Estrategias Evolutivas

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

Estrategias Evolutivas

- Las Estrategias Evolutivas (EE) fueron propuesta en los años 60 por Rechenberg y Schwefel en Alemania, corresponden a una de las principales variantes dentro de los Algoritmos Evolutivos.

- Son enfocados a resolver problemas de optimización continua de parámetros para control numérico.

Estrategias Evolutivas

- Aunque fueron desarrollados de forma paralela (no conjunta) con los AG, pueden ser descritos básicamente como un AG de representación real que usa un operador de mutación basado en una distribución normal.

- La característica principal de la EE es la adaptación de los parámetros que manejan el proceso evolutivo.

Estrategias Evolutivas

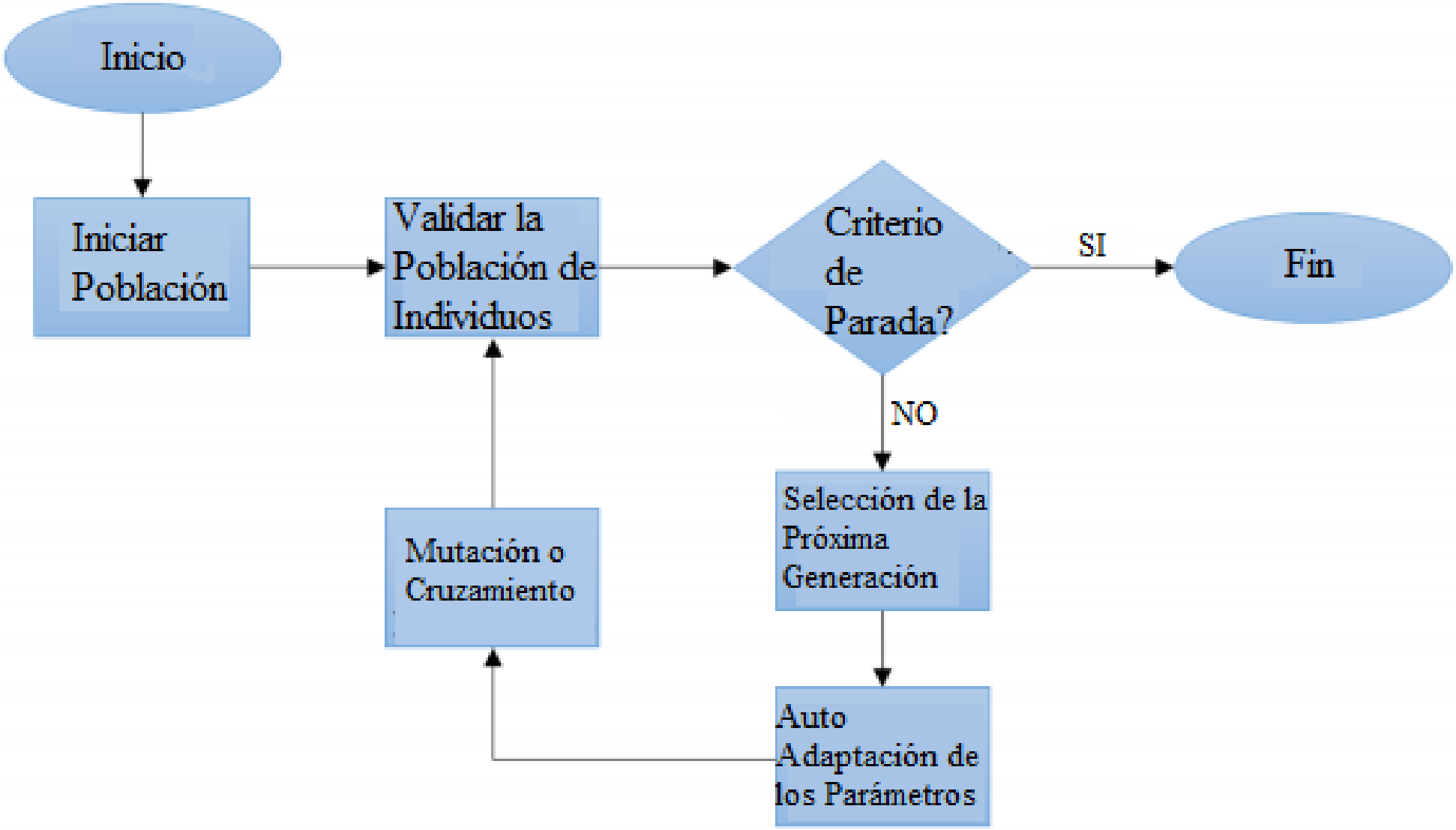
- Como consecuencia, un algoritmo basado en ES ejecuta simultáneamente dos tareas:
  - La solución de un problema de optimización específico; y
  - El ajuste del propio algoritmo para resolver el problema.

