

Computación Bioinspirada

Dr. Edward Hinojosa Cárdenas
ehinojosa@unsa.edu.pe

Programación Evolutiva

- Por razones históricas, la PE ha sido asociada durante mucho tiempo a tareas de predicción y al uso de MEF como su representación.
- A partir de los '90, las variantes de PE para optimización de vectores de parámetros reales se han vuelto más frecuentes
 - Estas variantes se han posicionado como variantes estándar de PE

Programación Evolutiva

- Para representación de vectores con valores reales, la programación evolutiva es muy similar a las estrategias evolutivas sin cruzamiento.
- Un método de selección típico es seleccionar a todos los individuos en la población para que sean μ ascendentes, para mutar a cada ascendente para generar μ descendientes. Después seleccionar los $\mu/2$ mejores ascendentes y $\mu/2$ descendientes para la siguiente población.

Programación Evolutiva

- PE aplica auto-adaptación de los parámetros de mutación.
- En esta variante un individuo sufre esta transformación:

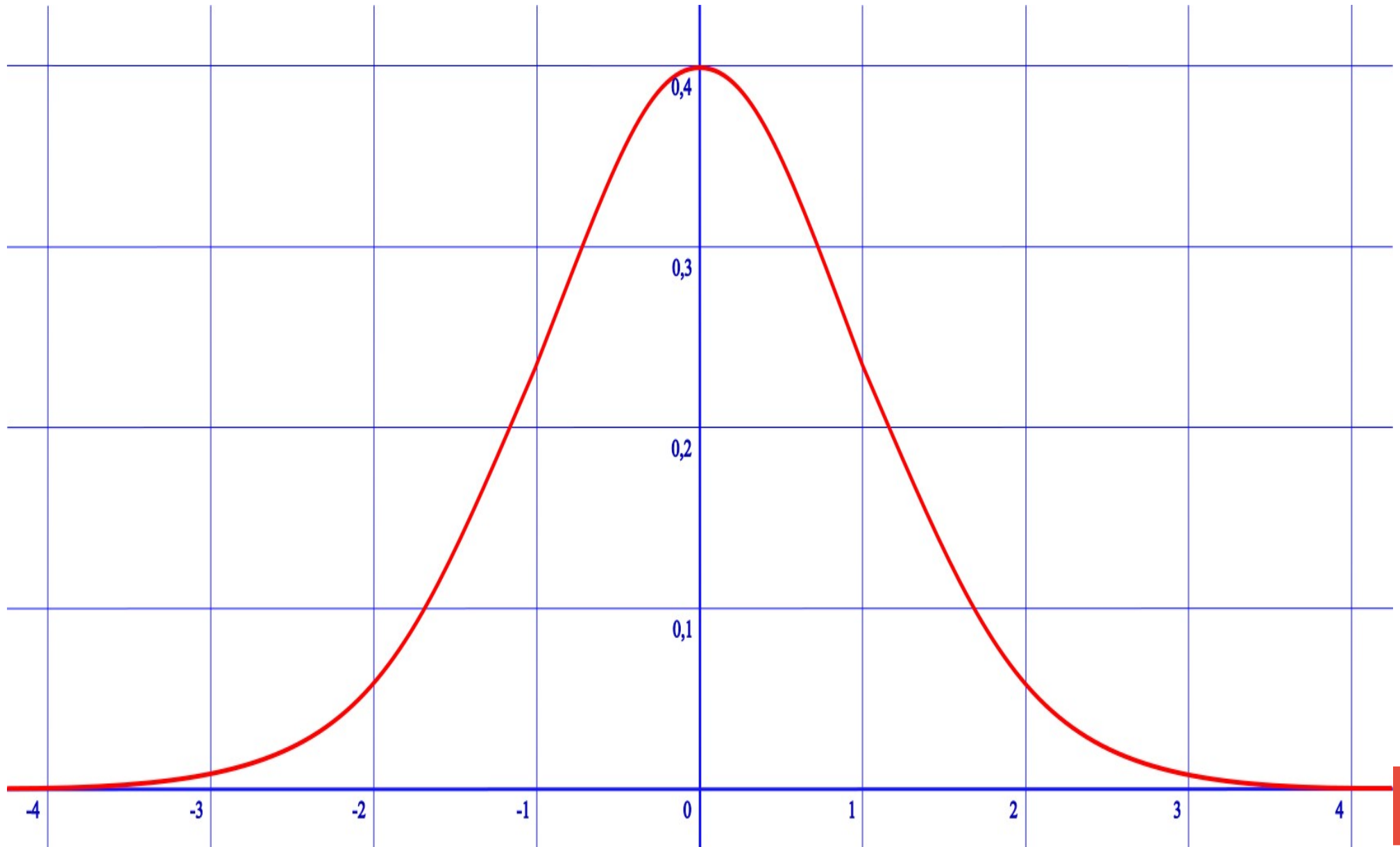
$$\langle x_1, \dots, x_n, \sigma_1, \dots, \sigma_n \rangle \quad \begin{aligned} \sigma'_i &= \sigma_i (1 + \alpha N(0,1)), \\ x'_i &= x_i + \sigma'_i N_i(0,1). \end{aligned} \quad \langle x'_1, \dots, x'_n, \sigma'_1, \dots, \sigma'_n \rangle$$
$$\alpha \approx 2$$

