Maestría y Doctorado en Ciencia de la Computación

Inteligencia Artificial

Estrategias Evolutivas

Dr. Edward Hinojosa Cárdenas
ehinojosa@unsa.edu.pe
29 de Agosto del 2020

UNSA
UNIVERSIDAD NACIONAL DE SAN AGUSTÍN DE AREQUIPA

Índice

Objetivos del Curso

- **Conocer, comprender e implementar algoritmos evolutivos para resolver problemas complejos.**
- Conocer, comprender e implementar algoritmos de inteligencia de enjambre para resolver problemas complejos.
- Conocer, comprender e implementar algoritmos inmunes artificiales para resolver problemas complejos.
- Conocer, comprender e implementar sistemas basados en lógica difusa para resolver problemas complejos.

Computación Evolutiva

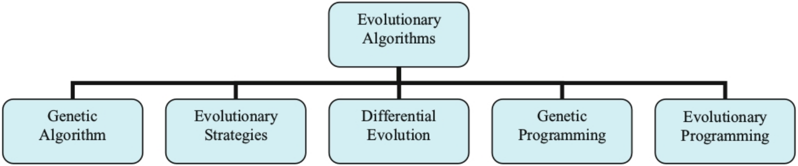

Figure: Algoritmos Evolutivos

Estrategias Evolutivas

- Las Estrategias Evolutivas (EE) fueron propuesta en los años 60 por Rechenberg y Schwefel en Alemania [1], 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 Algoritmos Genéticos, pueden ser descritos básicamente como un Algoritmo Genético 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 EE ejecuta simultáneamente dos tareas:
  - La solución de un problema de optimización específico; y
  - El ajuste del propio algoritmo para resolver el problema.
- Las EE utilizan como operadores genéticos: una mutación auto-adaptativa, cruzamiento y un operador 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 la cual es la base de la próxima generación. Esa es justamente la característica que diferencia los tipos de estrategias evolutivas:

Figure: Estrategias Evolutivas

(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 solo 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 la siguiente fórmula:

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

- 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 $\bar{\sigma}$.

Ejemplo de (1+1) EE

- Por ejemplo, considere el siguiente ejemplo de una (1+1) EE, para minimizar la siguiente función:

$$f(x, y) = (x + 2y - 7)^2 + (2x + y - 5)^2$$

$$-10.0 \leq x \leq 10.0$$

$$-10.0 \leq y \leq 10.0$$

(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:

$$\bar{x_1}^{t+1} = \bar{x_1}^t + N(0; 1.0) = -1.0 + 0.61 = -0.39$$

$$\bar{x_2}^{t+1} = \bar{x_2}^t + N(0; 1.0) = 1.0 - 0.57 = 0.43$$

(1+1) EE

- Ahora, comparamos al padre con el hijo:

$$\textit{Padre}: \; f(x_t) = f(-1.0; 1.0) = 72.0000$$

$$\textit{Hijo}: f(x_{t+1}) = f(-0.39; 0.43) = 71.2634$$

  Dado que: $71.2634 < 72.0000$ El hijo reemplazará al padre en la siguiente generanación.

- Si el descendiente es un descendiente no válido puede intentar otra mutación de todo el individuo o del gen no válido.

(1+1) EE

- El operador de mutación usado en las EE está basado en una distribución de probabilidades normal o Gaussiana de media 0 y desviación estándar σ (para cada desviación estándar), representada por $N(0; \sigma)$ y conocida como distribución normal estándar.

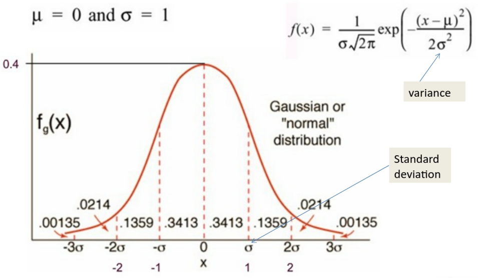

Figure: Distribucion Normal

(1+1) EE

- El área bajo la distribución de probabilidades corresponde a la probabilidad de ocurrencia del valor x. Por lo tanto el área total de una distribución de probabilidad es igual a 1.

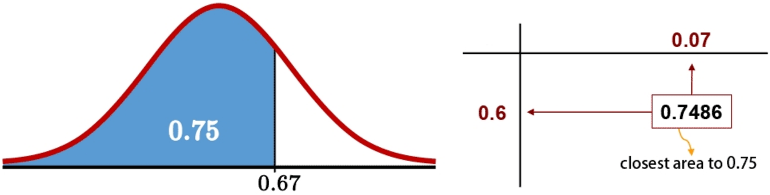

Figure: Área de la Distribucion Normal

`https://homepage.divms.uiowa.edu/~mbognar/applets/normal.html`

(1+1) EE

- 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 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 ε.

