

# BIOINFORMÁTICA

## 2013 - 2014

---

### PARTE I. INTRODUCCIÓN

- Tema 1. Computación Basada en Modelos Naturales

### PARTE II. MODELOS BASADOS EN ADAPTACIÓN SOCIAL (Swarm Intelligence)

- Tema 2. Introducción a los Modelos Basados en Adaptación Social
- Tema 3. Optimización Basada en Colonias de Hormigas
- Tema 4. Optimización Basada en Nubes de Partículas (Particle Swarm)

### PARTE III. COMPUTACIÓN EVOLUTIVA

- Tema 5. Introducción a la Computación Evolutiva
- Tema 6. Algoritmos Genéticos I. Conceptos Básicos
- Tema 7. Algoritmos Genéticos II. Diversidad y Convergencia
- Tema 8. Algoritmos Genéticos III. Múltiples Soluciones en Problemas Multimodales
- Tema 9. Estrategias de Evolución y Programación Evolutiva
- Tema 10. Algoritmos Basados en Evolución Diferencial (Differential Evolution – DE)
- Tema 11. Modelos de Evolución Basados en Estimación de Distribuciones (EDA)
- Tema 12. Algoritmos Evolutivos para Problemas Multiobjetivo
- Tema 13. Programación Genética
- Tema 14. Modelos Evolutivos de Aprendizaje

### PARTE IV. OTROS MODELOS DE COMPUTACIÓN BIOINSPIRADOS

- Tema 15. Sistemas Inmunológicos Artificiales
- Tema 16. Otros Modelos de Computación Natural/Bioinspirados

# BIOINFORMÁTICA

## TEMA 9. ESTRATEGIAS DE EVOLUCIÓN Y PROGRAMACIÓN EVOLUTIVA

---

### 1. ESTRATEGIAS DE EVOLUCIÓN

### 2. PROGRAMACIÓN EVOLUTIVA

*Schwefel, H.P. Evolution and Optimum Seeking. Sixth-Generation Computer Technology Series. Wiley, New York, 1995.*

*D. Fogel, Evolutionary Computation. Toward a New Philosophy of Machine Intelligence. IEEE Press, 1995.*

# 1. ESTRATEGIAS DE EVOLUCIÓN

---

## 1. INTRODUCCION

## 2. MODELO EE-(1+1)

## 3. MODELOS EE-( $\mu, \lambda$ ), EE-( $\mu + \lambda$ )

*Schwefel, H.P. Evolution and Optimum Seeking. Sixth-Generation Computer Technology Series. Wiley, New York, 1995.*

# 1.1. EE. INTRODUCCIÓN

---

Desarrolladas en 1964 en la Universidad Técnica de Berlín (TUB) por Rechenberg y Schwefel como una técnica experimental de optimización para la representación basada en codificación real.

La primera variante fue llamada EE-(1+1), trabajando sobre la base de dos individuos, **un padre y un descendiente por generación**. El mejor de ambos pasaba a ser padre de la siguiente generación.

Rechenberg (1973) desarrolló una teoría sobre la velocidad de convergencia y obtuvo una regla para determinar la adaptación de la desviación estandar de la mutación de acuerdo a una medida de frecuencia de éxitos, regla llamada **como regla de 1/5-éxito**.

Posteriormente la EE-(1+1) se sustituyo por variantes con  $\mu > 1$  padres y  $\lambda > 1$  descendientes por generación. Con una característica básica, la **autoadaptación** de los parámetros durante el proceso de búsqueda evolutiva.