Programación Evolutiva

- Donde $N(0,1)$ es un vector de números Gaussianos independientes con una media de 0 y desviaciones estándar 1.
- La fórmula de una distribución normal está dada por:

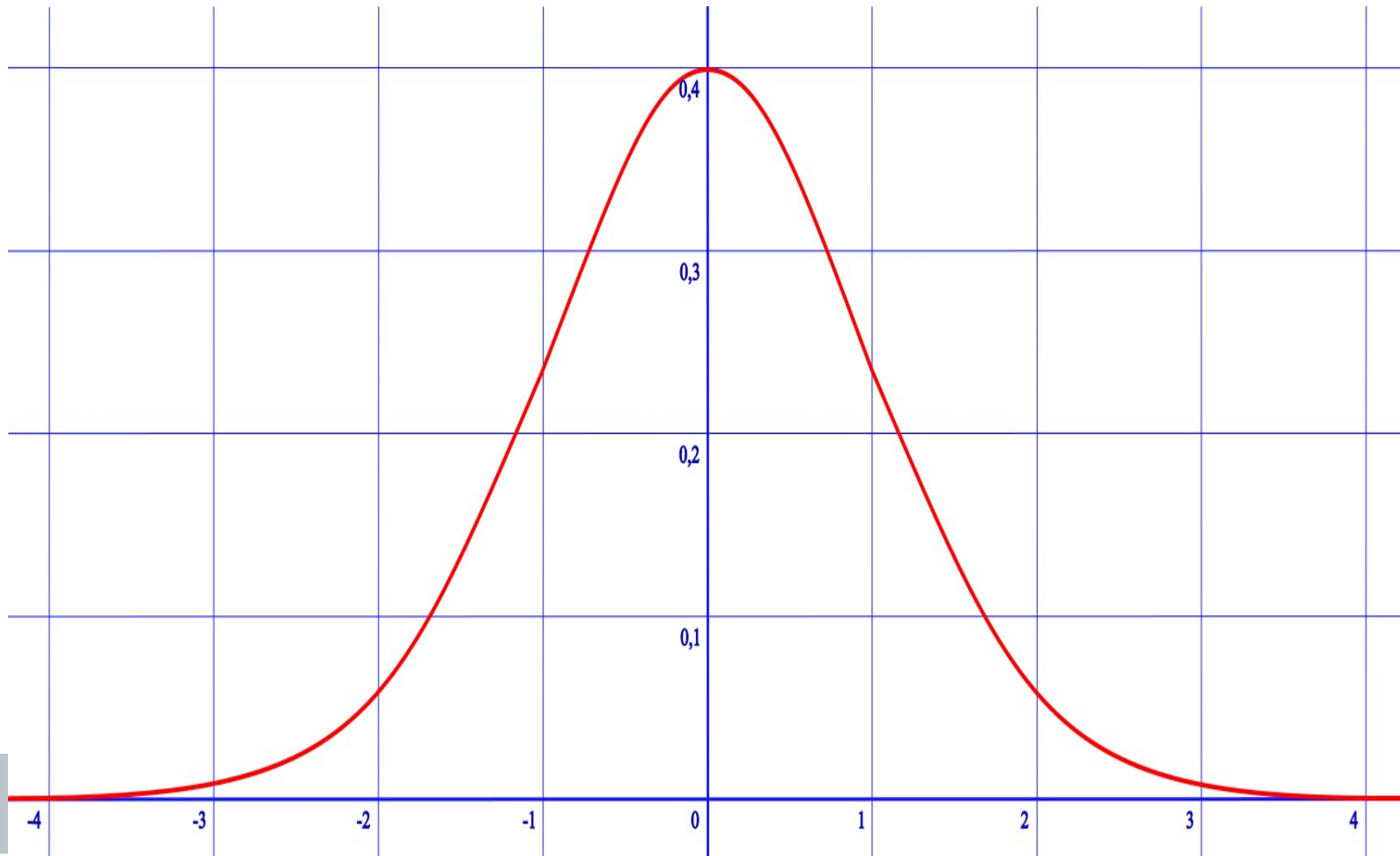
$$N(0, \sigma, x) = \frac{e^{-\frac{1}{2} \left(\frac{x}{\sigma} \right)^2}}{\sigma \sqrt{2\pi}}$$