(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. Una opción es el método de los trapecios.

(1+1) EE

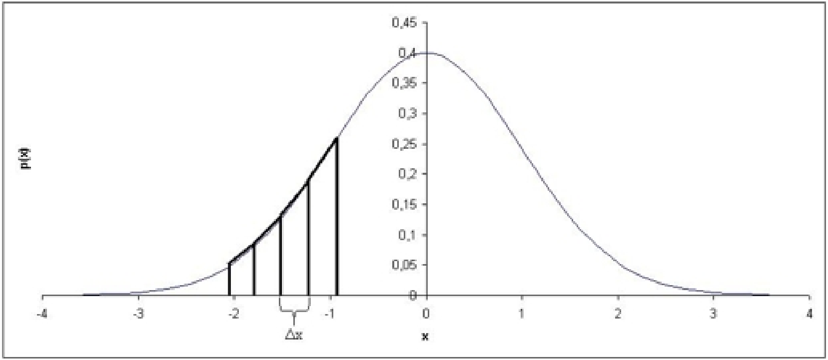

Figure: Cálculo de la Integral en una Distribución Normal

(1+1) EE

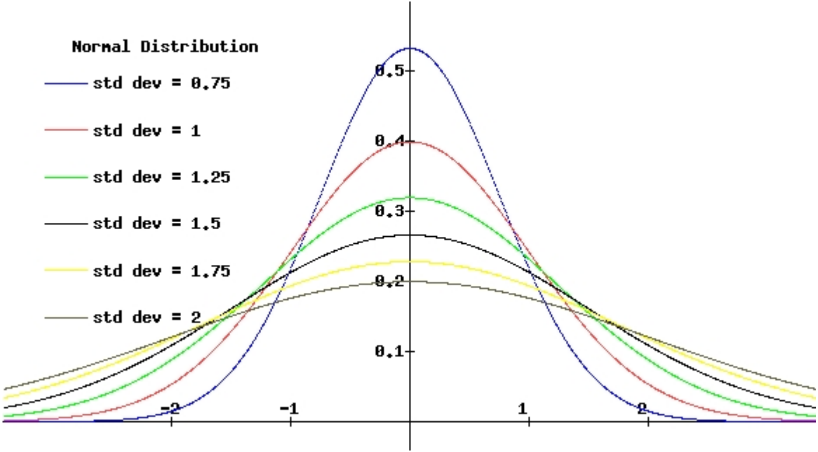

Figure: Cambio Desviación Estándar

(1+1) EE - Auto Adaptación

- Rechenberg [1] 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”, la cual define que se tiene éxito el paso (o cambio) debe aumentar porque la búsqueda puede estar cayendo en una búsqueda local; y el paso debe disminuir si no se tiene éxito indicando que el paso es muy alto.
- Entre mayor la desviación standard mayor el cambio.

Algoritmo (1+1) EE

Algorithm: (1+1)-ES with 1/5 success-rule

1.	Initialize $\boldsymbol{X}_0$, σ_0	
2.	**repeat**	
3.	$\widetilde{\boldsymbol{X}}_n = \boldsymbol{X}_n + \sigma_n \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$	Sample one offspring
4.	**if** $f(\widetilde{\boldsymbol{X}}_n) \leq f(\boldsymbol{X}_n)$ **then**	If f(offsp.) $\leq$ f(parent)
5.	$\boldsymbol{X}_{n+1} = \widetilde{\boldsymbol{X}}_n$	New parent = offsp.
6.	$\sigma_{n+1} = 1.5\, \sigma_n$	Step-size is increased
7.	**else**	If offspring strictly worse
8.	$\boldsymbol{X}_{n+1} = \boldsymbol{X}_n$	New parent = old parent
9.	$\sigma_{n+1} = 1.5^{-1/4}\, \sigma_n$	Step-size is decreased
10.	**until** stopping criteria is met	

(1+1) EE

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

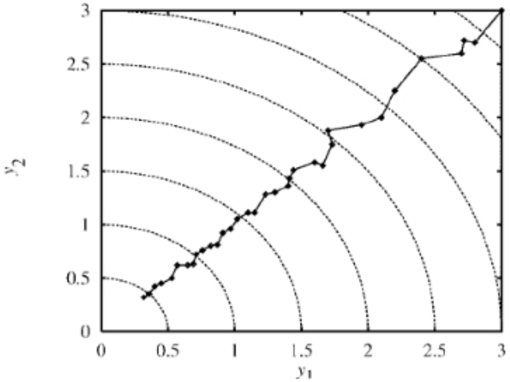

Figure: Hill Climbing

(1+1) EE

- 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.