# 1.1. EE. INTRODUCCIÓN. RÁPIDO RESUMEN

---

- Desarrollo: Alemania en los años 70
- Primeros autores: I. Rechenberg, H.-P. Schwefel
- Aplicación típica:
  - Optimización numérica
- Características atribuidas:
  - Rapidez
  - Buenos optimizadores para problemas con variables reales
  - Bastantes estudios teóricos
- Característica Especial:
  - Auto-adaptación de (mutación) los parámetros estándar

## 1.2. MODELO EE-(1+1)

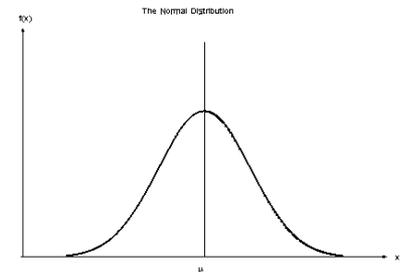
---

Obtiene un descendiente aplicando un operador de mutación sobre el padre

$$\text{Mut: } \mathbf{I} \longrightarrow \mathbf{I}$$

$$\text{Mut}(\mathbf{x}) = \mathbf{x}' = (x_1 + z_1, \dots, x_n + z_n)$$

$$z_i \sim N_i(0, \sigma'^2)$$



Mutación asociada a una distribución normal de media 0 y varianza  $\sigma'^2$

## 1.2. MODELO EE-(1+1)

---

La mutación se adapta de acuerdo a la regla de 1/5 de éxitos con  $c = 0.85$  (Schwefel, 95):

$$\sigma' = \text{mut}(\sigma) = \begin{cases} \sigma/c & \text{si } p > 1/5, \\ \sigma \cdot c & \text{si } p < 1/5, \\ \sigma & \text{si } p = 1/5. \end{cases}$$

**Modo de actuación:** Después de  $n$  mutaciones se chequea cuantos éxitos han ocurrido sobre las precedentes  $10 \cdot n$  mutaciones. Si el número de éxitos es menor que  $2 \cdot n$  se multiplica la longitud de paso por el factor 0.85; si es mayor que  $2 \cdot n$  se divide por 0.85.

## 1.2. MODELO EE-(1+1)

---

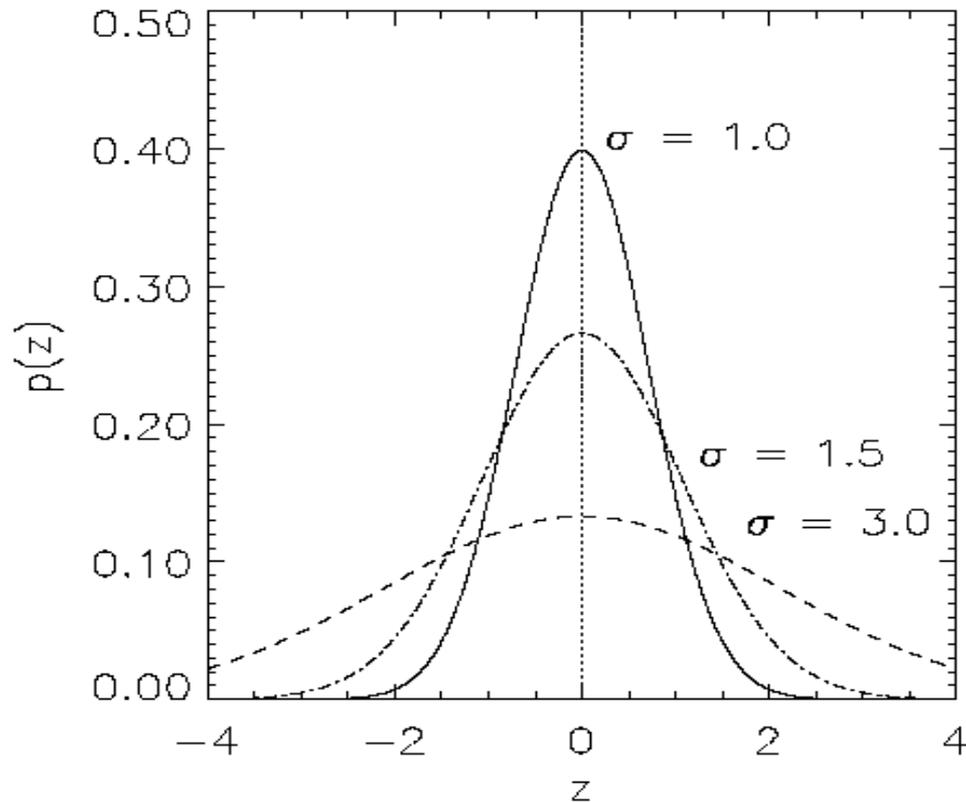
### ALGORITMO EE-(1+1)