Programación Evolutiva



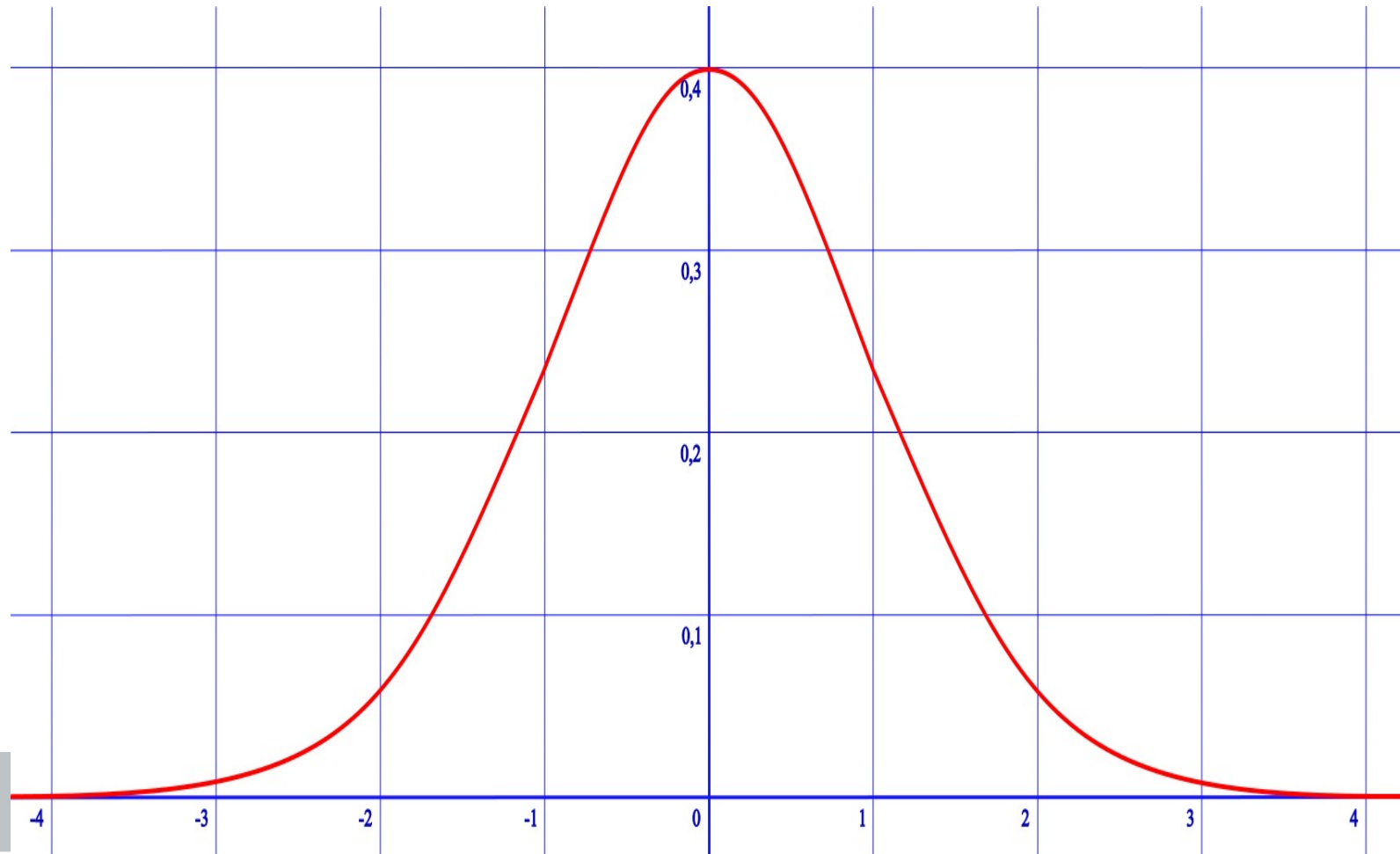
Programación Evolutiva

$$\frac{\left(e^{-\left(0.5 \times (\theta^2)\right)}\right)}{\sqrt{2 \times \pi}} = 0.3989422804014327$$



