

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE CIENCIAS



Tarea 06:
Estrategias evolutivas

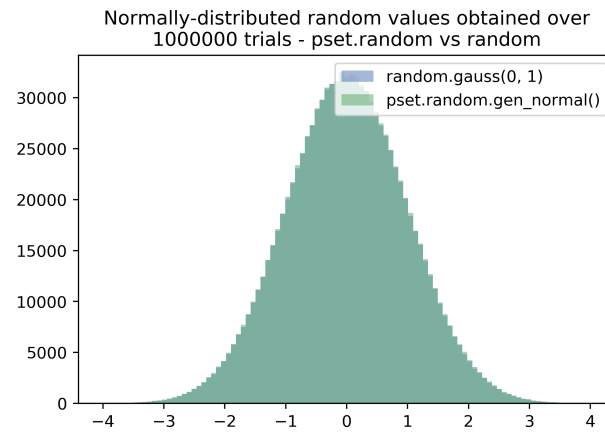
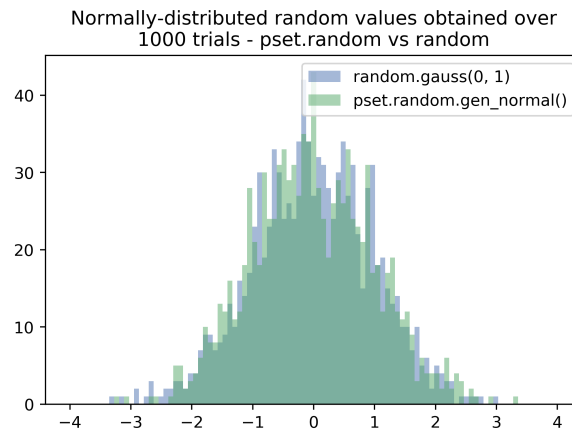
Pablo A. Trinidad Paz - 419004279

Trabajo presentado como parte del curso de **Cómputo Evolutivo** impartido por el profesor **Mario Iván Jaen Márquez**.

Fecha de entrega: **Jueves 4 de Abril de 2019**.

1. **[Ejercicio de programación]** Escribe una función que genere números pseudo-aleatorios de las distribución normal estándar $N(0, 1)$ a partir de números uniformemente distribuidos. Indica el método usado.

Solución: Se implementó el método de muestreo de números pseudo-aleatorios descrito por Box-Muller¹. A continuación se presentan los resultados de la implementación comparados con el método `random.gauss(0, 1)` de la librería estándar de Python.



¹https://en.wikipedia.org/wiki/Box-Muller_transform

2. **[Ejercicio de programación]** Implementa el algoritmo (1+1)-ES. Prueba tu algoritmo sobre la función *Sphere*, la cuál es una función unimodal d -dimensional definida como:

$$f(\vec{x}) = \sum_{i=1}^d x_i^2$$

Donde cada $x_i \in [-100, 100]$. Utiliza un parámetro $\sigma = 1$ y un punto inicial $\vec{x} = (x_1, \dots, x_d) = (-99, \dots, -99)$

- a) Ejecuta tu algoritmo para $d = 10$ y para $d = 100$. ¿Qué tan cerca del óptimo converge y qué tan rápido?

Respuesta: Para $d = 10$ con una precisión de 0.01, el algoritmo converge después 391 mutaciones exitosas con un valor promedio para cada x_i de 0.00263. Para $d = 100$ con la misma precisión, el algoritmo converge después de 1337 mutaciones con un valor promedio para cada x_i de 0.1486.

- b) Implementa la regla del 1/5 y vuelve a ejecutar tu algoritmo. ¿Qué diferencias observas respecto a la ejecución anterior?

Respuesta: La diferencia sólo se notó en el número de mutaciones exitosas siendo reducido a 374 con $d = 10$ y un x_i promedio de -0.1249 y para $d = 100$ un número de mutaciones exitosas 1211 con un x_i promedio de -0.03162 .

```
Starting simulation with d=10 (fifth rule disabled)
Objective reached
  Generations: 100000
  Successful mutations: 391
  Chromosome mean: 0.0026387421236935328
  Fitness: 0.4259425692724451
Starting simulation with d=100 (fifth rule disabled)
Objective reached
  Generations: 100000
  Successful mutations: 1337
  Chromosome mean: 0.14864745560856008
  Fitness: 158.61422411438807
Starting simulation with d=10 (fifth rule enabled)
Objective reached
  Generations: 100000
  Successful mutations: 374
  Chromosome mean: -0.12495015578777005
  Fitness: 0.28857368094209557
Starting simulation with d=100 (fifth rule enabled)
Objective reached
  Generations: 100000
  Successful mutations: 1211
  Chromosome mean: -0.03162669553738553
  Fitness: 91.53402843247675
```