

# Introducción a la Computación Evolutiva

Dr. Carlos Artemio Coello Coello

Tarea No. 1

13 de mayo de 2022

## 1 Métodos de optimización

El método de operación evolutiva (EVOP) fue propuesto por el estadístico inglés George E.P. Box en 1957. Este algoritmo requiere  $(2^N + 1)$  puntos ( $N$  es el número de variables del problema), de los cuales  $2^N$  son las esquinas de un hipercubo, con centro en el otro punto. Se comparan entre sí los  $(2^N + 1)$  valores de la función y se identifica el mejor punto entre ellos. En la siguiente iteración, se forma otro hipercubo alrededor del mejor punto. Si en cualquier iteración no se mejora el punto actual, entonces se reduce el tamaño del hipercubo. Este proceso continúa hasta que el hipercubo se vuelve muy pequeño. El algoritmo es el siguiente:

Paso 1:	Elegir un punto inicial $x^{(0)}$ y parámetros de reducción de tamaño $\Delta_i$ para todas las variables de decisión, $i = 1, 2, \dots, N$ . Elegir una tolerancia $\epsilon$ . Hacer $\bar{x} = x^{(0)}$
Paso 2:	IF $\ \Delta\  < \epsilon$ THEN Terminar ELSE crear $2^N$ puntos sumando y restando $\Delta_i/2$ de cada variable en el punto $\bar{x}$
Paso 3:	Calcular los valores de la función en los $(2^N + 1)$ puntos. Encontrar el punto con el menor valor de la función. Hacer que el punto mínimo sea $\bar{x}$ .
Paso 4:	IF $\bar{x} = x^{(0)}$ , reducir los parámetros de tamaño $\Delta_i = \Delta_i/2$ . GOTO Paso 2. ELSE $x^{(0)} = \bar{x}$ . GOTO Paso 2.

Nótese que en el algoritmo,  $x^{(0)}$  siempre contiene al mejor punto. Por tanto, al final de su ejecución,  $x^{(0)}$  se vuelve nuestra mejor aproximación al óptimo.

Implemente en C/C++ este algoritmo. Las entradas del programa deben ser  $x^{(0)}$ ,  $\Delta_i$  ( $i = 1, 2, \dots, N$  y  $N$  es el número de variables de decisión del problema) y la tolerancia  $\epsilon$ . El código fuente de su programa deberá incluirse en el reporte de su tarea y deberá incluirse también en un CD (el programa deberá ser capaz de compilar y ejecutarse correctamente usando GNU C/C++ bajo ambiente Linux). La salida del programa

deberá ser el valor de  $x^*$ , y el valor correspondiente de  $f(x^*)$  a cada iteración, así como el obtenido al final.

(20 puntos) Resuelva el problema siguiente usando su implementación de EVOP:

$$\text{Min } f(x_1, x_2) = 50(x_2 - x_1^2)^2 + (2 - x_1)^2 \quad (1)$$

Utilice:  $x^{(0)} = [0, 0]^T$  y  $\epsilon = 1 \times 10^{-4}$ . Defina los valores de  $\Delta_i$  que considere adecuados. Discuta brevemente en su reporte el por qué de su elección. Recuerde que el objetivo es que el método logre llegar al óptimo del problema. En su reporte debe incluirse el valor final de  $x^*$  y el de  $f(x^*)$  correspondiente. Indique también el número de iteraciones que realizó su programa.

## 2 Estrategias evolutivas

El objetivo de este problema es que se familiarice con el funcionamiento de las estrategias evolutivas. Para ello, tendrá que implementar la denominada  $(1 + 1) - EE$ , cuyo algoritmo es el siguiente:

```

t = contador de generaciones, n = número de variables
Gmax = número máximo de generaciones
t ← 0
Inicializar variables  $\bar{x}$ ,
Evaluar  $f(\bar{x})$ 
while (t ≤ Gmax) do
    inicializar semilla de aleatorios
    mutar el vector  $x_i$  usando:
         $x'_i = x_i + \sigma[t] \times N_i(0, 1) \quad \forall i \in n$ 
    Evaluar  $f(\bar{x}')$ 
    Comparar  $\bar{x}$  con  $\bar{x}'$  y seleccionar el mejor
    Imprimir en un archivo los resultados
    t = t+1
    if (t mod n == 0) then
        
$$\sigma[t] = \begin{cases} \sigma[t - n]/c & \text{if } p_s > 1/5 \\ \sigma[t - c] \cdot c & \text{if } p_s < 1/5 \\ \sigma[t - n] & \text{if } p_s = 1/5 \end{cases}$$

        else  $\sigma[t] = \sigma[t - 1]$ 