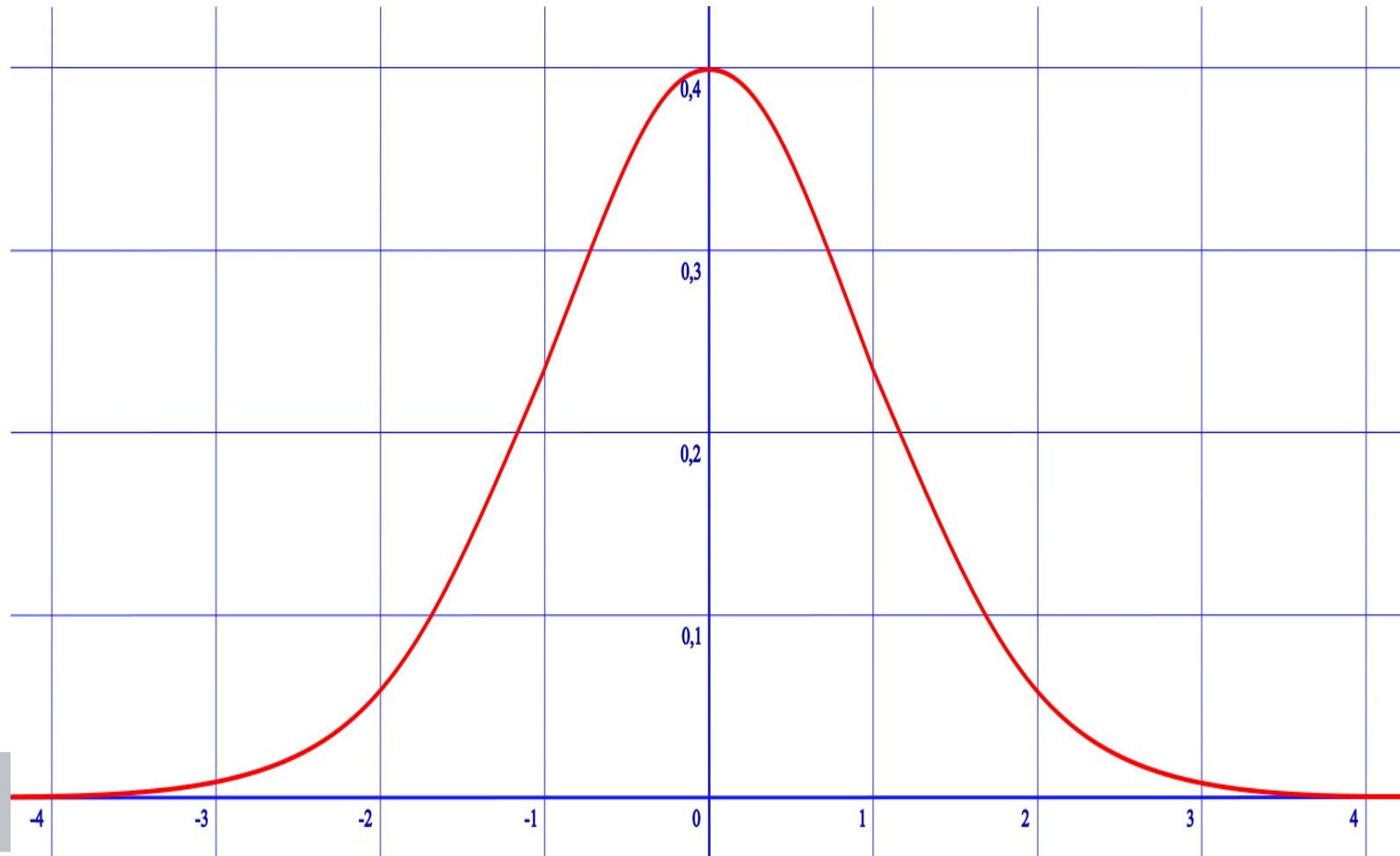
Programación Evolutiva

$$\frac{\left(e^{-\left(0.5 \times ((-2)^2)\right)}\right)}{\sqrt{2 \times \pi}} = 0.0539909665131881$$