- La EE utiliza como operaciones una mutación auto-adaptativa y el cruzamiento con operadora de selección determinista en el proceso de búsqueda de la solución.

Estrategias Evolutivas

- Los algoritmos de EE trabajan con cantidades diferentes de individuos en la población y en el conjunto a partir del cual la próxima generación es creada. Esa es justamente la característica que diferencia los tipos de estrategias evolutivas:

Diagrama de Flujo del funcionamiento de los Algoritmos de EE



(1+1)-EE

- (1 + 1) − EE o estrategia evolutiva de dos miembros, utiliza sólo un padre y generaba un solo hijo.

- Este hijo se mantiene sólo si es mejor que el padre. A este tipo de selección se le llama extintiva, porque los peores individuos tienen una probabilidad de cero de ser seleccionados.

(1+1)-EE

- En la (1+1)-EE, un individuo nuevo es generado usando:

$$\bar{x}^{t+1} = \bar{x}^{t} + N(0, \bar{\sigma})$$

- Donde t se refiere a la generación (o iteración) en la que nos encontramos, y
- $N(0, \bar{\sigma})$ es un vector de números Gaussianos independientes con una media de cero y desviaciones estándar $\bar{\sigma}$.

(1+1)-EE

- Considere el siguiente ejemplo de una (1+1)-EE (estrategia evolutiva de dos miembros):

- Supongamos que queremos optimizar:

$$f(x_1, x_2) = 100(x_1^2 - x_2)^2 + (1 - x_1)^2$$

donde: $$-2.048 \leq x_1, x_2 \leq 2.048$$

(1+1)-EE

- Ahora, supongamos que nuestra población consiste del siguiente individuo (generado de forma aleatoria):

$$(\bar{x}^t, \bar{\sigma}) = (-1.0, 1.0), (1.0, 1.0)$$

- Supongamos también que las mutaciones producidas son las siguientes:

$$x_1^{t+1} = x_1^t + N(0, 1.0) = -1.0 + 0.61 = 0.39$$
$$x_2^{t+1} = x_2^t + N(0, 1.0) = 1 + 0.57 = 1.57$$

(1+1)-EE

- Ahora, comparamos al padre con el hijo:

Padre $\quad f(x_t) = f(-1.0, 1.0) = 4.0$

Hijo $\quad f(x_{t+1}) = f(-0.39, 1.57) = 201.416$

Dado que: $201.416 > 4.0$

el hijo reemplazará al padre en la siguiente generación.