- Set  $t \leftarrow 0$
- Create initial point  $x^t = \langle x_1^t, \dots, x_n^t \rangle$
- REPEAT UNTIL (*TERMIN.COND* satisfied) DO
  - Draw  $z_i$  from a normal distr. for all  $i = 1, \dots, n$
  - $y_i^t \leftarrow x_i^t + z_i$
  - IF  $f(x^t) < f(y^t)$  THEN  $x^{t+1} \leftarrow x^t$  ELSE  $x^{t+1} \leftarrow y^t$
  - ACTUALIZAR  $\sigma$
  - Set  $t \leftarrow t+1$

## 1.2. MODELO EE-(1+1)

---

### Ilustración de una distribución normal



## 1.2. MODELO EE-(1+1)

---

Se pueden especificar diferentes tamaños de salto si definimos  $z_i \sim N_i(0, (\sigma' \cdot s_i)^2)$ , donde  $s_i$  es fijo, se utiliza para adaptar  $\sigma$  a cada variable y su valor se determina en la etapa inicial del algoritmo.

**Característica:** La estrategia **1/5 de éxitos** puede causar **convergencia prematura** debido al rápido descenso de la longitud de salto cuando no se producen éxitos.

El algoritmo se puede catalogar como una **estrategia de búsqueda local**.

## 1.3. MODELOS $EE-(\mu,\lambda)$ , $EE-(\mu+\lambda)$

---

La desventaja de la regla 1/5 éxito para controlar la longitud de salto de la estrategia de evolución simple  $EE(1+1)$  llevó a la búsqueda de una estrategia más robusta para ajustar los parámetros para la mutación del algoritmo.

La desviación estándar para mutar se representa como una parte de los individuos y evoluciona por medio de mutación y recombinación como una variable más en un proceso llamado **auto-adaptación** de los parámetros de la estrategia.

## 1.3. MODELOS EE-( $\mu, \lambda$ ), EE-( $\mu + \lambda$ )

---

La representación de un individuo viene dada por el espacio de búsqueda:

$$I = \mathcal{R}^n \times \mathcal{R}^{n_\sigma} \times [-\pi, \pi]^{n_\alpha}$$

**$n_\sigma = 1, n_\alpha = 0$ :** Mutación estandar con una solo parámetro para controlar la mutación.

**$n_\sigma = n, n_\alpha = 0$ :** Mutaciones estandar con diferentes valores para controlar la mutación de las variables.

**$n_\sigma = n, n_\alpha = n(n-1)/2$ :** Mutaciones correladas con una matriz de covarianza para cada individuo.

**$n_\sigma = n, n_\alpha = n-1$ :** Se da una dirección arbitraria para el espacio de búsqueda.

## 1.3. MODELOS EE- $(\mu, \lambda)$ , EE- $(\mu + \lambda)$

---

### MODELO EE- $(\mu, \lambda)$

ALGORITHM 3  $((\mu, \lambda)$ -ES)

```
t := 0;
initialize  $P^{(0)} = \{\bar{a}_1, \dots, \bar{a}_\mu\} \in I^\mu$ ;
evaluate  $f(\bar{x}_1), \dots, f(\bar{x}_\mu)$ ;
while  $(T(P^{(t)}) = 0)$  do
     $\tilde{P} := \emptyset$ ;
    for i := 1 to  $\lambda$  do
         $(\tilde{x}, \tilde{\sigma}, \tilde{\alpha}) := \text{mut}(\text{rec}(P^{(t)}))$ ;
        evaluate  $f(\tilde{x})$ ;
         $\tilde{P} := \tilde{P} \sqcup \{(\tilde{x}, \tilde{\sigma}, \tilde{\alpha})\}$ ;
    od
     $P^{(t+1)} := \text{sel}_\mu^\lambda(\tilde{P})$ ;
    t := t + 1;
od
```

## 1.3. MODELOS EE- $(\mu, \lambda)$ , EE- $(\mu + \lambda)$

---

### MODELO EE- $(\mu, \lambda)$

El operador **rec**:  $\mathbf{I}^\mu \rightarrow \mathbf{I}$  crea un individuo por aplicación. Inicialmente selecciona  $\rho$  ( $1 \leq \rho \leq \mu$ ) padres de la población con probabilidad uniforme y los combina para crear un descendiente sobre el que aplicar el operador de mutación.