Programación Evolutiva

$$\frac{\left(e^{-\left(0.5 \times (1^2)\right)}\right)}{\sqrt{2 \times \pi}} = 0.2419707245191434$$



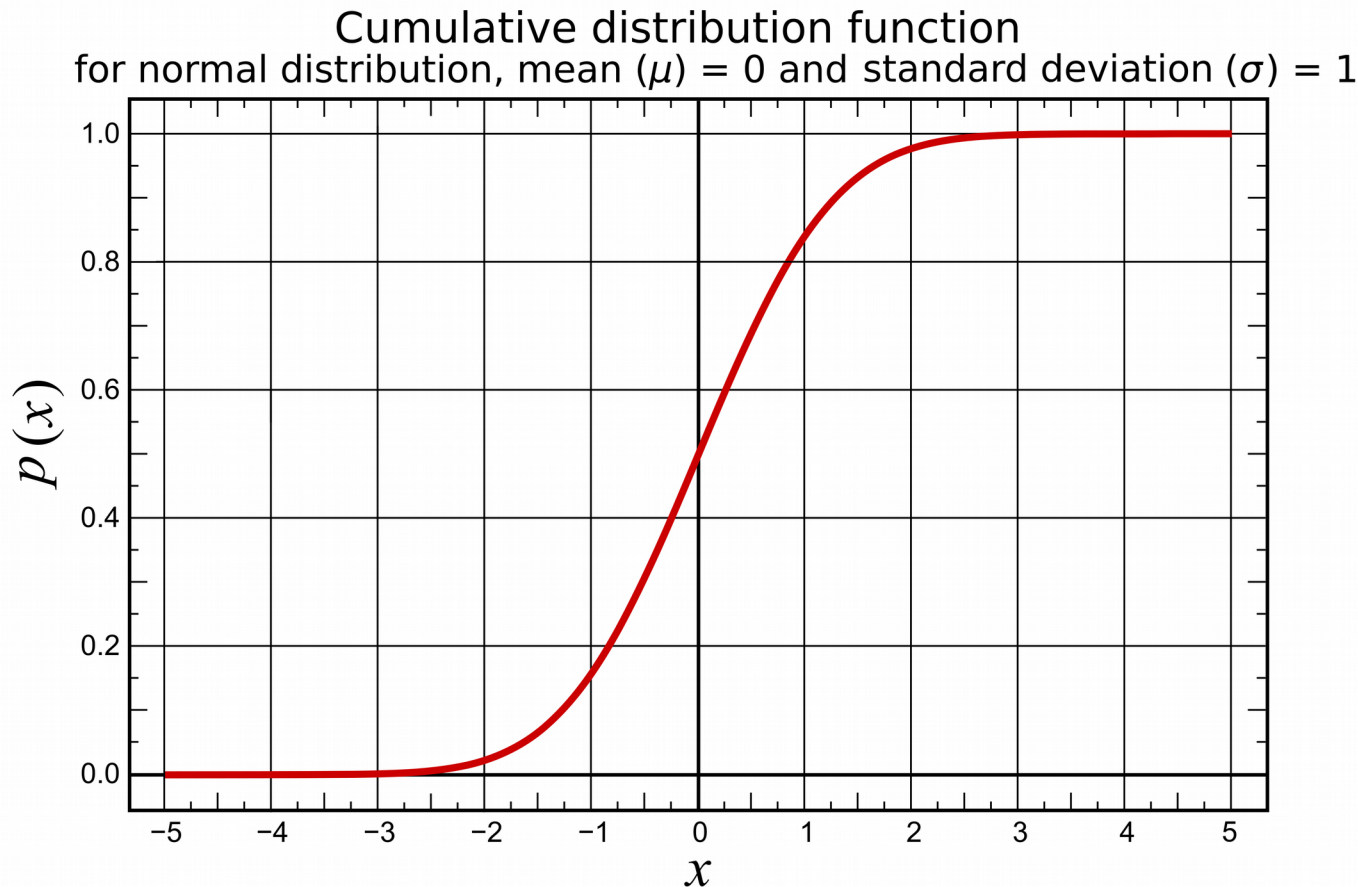
Programación Evolutiva

- Una distribución normal de media cero es implementada de la siguiente manera:

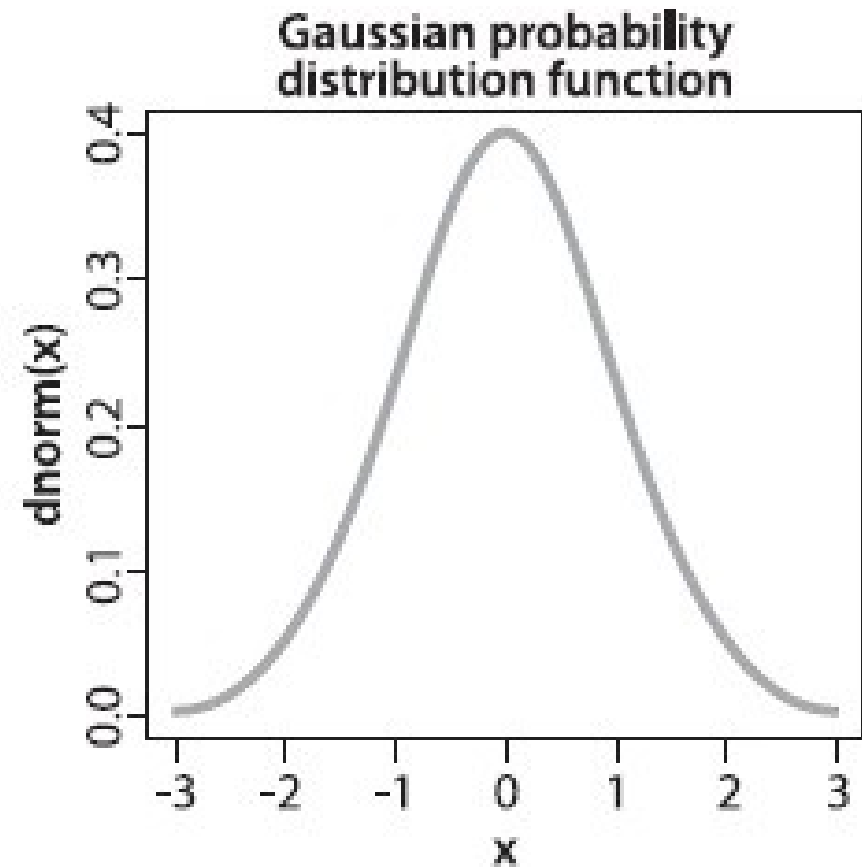
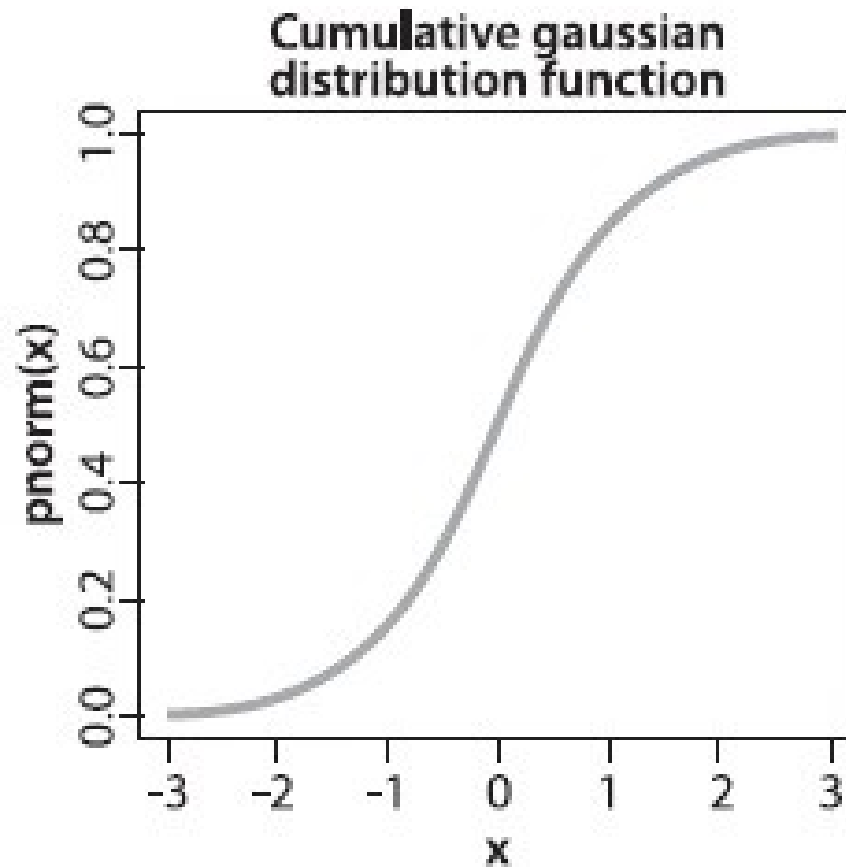
```
public static double normal(double x, double desvio) {  
    double retorno = -0.5 * ((x / desvio) * (x / desvio));  
    retorno = Math.exp(retorno);  
    retorno = retorno / (desvio * Math.sqrt(6.283184));  
    return retorno;  
}
```