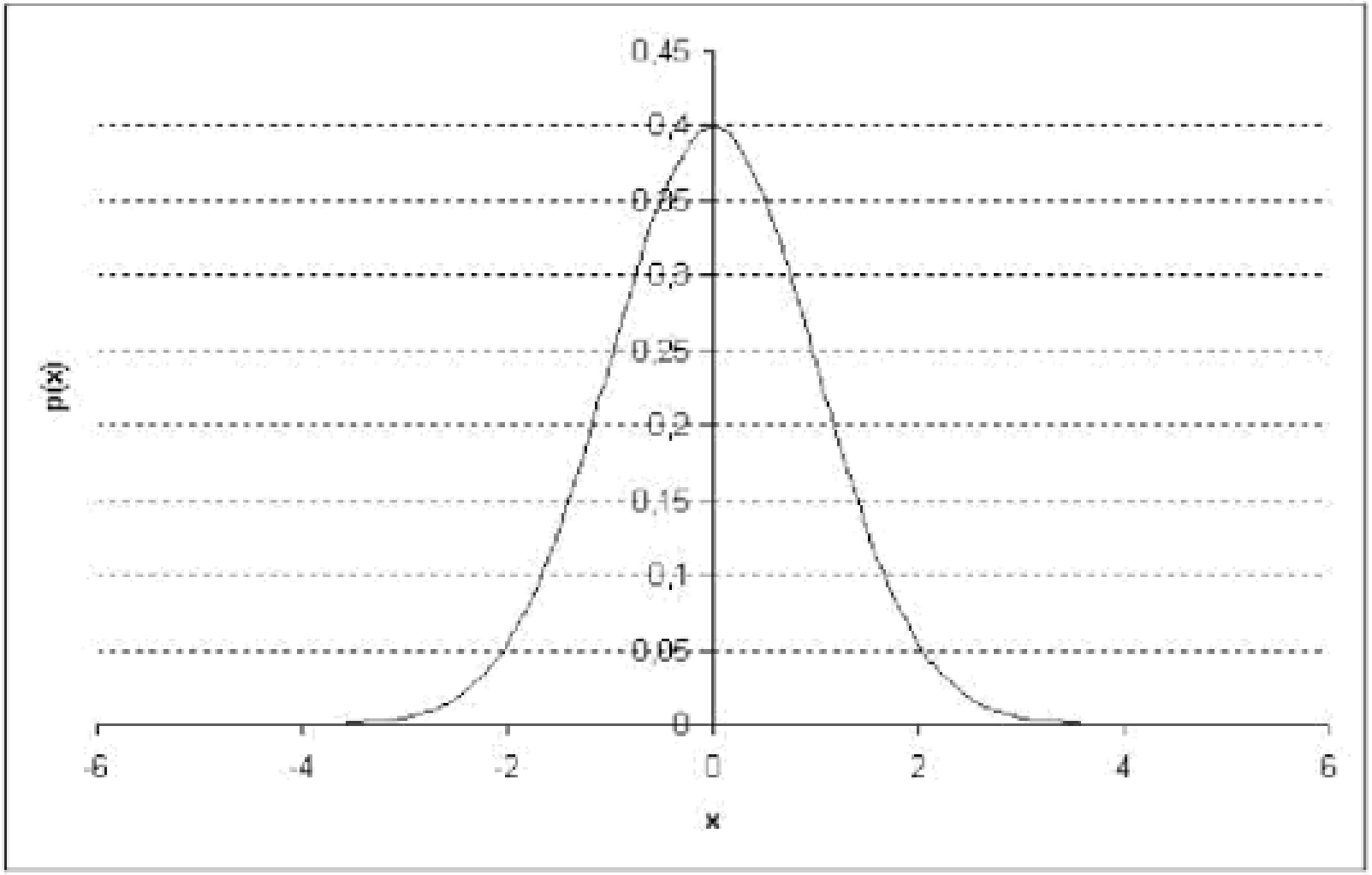
(1+1)-EE

- El operador de mutación usado en la EE está basado en una distribución de probabilidades normal o Gaussiana de media cero y desviación estándar $\bar{\sigma}$. representada por $N(0, \bar{\sigma})$ y conocida como distribución normal estándar.

- La fórmula de una distribución normal está dada por:

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

(1+1)-EE



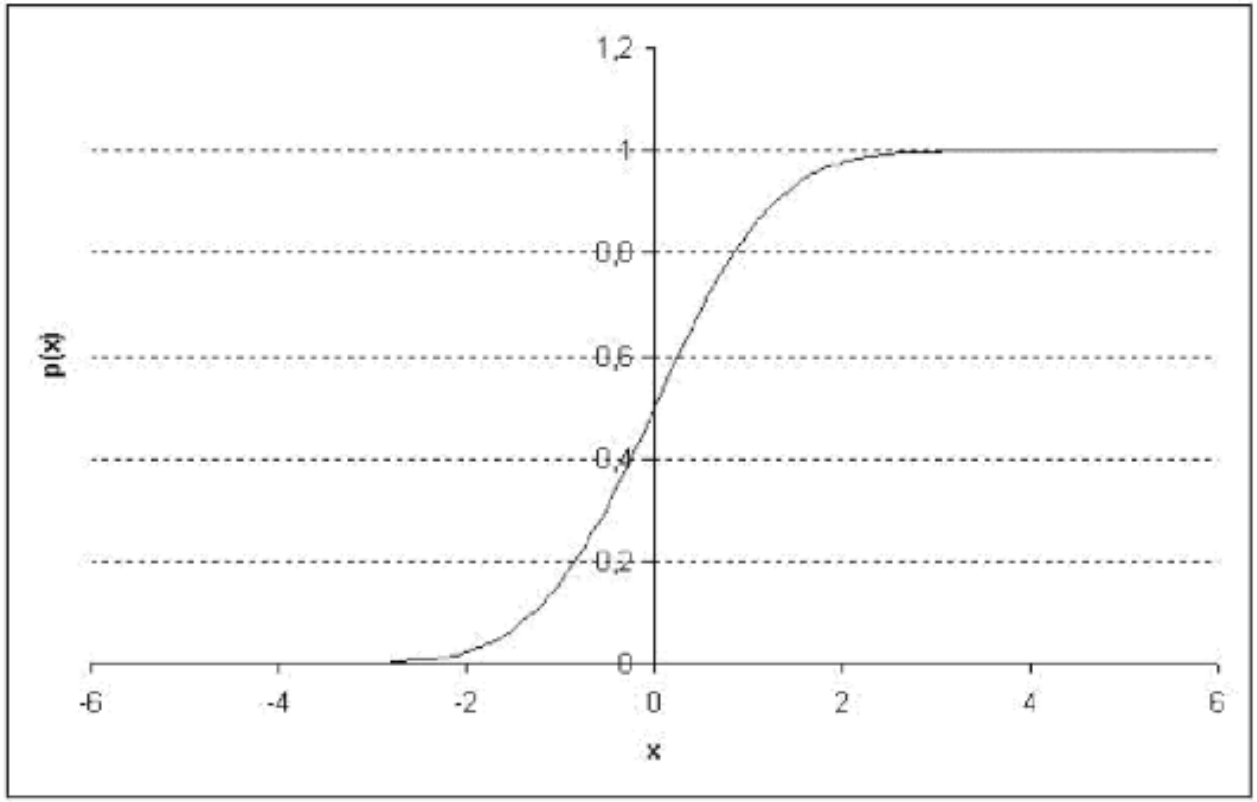
(1+1)-EE

- Una distribución normal de media zero 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;
}
```

(1+1)-EE

- 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.



(1+1)-EE

- En base a ese concepto, podemos escoger cual será la variación de coordenadas sorteando un valor Ɛ aleatorio entre el intervalo (0,1) y determinar el valor de x, para el cual el área bajo al 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 Ɛ.

(1+1)-EE

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

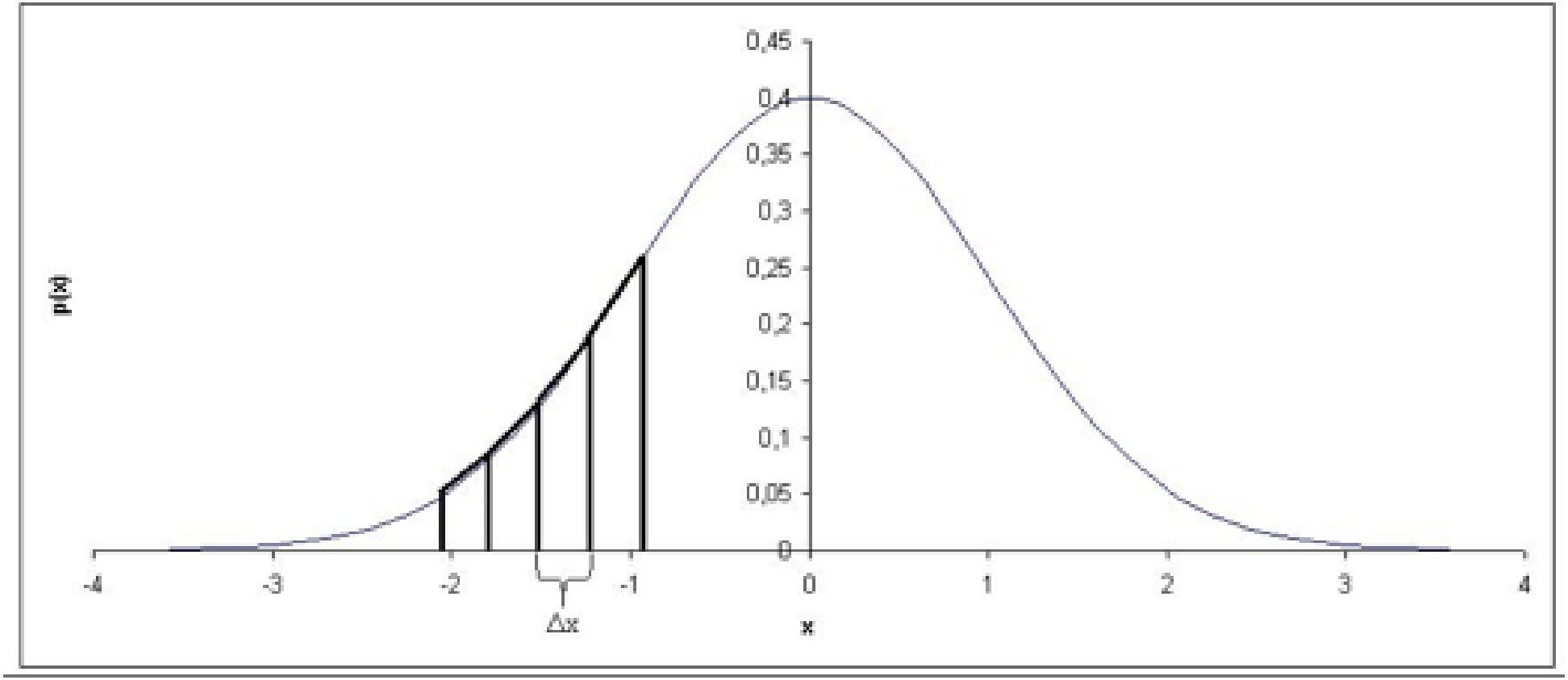
$$\int_{-\infty}^{x} N(0,\sigma,x)dx$$

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

(1+1)-EE

- 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

(1+1)-EE



(1+1)-EE

- 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;
}
```

(1+1)-EE

- Al final multiplicamos la suma obtenida por (delta/2), para obtener el área efectiva.

- Recordemos que el área de una paralelogramo está dada por base*(altura1 + 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.

