# Evolutionary Computation, 2017/2018
# Programming assignment 4

## Important information

- Deadline: 06/Nov/2017, 23:59.
- Mooshak: none. Implement and test it on your own.
- Send PDF file and a ZIP file with your code (C, C++, Java, or Python) by email to `fernando.lobo@gmail.com`

## About this assignment

The purpose of this programming assignment is to have you program simple evolution strategies. You should report the results obtained on a PDF document and send it to `fernando.lobo@gmail.com` by the deadline. Please attach a ZIP file with all of your code and be sure to include a `readme.txt` file with directions on how to compile and execute the various problems.

## Problem A: generating pseudo-random numbers from the Standard Normal Distribution

Write a function that generates pseudo-random numbers from the Standard Normal Distribution, from now on referred as N(0,1). One way to do that is to implement the *Marsaglia polar method* (see http://en.wikipedia.org/wiki/Marsaglia_polar_method).

## Problem B: (1+1)-ES

Implement a (1+1)-ES for real-valued vectors. When doing mutation, each variable is modified by adding the outcome of a sample from N(0,1). Test your algorithm on the so-called *Sphere Model*, an easy unimodal test function defined as follows for $n$-dimensional vectors.

$$f(\bar{x}) = \sum_{i=0}^{n-1} x_i^2$$

with each $x_i$ initially generated in [-100.0,+100.0]. The optimum value of this function is $x_i = 0.0$ for all $i$. Test it for different values of $n$, say $n = 10$ and $n = 100$. How close can you get to the optimum? And how fast?

# Problem C: (1+1)-ES with the 1/5 rule

Do the same as problem B but implement the 1/5 rule for adapting the mutation strength. Start with an initial $\sigma = 1$. What differences do you observe comparing with problem B?

# Problem D: ($\mu$+$\lambda$)-ES and ($\mu$,$\lambda$)-ES with one step size (one sigma)

Implement the ($\mu$+$\lambda$)-ES and ($\mu$,$\lambda$)-ES with one uncorrelated mutation. For adapting the mutation rate, use the recommendation suggested in class (by Hans-Paul Schwefel) and set $\tau = 1/\sqrt{n}$. Test the algorithms in the Sphere Model as well as on Ackley's function defined as follows:

$$f(\bar{x}) = -20 \cdot \exp\left(-0.2 \cdot \sqrt{\frac{1}{n} \cdot \sum_{i=0}^{n-1} x_i^2}\right) - \exp\left(\frac{1}{n} \cdot \sum_{i=0}^{n-1} \cos(2\pi x_i)\right) + 20 + \exp(1)$$

with each $x_i$ initially generated in [-30.0,+30.0]. The optimum value of this function is also $x_i = 0.0$ for all $i$. Test it for different values of $n$, say $n = 10$ and $n = 100$, and for different values of $\mu$ and $\lambda$.

# Problem E: ($\mu$+$\lambda$)-ES and ($\mu$,$\lambda$)-ES with $n$ step size ($n$ sigmas)

Same as problem D but this time your ES has $n$ sigmas, one for each variable. Set the global learning rate to $1/\sqrt{2 \cdot n}$, and the local learning rate to $1/\sqrt{2 \cdot \sqrt{n}}$