value, the bigger the individual modification.

Rechenberg proposed the famous **1/5 success rule**, it states that the ratio of successful mutations (those in which the child is fitter than the parent) to all mutations should be 1/5. Hence if the ratio is greater than $1/5$ the step size should be increased to make a wider search of the space, and if the ratio is less than $1/5$ then it should be decreased to concentrate the search more around the current solution. The rule is executed at periodic intervals, for instance, after k iterations, each σ is reset by

$$\sigma = \begin{cases} \sigma/c & si\ p > 1/5 \quad\quad Increase \\ \sigma * c & si\ p < 1/5 \quad\quad Decrease \\ \sigma & si\ p = 1/5 \end{cases}$$

Where p is the relative frequency of successful mutations measured over a number of trials, and the parameter c is in the range $0.817 \leq c \leq 1$.

We describe two special cases of mutation in evolution strategies:

- **Uncorrelated Mutation with One Step Size**. In this case, the individual is represented by its position $\vec{x}$ and one scalar that represents the mutation step size σ.

$$\underbrace{< x_1, x_2, \ldots, x_d,}_{\vec{x}} \underbrace{\sigma >}_{\sigma}$$

In the mutation, we add to all the vector entries a random number obtained from a normal distribution zero-centered with standard deviation equals to standard σ. The mutation step size must also be modified for self-adaptation. The mutation is performed as follows:

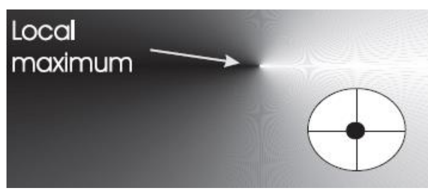
$$\sigma' = \sigma * e^{N(0,\tau)}$$
$$x_i' = x_i + N(0,\sigma)$$

the recommended value for τ is $\tau = 1/\sqrt{n}$. For avoiding step sizes equals to 0, it is recommended

$$If\ \sigma' < \epsilon \ \Rightarrow \sigma' = \epsilon$$

where ϵ is an small scalar, it is recommended $\epsilon = 1e-3$.

The mutation can be seen as modification in a hypersphere, where the modifications near to the original position have more probability than the ones far from the original position. This mutation is recommended for problems whose features have the same values range.



En este caso, la mutación de un individuo se puede ver como una nueva posición en una hiperesfera cuyo centro es la posición actual del individuo, además de que posiciones cercanas al individuo son más probables (por la distribución Gaussiana).

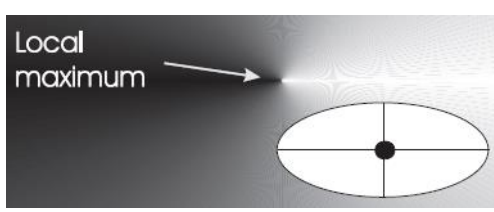
- **Uncorrelated Mutation with d Step Sizes**. The main idea is to treat each feature independently. It is useful where the features have different ranges values. In this case, the mutation parameter is a vector $\vec{\sigma} \in \mathbb{R}^d$. The individuals' representation is:

$$\underbrace{< x_1, x_2, \ldots, x_d,}_{\vec{x}} \underbrace{\sigma_1, \sigma_2, \ldots, \sigma_d >}_{\vec{\sigma}}$$

The mutation is performed as follows:

$$\sigma_i' = \sigma_i * e^{N(0,\tau_1)+N(0,\tau_2)}$$
$$x_i' = x_i + N(0,\sigma_i')$$

where $\tau_1 = 1/\sqrt{2n}$ y $\tau_2 = 1/\sqrt{2\sqrt{n}}$. It is important to avoid values of σ_i near to 0.



Recombination

The recombination consists of create one child from two parents. In Evolution Strategies there are two recombination variants. In *intermediate recombination* the values of the parents are averaged. Using *discrete recombination* one of the parent's values is randomly chosen with equal chance for either parents. The parents can be represented as the vectors $\overrightarrow{p_1}$ and $\overrightarrow{p_2}$, and the child is the resultant vector $\vec{c}$. Using this notation, the recombination techniques are defined as:

$$c_i = \begin{cases} (p_{1i} + p_{2i})/2 & intermediate\ recombination \\ random\ selection\colon p_{1i}\ or\ p_{2i} & discrete\ recombination \end{cases}$$

Selection of Parents

Parent selection in ES is completely random, it is because here the whole population is seen as parent.

Survivor Selection

After creating λ offspring and calculating their fitness, the best μ of them are chosen deterministically. There are two schemes of survivor selection:

- (μ, λ) selection, where only the best μ offspring are selected
- $(\mu + \lambda)$ selection, where the best μ individuals (from the union of parents and offspring) are selected

To sum up, the book Introduction to Evolutionary Algorithms (Eiben) presents the following summary of ES:

- Representation: real-valued vectors
- Recombination: discrete or intermediary
- Mutation: Gaussian perturbation
- Parent selection: Uniform random
- Survivor selection: (μ, λ) or $(\mu + \lambda)$
- Specialty: self-adaptation of mutation step sizes

Evolution Strategies Algorithm

Parameters:
    N, population size
    λ, offspring size
    G, Maximum number of generations
 	
Return: the elite individual
 	
Begin
  Create the initial population
  Calculate the population fitness
  Get the elite
  While the number of generations is less than G or we haven't found a good solution
      Recombination, generate λ offspring
      Mutation of parents and offspring
      Calculate the population fitness
      Survivor selection (the best individuals)
      Get the elite or include the elite in the population
  End while
End