

Desarrollo de una herramienta software para la resolución de problemas de optimización con algoritmos genéticos

1. PROBLEMAS DE OPTIMIZACIÓN A RESOLVER

1.1 Caso general

El AG que se programe en C se utilizará para resolver problemas de optimización como el que se describe a continuación.

Sea define el *vector de parámetros* θ :

$$\theta = [x_1, \dots, x_n]$$

Donde los elementos x_i son números reales acotados dentro de un cierto espacio S .

$$S = \{x_1 \in [x_1^{\min}, x_1^{\max}], x_2 \in [x_2^{\min}, x_2^{\max}], \dots, x_n \in [x_n^{\min}, x_n^{\max}]\}$$

Es decir el valor que puede tomar cada elemento x_i se encuentra acotado dentro de un cierto valor mínimo x_i^{\min} y de un cierto valor máximo x_i^{\max} .

Sea la función de coste $J: R^n \rightarrow R$

$$J = f(\theta)$$

Se verifica la siguiente condición que la función $J > 0$ para cualquier θ considerado.

El problema a resolver es el de obtener el vector de parámetros óptimo $\theta_{\text{opt}} \in S$ que genera el valor máximo J_{opt} de la función de coste J .

$$J_{\text{opt}} = J(\theta_{\text{opt}}) = \max_{\theta \in S} (J(\theta))$$

1.2 Casos particulares

A continuación se listan algunas funciones, con las que se debe comprobar el buen funcionamiento del algoritmo genético programado, el alumno/a puede aplicar también el AG para la resolución de cualquier otro problema de optimización que considere oportuno.

Caso 1:

$$f(x_1, x_2) = 21.5 + x_1 \cdot \sin(4 \cdot \pi \cdot x_1) + x_2 \cdot \sin(4 \cdot \pi \cdot x_2)$$

$$S = \{x_1 \in [-3.0, 12.1], x_2 \in [4.1, 5.8]\}$$

Caso 2:

$$f(x_1, x_2) = 100 - (x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_5^2 + x_6^2)$$

$$S = \left\{ \begin{array}{l} x_1 \in [-3.0, 5.1], x_2 \in [2.1, 7.8], x_3 \in [-10.1, 20.3], \\ x_4 \in [-3.3, 4.2], x_5 \in [-15.3, 70.1], x_6 \in [-0.25, 0.35] \end{array} \right\}$$

Caso 3: *Función de Ackley*

$$f(x_1, x_2) = -c_1 \cdot e^{\left(-c_2 \sqrt{\frac{1}{2}(x_1^2 + x_2^2)}\right)} - e^{\frac{1}{2}(\cos(c_3 x_1) + \cos(c_3 x_2))} + c_1 + e$$

$$c_1 = 20 \quad c_2 = 0.2 \quad c_3 = 2\pi \quad e = 2.71282$$

$$S = \{-5 \leq x_1 \leq 5 \quad -5 \leq x_2 \leq 5\}$$

Caso 4: *Función de Schaffer*

$$f(x_1, x_2) = 100 - [(x_1^2 + x_2^2)^{0.25} \cdot [\sin^2(50 \cdot (x_1^2 + x_2^2)^{0.1}) + 1.0]]$$

$$S = \{x_1 \in [-100, 100], x_2 \in [-100, 100]\}$$

Caso 5:

$$f(x_1, x_2) = 500 - \left[a \cdot (x_2 - b \cdot x_1^2 + c \cdot x_1 - d)^2 + e \cdot (1 - f) \cdot \cos(x_1) + e \right]$$

$$a = 1 \quad b = \frac{5.1}{4 \cdot \pi^2} \quad c = \frac{5}{\pi} \quad d = 6 \quad e = 10 \quad f = \frac{1}{8\pi}$$

$$S = \{-5 \leq x_1 \leq 10 \quad 0 \leq x_2 \leq 15\}$$

Caso 6:

$$f(x, y) = 0.5 - \frac{\sin^2 \sqrt{x^2 + y^2} - 0.5}{(1 + 0.001 \cdot (x^2 + y^2))^2}$$

$$S = \{x, y \in [-100, 100]\}$$

2. CARACTERÍSTICAS DE UN AG BÁSICO PARA RESOLVER PROBLEMAS DE OPTIMIZACIÓN

A continuación se describen las características de un algoritmo genético (AG) básico, para que sirvan de referencia al alumno/a en la realización de su proyecto fin de carrera (PFC). Por supuesto el alumno/a puede realizar las modificaciones que considere oportunas con el fin de mejorar dicho AG

2.1 Codificación del cromosoma

El tipo de *cromosoma* que se va a considerar para la resolución de este problema es el definido por el vector de parámetros θ . La codificación que se utilizará para los elementos o *genes* x_i del cromosoma i es en coma flotante, es decir, se trabajará directamente considerando que son números reales sin codificarlos en binario.