(1+1)-EE

- 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;
}
```

(1+1)-EE

- La fórmula de mutación:

$$\bar{x}^{t+1} = \bar{x}^{t} + N(0, \bar{\sigma})$$

- Implica que solo podemos adicionar valores positivos al valor de una coordenada. Verdadero o Falso? Justifique.

(1+1)-EE

- Falso. Recuerde que para calcular el valor a ser adicionado, es generado un número aleatorio entre 0 y 1 y calculamos la integral de la función normal, de -Infinito hasta x, de forma que el valor de la integral sea igual al valor aleatorio.

- Cuando la distribución normal es simétrica en base a cero, mitad del área queda en el cuadrante negativo y la otra mitad en la mitad del cuadrante positivo.

(1+1)-EE

- Así, que si el número aleatorio fuera en le intervalo (0, 0.5), el valor de x siempre será menor que cero y el valor negativo será adicionado a la coordenada i.

- Podemos garantizar que no solo es falsa la sentencia, sino que la probabilidad de adicionar un número negativo es del 50%.

(1+1)-EE – Auto Adaptación

- Manteniendo un valor de desviación estándar constante, las EE convergirán en una solución aceptable u óptima.

- No existe una limitación para el tiempo en que ello ocurrirá.

- El valor inicial de la desviación estándar es aleatorio (Se suele utilizar 1.0, 0.5 o 0.3).

(1+1)-EE – Auto Adaptación

- Rechenberg formuló una regla para ajustar la desviación estándar de forma determinística durante el proceso evolutivo de tal manera que el procedimiento convergiera hacia el óptimo.

- Esta regla se conoce como la “regla del éxito 1/5”, y en palabras dice:

(1+1)-EE – Auto Adaptación

- “La razón entre mutaciones exitosas y el total de mutaciones debe ser 1/5. Si es mayor, entonces debe incrementarse la desviación estándar. Si es menor, entonces debe decrementarse”.

- La idea básica de esta regla era que idealmente 1/5 de los hijos generados debe ser mejor que la de sus padres. Si el índice de mejora fuera menor de que 1/5, quiere decir que estamos cerca a un máximo local y la búsqueda debe seguir en pasos menores, lo que implica en concentrarse cerca al padre actual, disminuyendo la desviación estándar.

(1+1)-EE – Auto Adaptación

- Si el índice fuera fuera mayor que 1/5, probablemente estamos lejos de una máximo, lo que nos dice que debemos aumentar al desviación estándar de forma de hacer una evaluación más amplia del espacio de búsqueda, lo que puede hacer que aceleremos la convergencia del algoritmo.