Programación Evolutiva

- El área bajo la distribución de probabilidades corresponde a la probabilidad de ocurrencia del valor x . Por lo tanto, como se puede ver en la siguiente figura, el área total de una distribución de probabilidad es igual a 1.



Programación Evolutiva



Programación Evolutiva

- En base a ese concepto, podemos escoger cual será la variación de coordenadas sorteando un valor ϵ aleatorio entre el intervalo $(0,1)$ y determinar el valor de x , para el cual el área bajo la curva hasta x es igual al sorteado, es decir, el número x , para el cual la probabilidad de que un valor sorteado, cualquiera sea menor de que el, sea igual a ϵ .

Programación Evolutiva

- Para determinar esta probabilidad y calcular el valor de la mutación a aplicar, tenemos que calcular el valor de la integral dada por:

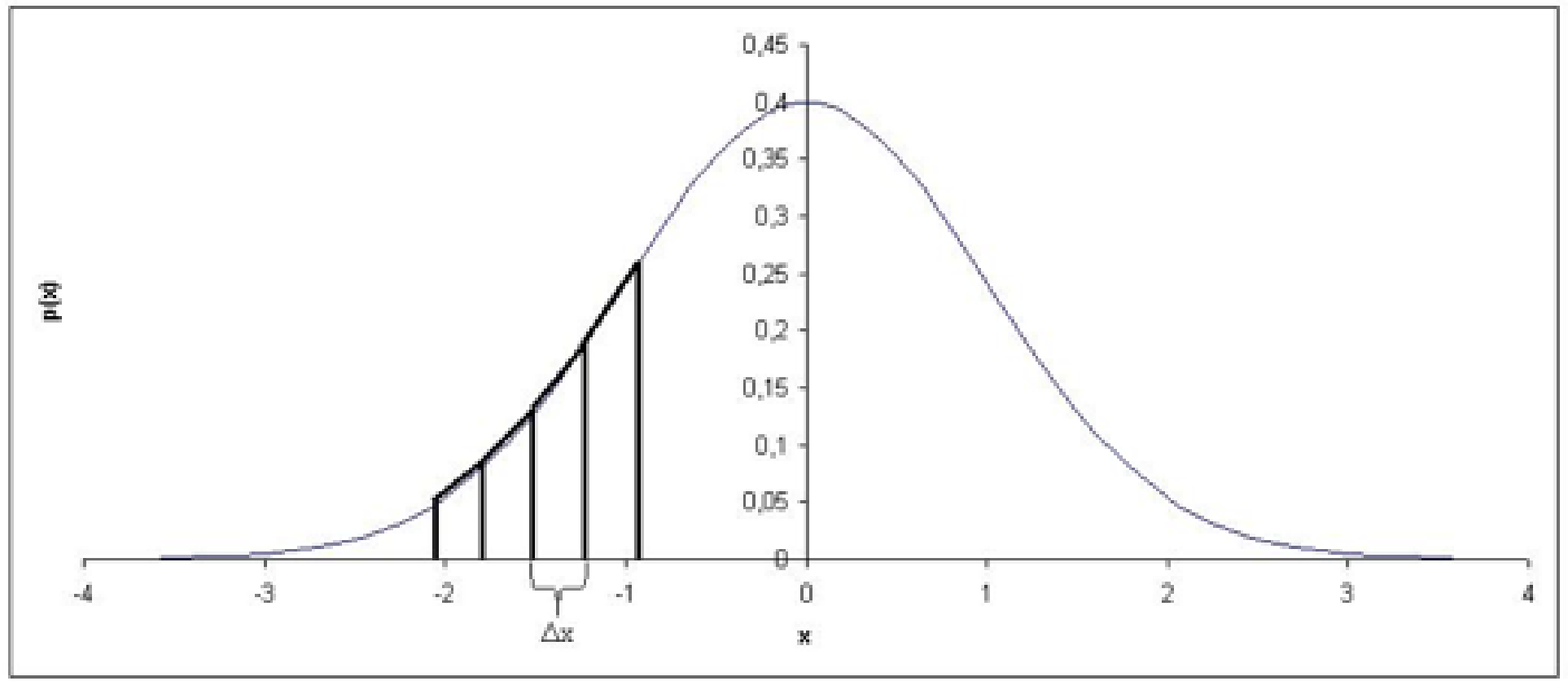
$$\int_{-\infty}^x N(0, \sigma, x) dx$$

- Como sabemos, no existe una forma definitiva para calcular esta integral, para ello usamos técnicas numéricas para implementarla.

Programación Evolutiva

- Usaremos el método de los trapecios repetidos:
- Idea:
 - Aproximar la curva a una serie de trapecios.
 - La base igual a Δx
 - Los dos lados son dados por los valores de la función de los puntos que distan Δx uno del otro.
 - Consiste básicamente en hacer una aproximación lineal por partes de la función que deseamos integrar.
 - La aproximación puede ser tan adecuada como desiemos, basta disminuir el valor de Δx

Programación Evolutiva



Programación Evolutiva

- El código que implementa la integración es el siguiente:

```
public static double integral(double lim_inf, double lim_sup, double desvio, double delta) {  
    double area = 0;  
    double aux_suma, aux = normal(lim_inf, desvio);  
  
    for (double i = lim_inf + delta; i < lim_sup; i += delta) {  
        aux_suma = normal (i, desvio);  
        area += (aux + aux_suma);  
        aux = aux_suma;  
    }  
    area *= (delta/2);  
    return area;  
}
```

Programación Evolutiva