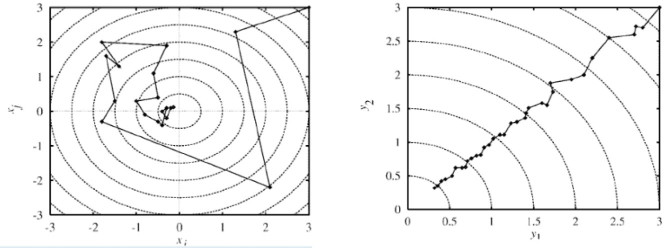

Figure: (1+1) EE vs. Hill Climbing

Ejemplo de Implementación del (1+1) EE

- Minimizar la siguiente función (Considere $\sigma = 0.2$ como valor inicial para todos lo valores de σ):

$$f(x, y) = (x + 2y - 7)^2 + (2x + y - 5)^2$$

$$-10.0 \leq x \leq 10.0$$

$$-10.0 \leq y \leq 10.0$$

Ejemplo de Implementación del (1+1) EE

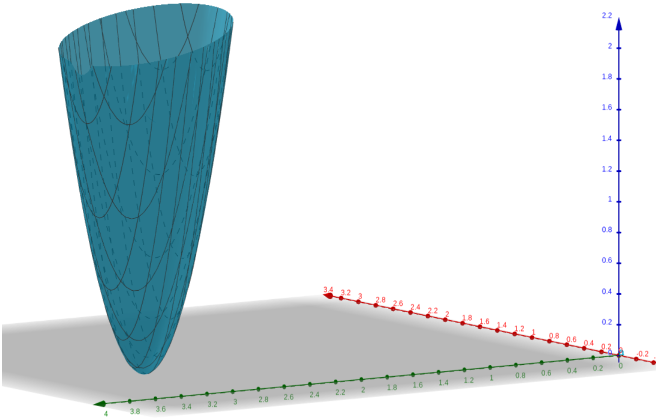

Figure: Función a Optimizar

Estrategias Evolutivas Múltiples

- 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:

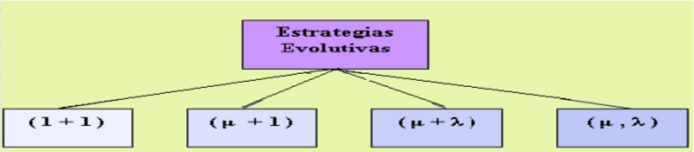

Figure: Estrategias Evolutivas

Estrategias Evolutivas Múltiples

- Se distinguen por utilizar poblaciones con más de un individuo.
- Se puede utilizar el operador de cruzamiento, tenemos que definirlo para vectores de números reales
- No se utiliza la mutación mediante la regla de éxito 1/5,
- No existe una estadística de mejoras para cada individuo.
- Se define un nuevo operador de cruce y un nuevo operador de mutación.

Algoritmo EE Múltiples

Algorithm : Evolution Strategies with Self-adaptation and Recombination $(\mu + \lambda)$

Create an initial population $\{x^1, \ldots, x^\mu\}$ of parent vectors, each x^i being of the form $x^i = (x^i_1, \ldots, x^i_n)$, $i = 1, \ldots, \mu$;

repeat
- **repeat**
  - Select ρ parents from $\{x^i : i = 1, \ldots, \mu\}$;
  - Recombine the ρ parents to form a child, forming both a new solution vector and a new strategy vector;
  - Mutate the strategy vector for the child;
  - Mutate the solution vector of the child using its newly mutated strategy vector;
- **until** λ *children are created*;
- Rank the $\mu + \lambda$ parents and children from best (those producing least error on the training dataset) to worst;
- Select the best μ of these to continue into the next generation;

until *terminating condition*;

Cruzamiento

- Actúa de forma separada en el vector de codificación, y en el vector de desviación estándar o varianzas.
- Vector de codificación. Cada nuevo valor se calcula como la media de los valores de los progenitores (p) en posiciones análogas:

$$\bar{x}' = \left(\frac{1}{2}(x_1^1 + x_1^2), \frac{1}{2}(x_2^1 + x_2^2), \ldots, \frac{1}{2}(x_p^1 + x_p^2)\right)$$

Cruzamiento

- Para las varianzas se realiza una operación similar:

$$\bar{\sigma}' = \left(\sqrt{\sigma_1^1 \times \sigma_1^2}, \sqrt{\sigma_2^1 \times \sigma_2^2}, \ldots, \sqrt{\sigma_p^1 \times \sigma_p^2}\right)$$

- Se le denomina cruce uniforme. No se eligen los genes, se combinan para generar alelos nuevos.