if $t \mod n = 0$ **then**

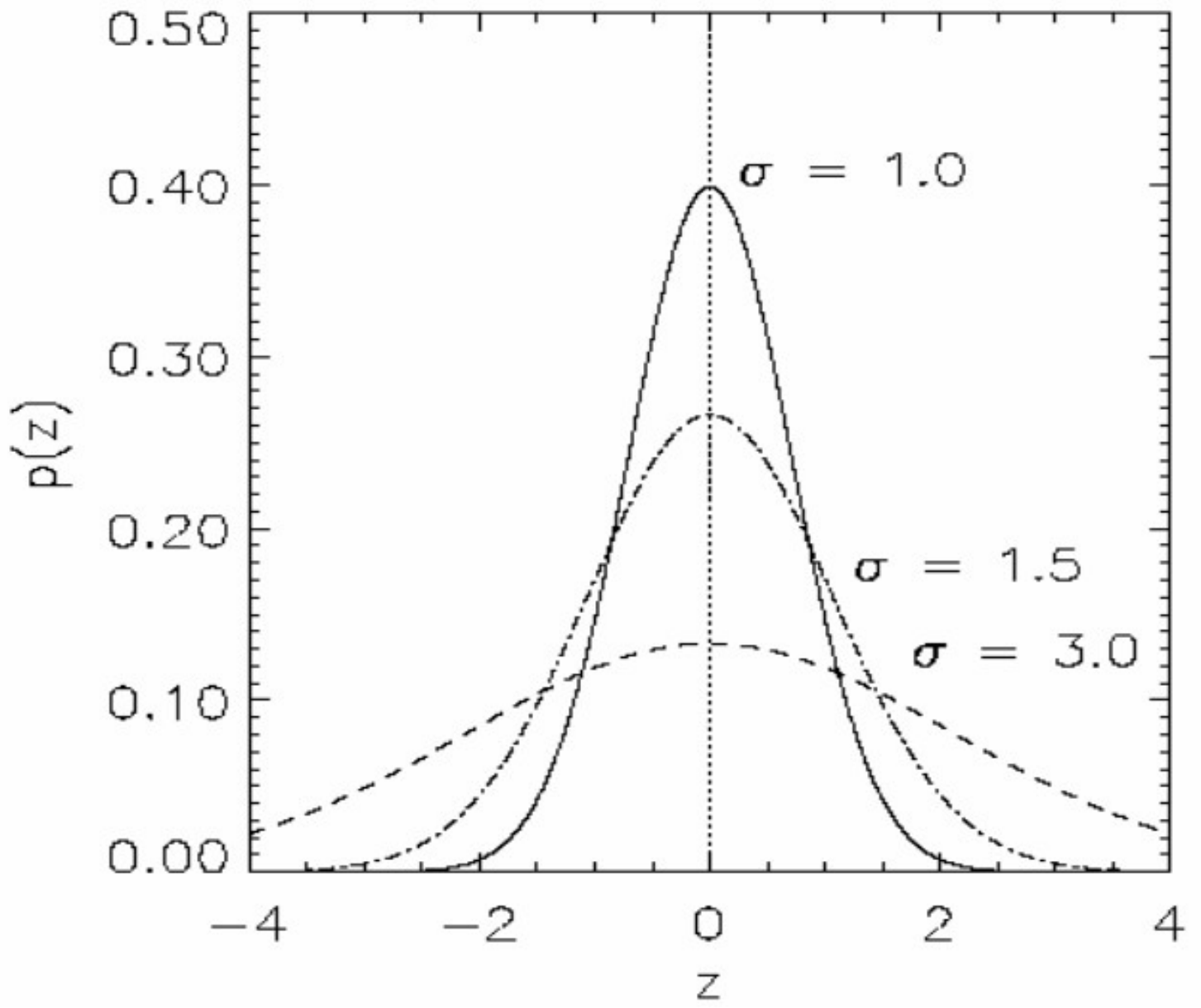
get #*successes* and #*failures*

$$p_S = \frac{\#successes}{\#successes + \#failures}$$

$$\sigma' \leftarrow \begin{cases} \sigma \cdot c \text{ if } p_S < 1/5 \\ \sigma / c \text{ if } p_S > 1/5 \\ \sigma \text{ if } p_S = 1/5 \end{cases}$$

end if

(1+1)-EE – Auto Adaptación



(1+1)-EE – Auto Adaptación

- Donde n es la cantidad de mutaciones, p_s es la frecuencia relativa de mutaciones exitosas medida en n mutaciones, y c=0.817 (este valor fue derivado teóricamente por Schwefel).

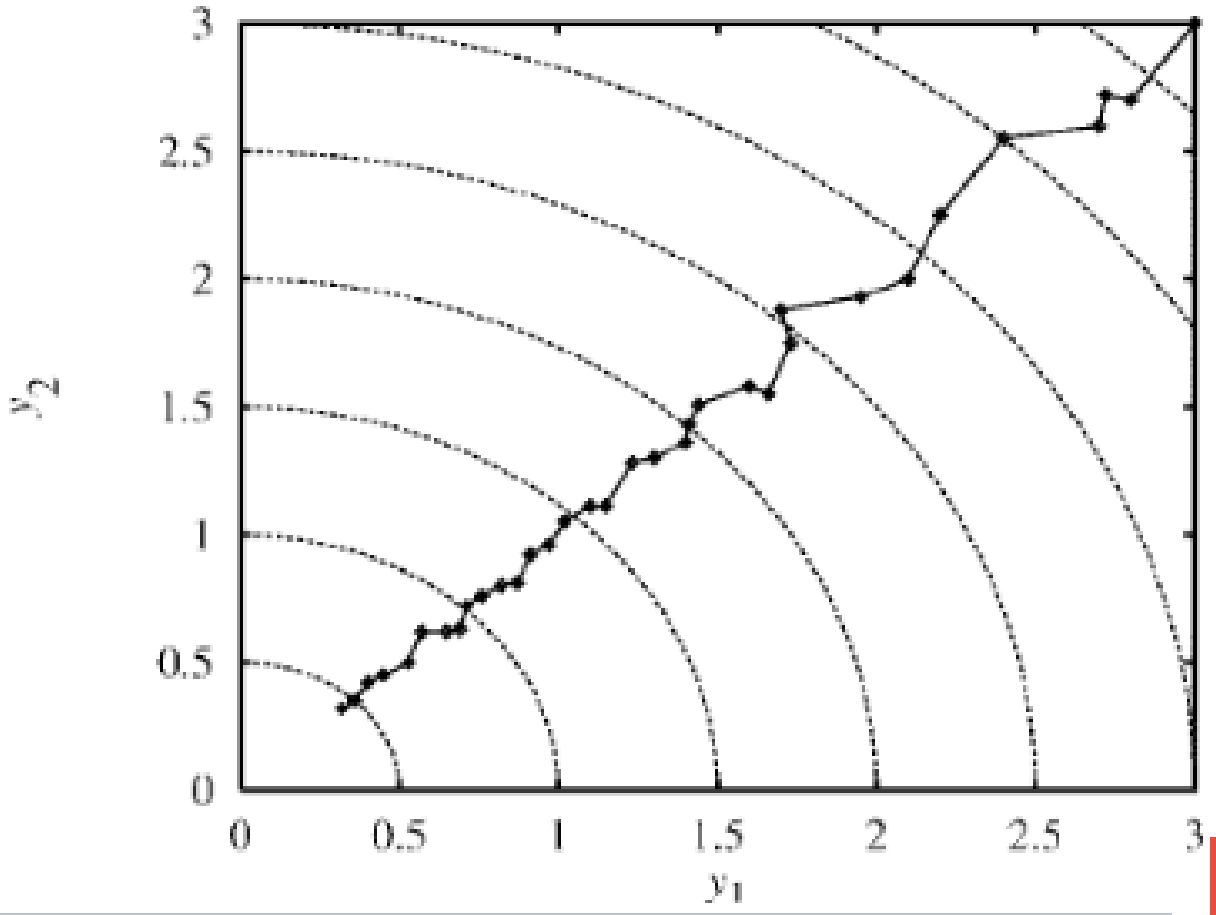
- El algoritmo (1+1)-EE sigue los siguientes pasos:

(1+1)-EE – Auto Adaptación

- En t = 0 se genera un individuo al azar.
- Se aplica el operador mutación al individuo obtenido.
- De los dos individuos (el original y el mutado) se selecciona el que tenga mejor evaluación.
- En caso se llegue a un número determinado n de mutaciones actualizar la desviación estándar.
- El proceso continúa hasta que se satisfaga la condición de terminación.