- Al final multiplicamos la suma obtenida por $(\text{delta}/2)$, para obtener el área efectiva.
- Recordemos que el área de un paralelogramo está dada por $\text{base} * (\text{altura1} + \text{altura2})/2$.
- En este caso, los valores de la normal en cada división del intervalo corresponde a las alturas. El valor de la base es igual para todos, por lo que puede ser multiplicado al final.

Programación Evolutiva

- Podemos calcular el valor de x mediante:


```
public static double valor_x(double lim_inf, double lim_sup, double desvio, double delta, double aleatorio) {  
    double area = 0;  
    double aux_suma, aux = normal(lim_inf, desvio);  
  
    for (double i = lim_inf + delta; i < lim_sup; i += delta) {  
        aux_suma = normal (i, desvio);  
        area += (aux + aux_suma);  
        if((area * (delta/2)) > aleatorio){  
            return i;  
        }  
        aux = aux_suma;  
    }  
    return -1*Double.MAX_VALUE;  
}
```

Programación Evolutiva

- Sigue el mismo procedimiento visto anteriormente:

```
Procedure EP{  
    t = 0;  
    Initialize P(t);  
    Evaluate P(t);  
    While (Not Done)  
    {  
        Parents(t) = Select_Parents(P(t));  
        Offspring(t) = Procreate(Parents(t));  
        Evaluate(Offspring(t));  
        P(t+1) = Select_Survivors(P(t), Offspring(t));  
        t = t + 1;  
    }
```

No existe cruzamiento,
solo Mutación



Programación Evolutiva

- Con la diferencia de que la selección de la siguiente población es sobre la población de ascendientes y descendientes.
- Esta modificación también es llamada de Meta-PE o $PE(\mu+\mu)$.

Práctica 5 (0 a 20)

- Aplicar la PE($\mu+\mu$) vistas en clase para minimizar la siguiente función:

$$f(x_1, x_2) = -\cos(x_1) \cos(x_2) \exp(-(x_1 - \pi)^2 - (x_2 - \pi)^2)$$

$$-10 \leq x_1 \leq 10, -10 \leq x_2 \leq 10$$

- Utilice los parámetros por defecto mencionados y desviación estándar inicial 0.3. Utilice por los menos 5 decimales para los valores de x_1 y x_2 . La aptitud debe considerar los decimales necesarios.

GRACIAS

Dr. Edward Hinojosa Cárdenas
ehinojosa@unsa.edu.pe