Mutación

- El vector de codificación muta de la siguiente forma siguiendo un esquema Gaussiano descrito por:

$$\sigma' = \sigma \times \exp^{N(0,\triangle\sigma)}$$

$$x' = x + N(0, \sigma')$$

- $\triangle\sigma$ es un parámetro del método. Se recomienda:

$$\triangle\sigma = \frac{1}{\sqrt{2\sqrt{n}}}$$

(μ+1) EE

- Se define el valor entero de μ. Donde ($\mu > 1$).
- En $t = 0$ se genera una población de μ individuos al azar.
- Se seleccionan 2 individuos de la población (los dos individuos son elegidos de acuerdo a su aptitud, por ejemplo ruleta o torneo).
- Se aplica el operador cruzamiento entre los 2 individuos.
- Se muta el nuevo individuo.
- +1 (ó $\lambda = 1$) es el nuevo individuo obtenido del proceso de cruzamiento y mutación.
- Se crea una sola población entre los μ individuos y el nuevo individuo.
- Se elimina al peor individuo de la población.
- El proceso continúa hasta que se satisfaga la condición de terminación.
- El mejor individuo representa la solución.

(μ+λ) EE

- Se definen los valores enteros de μ y λ. Donde ($\mu > \lambda$).
- En $t = 0$ se genera una población de μ individuos al azar.
- Se generan λ individuos a partir de los μ iniciales, empleando cruzamiento (ruleta o torneo para seleccionar 2 individuos).
- Los λ individuos nuevos son mutados.
- Se crea una sola población entre los μ individuos y los λ nuevos individuos.
- Se elimina a los peores λ individuos de la población.
- El proceso continúa hasta que se satisfaga la condición de terminación.
- El mejor individuo representa la solución.

(μ, λ) EE

- Es una modificación de la estrategia ($\mu + \lambda$).
- Se definen los valores enteros de μ y λ. Donde ($\mu \leq \lambda$).
- En $t = 0$ se genera una población de μ individuos al azar.
- Se generan λ individuos a partir de los μ iniciales, empleando cruzamiento (ruleta o torneo para seleccionar 2 individuos).
- Los λ individuos nuevos son mutados.
- Se crea una sola población entre los μ individuos y los λ nuevos individuos.
- Se elimina a los peores λ individuos de la población.
- El proceso continúa hasta que se satisfaga la condición de terminación.
- El mejor individuo representa la solución.

Ejemplo de Implementación EE Múltiples

- Minimizar la siguiente función (Considere $\sigma = 0.2$ como valor inicial para todos lo valores de σ):

$$f(x, y) = (x + 2y - 7)^2 + (2x + y - 5)^2$$

$$-10.0 \leq x \leq 10.0$$

$$-10.0 \leq y \leq 10.0$$

Ejemplo de Implementación del EE Múltiples

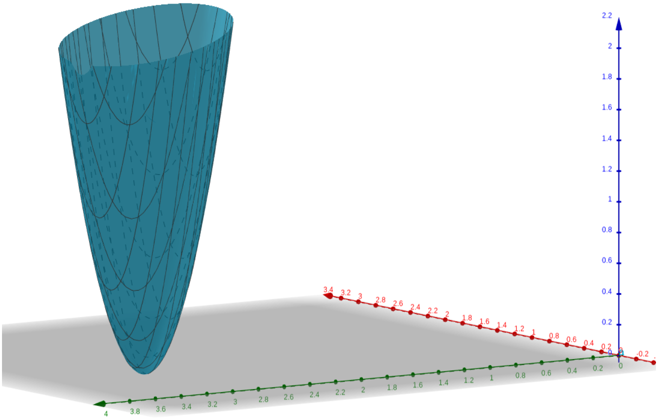

Figure: Función a Optimizar

¡GRACIAS!

Bibliografía

[1] I. Rechenberg, B. Toms, and R. A. Establishment. *Cybernetic Solution Path of an Experimental Problem:*. Library translation / Royal Aircraft Establishment. Ministry of Aviation, 1965.