- El último individuo obtenido es el que tiene mejor adaptación y es la solución del problema.

(1+1)-EE – Auto Adaptación

- Por último, cuál es la diferencia de este algoritmo en comparación con el Hill Climbing o Ascenso de Colina?

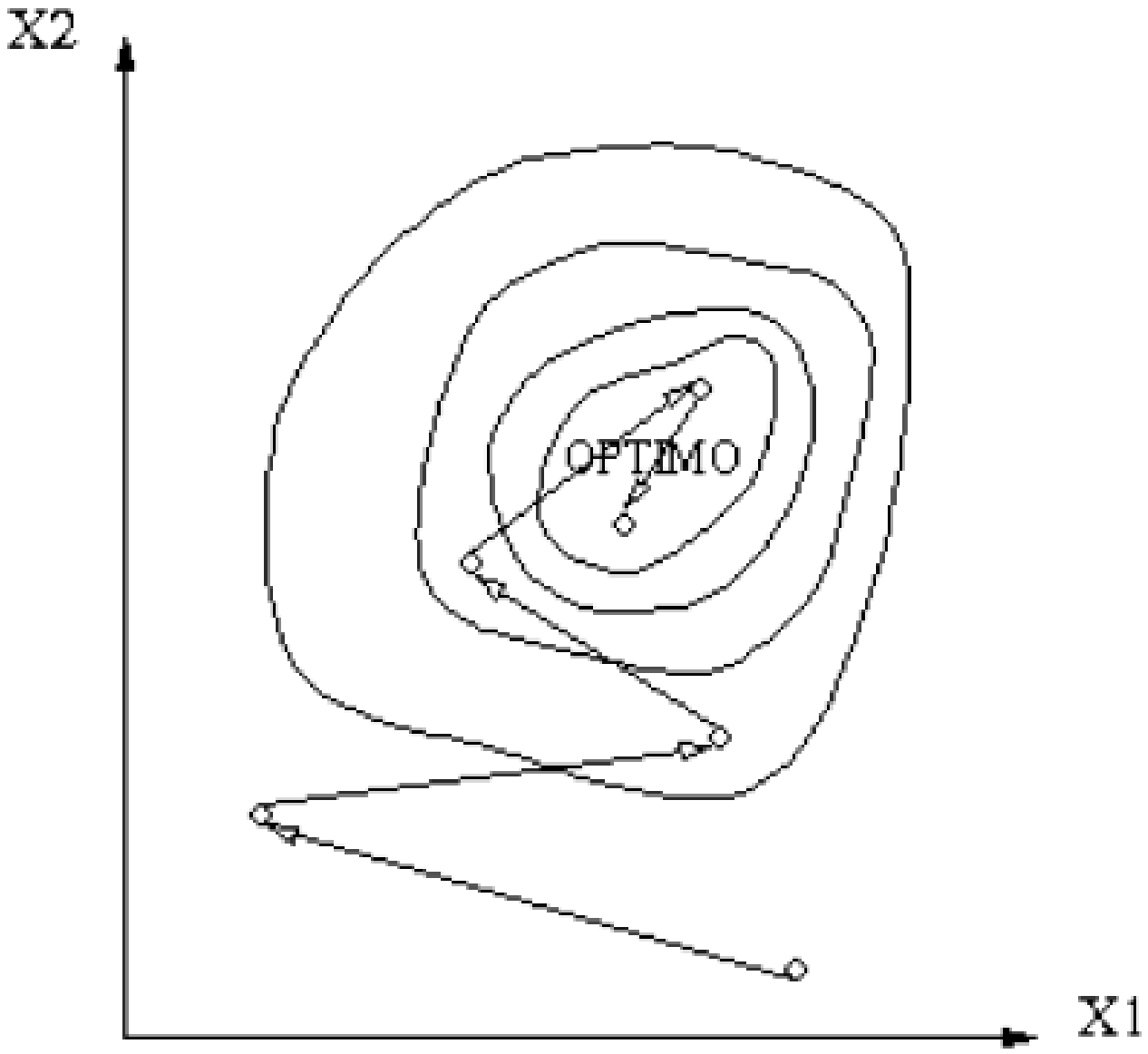


(1+1)-EE – Auto Adaptación

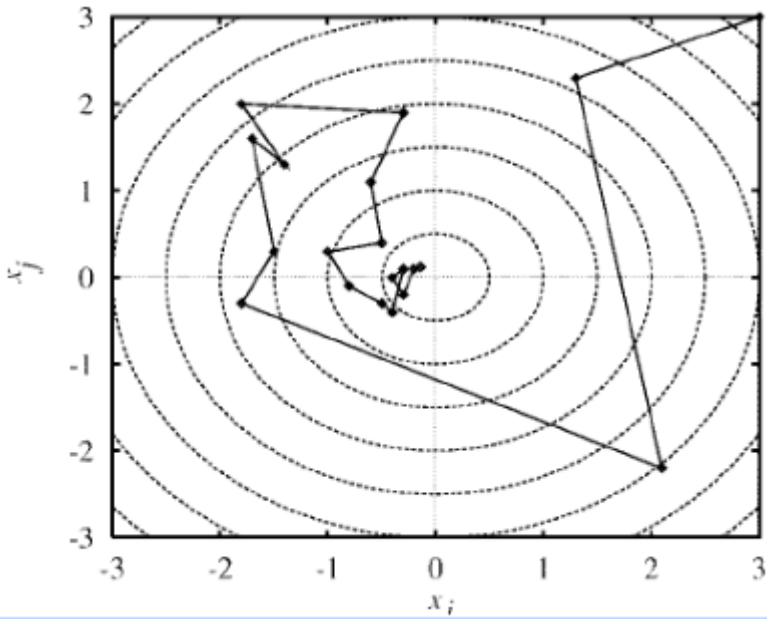
- Por último, cuál es la diferencia de este algoritmo en comparación con el Hill Climbing o Ascenso de Colina?

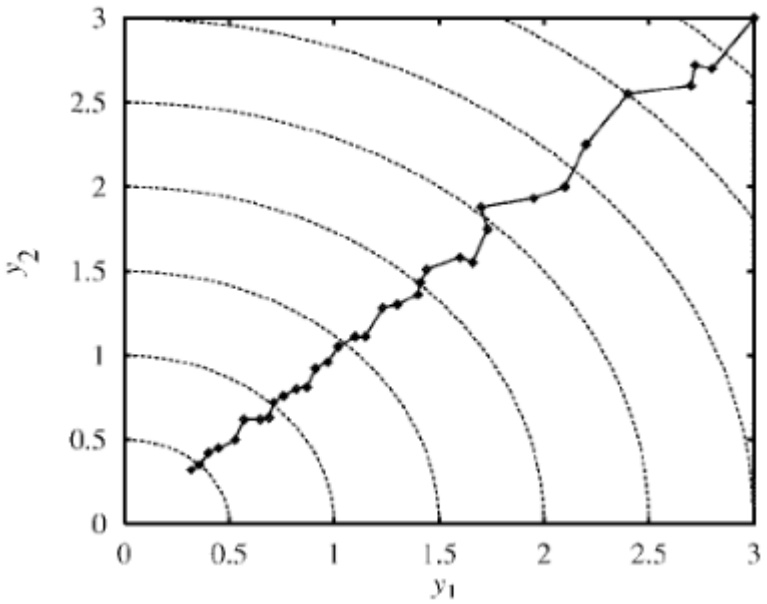
- En EE la mutación depende de la desviación estándar que puede ser mayor o menor, dado que dicho parámetros también evoluciona. Además, la trayectoria de búsqueda sigue un camino en zigzah.

(1+1)-EE – Auto Adaptación



(1+1)-EE – Auto Adaptación





Laboratorio 05 (0 a 20)

- Implemente el algoritmo (1+1)-EE para maximizar la función:

$$f(x_1, x_2) = 100(x_1^2 - x_2)^2 + (1 - x_1)^2$$

donde: $-2.048 \leq x_1, x_2 \leq 2.048$

- Utilice los parámetros por defecto mencionados y desviación estándar inicial 0.3. Muestre la aptitud y desviación estándar en cada iteración.

- Considere 1000 iteraciones.

GRACIAS

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