El **proceso de autoadaptación** de los parámetros de la estrategia está basado en la existencia de una relación indirecta entre los parámetros de la estrategia y el valor de la función objetivo junto con una diversidad interna suficientemente grande en la población de los padres:

**$\mu$  debe ser suficientemente grande, por ejemplo, Schwefel sugiere  $\mu = 15$  y un ratio  $\lambda/\mu \approx 7$ .**

## 1.3. MODELOS $EE-(\mu, \lambda)$ , $EE-(\mu + \lambda)$

---

*Para el modelo  $EE-(\mu + \lambda)$ , el único cambio está en la selección de la población final:*

*Los  $\mu$  individuos de la nueva población se seleccionan a partir de los  $\mu + \lambda$  individuos formados por los padres y sus descendientes, seleccionando los  $\mu$  mejores.*

*La **desventaja** del modelo  $EE-(\mu + \lambda)$  sobre el anterior es que la estrategia de autoadaptación de los parámetros no es totalmente efectiva porque parámetros no adaptados pueden sobrevivir durante un largo número de iteraciones.*

*La **ventaja** del modelo  $EE-(\mu + \lambda)$  está en que enfatiza el comportamiento de búsqueda local más que el de búsqueda global.*

# MODELO RECIENTE DE EE-(1, $\lambda$ )

## CMA-ES

### Evolution Strategy with Covariance Matrix Adaptation



**Nikolaus Hansen**

Technische Universität Berlin  
Fachgebiet Bionik u. Evolutionstechnik

Microsoft Research-INRIA joint centre, Adaptive Combinatorial Search for e-Sciences  
Centre de recherche INRIA Saclay – Île-de-France

**The CMA Evolution Strategy for Noisy and Global Optimization: Implementations in MATLAB**

## REFERENCIAS

- Hansen, N. and A. Ostermeier (2001). **Completely Derandomized Self-Adaptation in Evolution Strategies.** *Evolutionary Computation*, 9(2), pp. 159-195;
- Hansen, N., S.D. Müller and P. Koumoutsakos (2003). **Reducing the Time Complexity of the Derandomized Evolution Strategy with Covariance Matrix Adaptation (CMA-ES).** *Evolutionary Computation*, 11(1), pp. 1-18;

<http://www.lri.fr/~hansen/>

# 2. PROGRAMACIÓN EVOLUTIVA

---

## 1. INTRODUCCIÓN. PROGRAMACIÓN EVOLUTIVA VERSUS ESTRATEGIAS DE EVOLUCIÓN

## 2. MODELO BÁSICO

*D. Fogel, Evolutionary Computation. Toward a New Philosophy of Machine Intelligence. IEEE Press, 1995.*

## 2.1. INTRODUCCIÓN. RESUMEN RÁPIDO

---

- Desarrollo: USA en los años 60.
- Primer autor: L. Fogel
- Aplicación típica:
  - PE tradicional: aprendizaje automático con máquinas de estado finito.
  - PE contemporáneas: optimización (numérica).
- Características atribuidas:
  - Entorno de trabajo muy abierto: Cualquier representación y mutación son aceptadas.
  - Similitud con las EE (PE contemporánea).
  - Consecuentemente: es difícil decir qué es una PE estándar.
- Característica especial:
  - No utilizan recombinación.
  - Auto-adaptación de los parámetros estándar (PE contemporánea).

## 2.1. INTRODUCCIÓN. PROGRAMACIÓN EVOLUTIVA VERSUS ESTRATEGIAS DE EVOLUCIÓN

---

La **Programación Evolutiva (PE)** es **virtualmente equivalente** en muchos casos a los procedimientos utilizados en las **Estrategias de Evolución (EE)**.