```

Algunos puntos importantes que deben observarse en este programa son los siguientes:

- El valor de la constante  $c$  oscila entre 0.817 y 1.0, aunque muchos suelen establecerlo en 0.817 ó 0.85. Use el valor que desee, pero indíquelo claramente en su reporte y en su programa.
- $p_s$  es la frecuencia de éxito de las mutaciones. Para calcularla, se registrará como exitosa aquella mutación en la que el hijo reemplace a su padre. La actualización de  $p_s$  se efectuará cada  $10 \cdot n$  iteraciones.
- Usaremos  $\sigma[0] = 3.0$ .
- $Gmax$  será proporcionada por el usuario (puede usar cualquier valor entero  $\geq 10$ ).
- La semilla de aleatorios se inicializará con  $time(0)$ , que es una función en C que devuelve un entero con base en el reloj interno de la computadora.
- $N(0, 1)$  es una función que genera números aleatorios Gaussianos (distribución normal) con media cero y desviación estándar uno. Esta función está disponible en la página web del curso.
- Cuidar que los valores de las variables no se salgan del rango especificado. Si eso ocurre, deberá contarse con algún mecanismo que vuelva a generar el valor de la variable o que lo ajuste al rango deseado. Explique en su reporte qué mecanismo fue el que utilizó.

Escriba un programa en C/C++ que implemente el algoritmo arriba indicado y utilizarlo para **minimizar**:

$$f(x_1, x_2) = \left(4 - 2.1x_1^2 + \frac{x_1^4}{3}\right) x_1^2 + x_1x_2 + (-4 + 4x_2^2) x_2^2 \quad (2)$$

donde  $-3 \leq x_1 \leq 3$ ,  $-2 \leq x_2 \leq 2$ .

Se efectuarán varias corridas de prueba y se entregará un reporte que contenga lo siguiente:

1. **(80 puntos)** El código fuente del programa, con comentarios. Deberá entregarse una versión impresa y una en CD, incluyendo los archivos de TODAS las corridas de ejemplo efectuadas. El programa deberá ser capaz de compilarse y ejecutarse bajo Linux (usando **gcc** o **g++**) y, de ser necesario, podrá requerirse al autor que demuestre el uso del programa personalmente al instructor.
2. **(10 puntos)** Una gráfica de la función a optimizarse dentro de los rangos permisibles para las variables. La gráfica deberá estar en 3 dimensiones.
3. **(30 puntos)** Las estadísticas de al menos 20 corridas diferentes en las que se use el mismo valor de  $Gmax$ , pero diferentes valores para la semilla de aleatorios y para las variables de inicio  $\bar{x}$ . Las estadísticas deben incluir la media, la varianza, la desviación estándar, el mejor y el peor valor de  $f(\bar{x})$ . Incluya también la salida producida por su programa para una sola corrida (puede abreviarla en caso de que el texto sea muy largo), pero ponga TODAS las demás corridas en el CD.

4. (**Bonificación : 10 puntos**) Hay dos puntos que hacen que esta función llegue al óptimo. Muestre 2 corridas de la estrategia evolutiva que obtenga estas 2 soluciones. Se darán 5 puntos de bonificación por cada una de estas soluciones óptimas que se obtengan. No olvide incluir todos los parámetros utilizados. Los valores iniciales de la estrategia evolutiva deberán ser distintos de los que producen estas soluciones óptimas.

## Antecedentes Biológicos

1. (**20 puntos**) Investigue en qué consiste el **equilibrio acentuado** (*punctuated equilibrium*, en inglés) e indique si considera que se opone a los preceptos del Neo-Darwinismo o no. Fundamente bien sus argumentos.
2. (**20 puntos**) Investigue en qué consisten las **mutaciones neutrales** (*neutral mutations*, en inglés), e indique si considera que se opone o no a los preceptos del Neo-Darwinismo. Fundamente bien sus argumentos.

Responda a las preguntas anteriores de forma concisa y directa, sin recurrir a explicaciones elípticas o barrocas. Cite toda la información utilizada (artículos, libros, páginas web) y explique cualquier concepto complementario que pueda ser necesario para fundamentar sus argumentos.

**Fecha de entrega:** *Viernes 27 de mayo a las 12:00hrs.* Toda tarea entregada tarde será penalizada con 10% (sobre la calificación obtenida) por cada periodo de 24 horas que se retrase su entrega.