Existen dos diferencias esenciales entre ellas:

1. Las EE utilizan una selección determinística.  
La PE enfatiza la naturaleza probabilística de la selección mediante un torneo estocástico para la supervivencia de cada generación.

## 2.1. INTRODUCCIÓN. PROGRAMACIÓN EVOLUTIVA VERSUS ESTRATEGIAS DE EVOLUCIÓN

---

2. Las EE utilizan estructuras abstractas de codificación consideradas como individuos.

La PE utiliza estructuras abstractas de codificación como analogía a distintas especies.

Las EE pueden utilizar la recombinación entre individuos.

La PE no hace uso de ella, no existe comunicación entre las especies.

## 2.2. MODELO BÁSICO

---

Un **Modelo Básico** de **Programación Evolutiva** incluye los siguientes pasos:

**Inicialización:** Se generan aleatoriamente  $\mu$  individuos para constituir la población inicial.

**Proceso de Mutación:** Se utiliza un operador de mutación (una distribución gaussiana para codificación real) para generar un nuevo individuo  $X'$  vía una perturbación aleatoria de un individuo  $X$  ( $X' = X + N(0, \sigma)$  con codificación real).

## 2.2. MODELO BÁSICO

---

**Proceso de Selección:** Basado en la función objetivo, compara la actuación de la generación vieja con la nueva, individuo a individuo.

Se crea un población conjunta formada por los  $\mu$  padres y los  $\lambda$  hijos. Para cada individuo  $X$  de dicha población se eligen aleatoriamente otros  $q$  individuos y se cuenta cuántos de esos  $q$  tienen peor valor de la función objetivo que  $X$  (ganancia de  $X$ ).

## 2.2. MODELO BÁSICO

---

### Dos posibilidades en la Selección:

**a)** Los  $\mu$  individuos de la población combinada que mayor ganancia tengan pasan a ser los  $\mu$  padres de la nueva generación.

**b)** Los  $\mu$  mejores individuos (de acuerdo el fitness) de esa población combinada pasan a ser los  $\mu$  padres de la nueva generación (**igual que la EE-( $\mu+\lambda$ ), no recomendada**).

## 2.2. MODELO BÁSICO

---

### ALGUNOS COMENTARIOS

El modelo presentado es el básico con los principios fundamentales para utilizar la PE como algoritmo de búsqueda o optimización.

Existen numerosos esfuerzos siguiendo líneas de investigación similares a las Estrategias de Evolución y los Algoritmos Genéticos:

1. La extensión de la auto-adaptación de varianzas.
2. Hibridación con otras técnicas.
3. Paralelización de los algoritmos de PE.

# BIOINFORMÁTICA

## 2013 - 2014

---

### PARTE I. INTRODUCCIÓN

- Tema 1. Computación Basada en Modelos Naturales

### PARTE II. MODELOS BASADOS EN ADAPTACIÓN SOCIAL (Swarm Intelligence)

- Tema 2. Introducción a los Modelos Basados en Adaptación Social
- Tema 3. Optimización Basada en Colonias de Hormigas
- Tema 4. Optimización Basada en Nubes de Partículas (Particle Swarm)

### PARTE III. COMPUTACIÓN EVOLUTIVA

- Tema 5. Introducción a la Computación Evolutiva
- Tema 6. Algoritmos Genéticos I. Conceptos Básicos
- Tema 7. Algoritmos Genéticos II. Diversidad y Convergencia
- Tema 8. Algoritmos Genéticos III. Múltiples Soluciones en Problemas Multimodales
- Tema 9. Estrategias de Evolución y Programación Evolutiva
- **Tema 10. Algoritmos Basados en Evolución Diferencial (Differential Evolution – DE)**
- Tema 11. Modelos de Evolución Basados en Estimación de Distribuciones (EDA)
- Tema 12. Algoritmos Evolutivos para Problemas Multiobjetivo
- Tema 13. Programación Genética
- Tema 14. Modelos Evolutivos de Aprendizaje

### PARTE IV. OTROS MODELOS DE COMPUTACIÓN BIOINSPIRADOS

- Tema 15. Sistemas Inmunológicos Artificiales
- Tema 16. Otros Modelos de Computación Natural/Bioinspirados