Es importante recordar que los genes del cromosomas deben pertenecer siempre al espacio S .

2.2 Tamaño de la población

Cada población tendrá un tamaño *pop_size*, es decir constará de un conjunto de *pop_size* cromosomas θ_i $i=1, \dots, \text{pop_size}$.

$$\text{población} = [\theta_1, \dots, \theta_{\text{pop_size}}]^T$$

2.3 Diagrama de flujo de un AG básico

En la Figura se muestra el esquema de un AG básico propuesto por *Michalewicz*

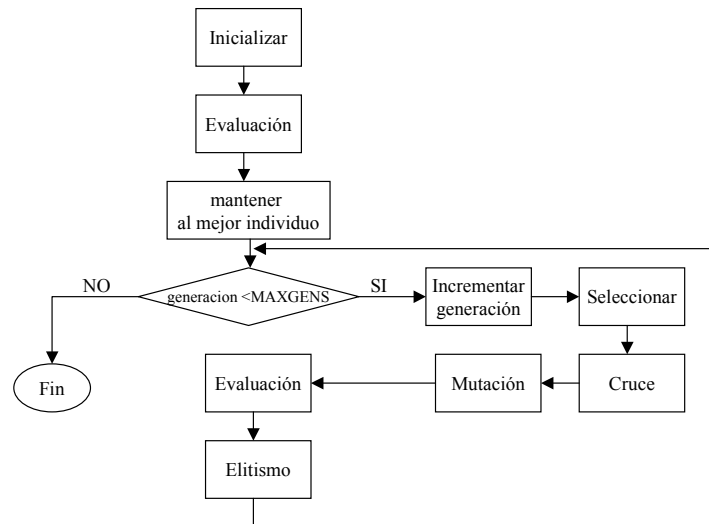


Figura: Esquema de un AG básico.

Paso 1: Inicialización

El primer paso del AG es generar de forma aleatoria una población inicial de individuos e inicializar el contador de generaciones a cero (*generaciones*=0). Es importante darse cuenta que entre otras funciones será necesario programar una función que dados los valores mínimo x_i^{\min} y máximo x_i^{\max} que puede tomar un gen de un cromosoma θ_i de la población genere un valor aleatorio para dicho gen x_i dentro de su rango correspondiente.

Paso 2: Evaluación

Se evalúa cada individuo o cromosoma θ_i $i=1, \dots, \text{pop_size}$ de la población inicial sobre la función de coste J del problema. Obviamente se debe programar en C la función de coste J , de tal forma que dados los genes de un cromosoma (elementos x_i) genere el valor J_i asociado a dicho cromosoma θ_i .

Paso 3: Mantener el mejor individuo

Se debe identificar y mantener para la próxima generación al individuo o cromosoma $\theta_{\text{old_best}}$ que presente el mayor valor de la función de coste $J_{\text{old_best}}$ de la generación actual. Dicho cromosoma se debe almacenar adicionalmente en la posición $\text{pop_size}+1$. También se debe almacenar el valor $J_{\text{old_best}}$

Paso 4: *Bucle* de ejecución

A continuación el algoritmo ejecuta un bucle del que no sale mientras el contador de generaciones sea inferior al número máximo de generaciones fijado (*MAXGENS*) .

Paso 4.1: *Incrementa el contador de generaciones.*

Paso 4.2: *Aplicar el operador de selección*

A continuación se aplica el operador de selección que utiliza el *método de la ruleta*. Que consta de los siguientes pasos:

- 1) Calcular la suma total F de los valores de la función de coste J_i asociada a cada cromosoma θ_i $i=1,\dots,\text{pop_size}$ de la población.

$$F = \sum_{i=1}^{\text{pop_size}} J_i$$

- 2) Calcular la probabilidad de selección p_i para cada cromosoma θ_i

$$p_i = \frac{J_i}{F}$$

- 3) Calcular la probabilidad acumulada q_i para cada cromosoma θ_i .

$$q_i = \sum_{k=1}^{\text{pop_size}} p_k$$

- 4) Se gira la ruleta pop_sizes veces. Cada vez que se gira la ruleta hay que hacer lo siguiente:

- Se genera un número real aleatorio r dentro del rango $[0,1]$.
- Si $r < q_1$ se selecciona el cromosoma θ_1 , sino se selecciona el cromosoma i -simo θ_i ($2 \leq i \leq \text{pop_size}$) tal que $q_{i-1} < r \leq q_i$ para una nueva población

Obsérvese que tras haber girado la ruleta pop_sizes se habrán seleccionado pop_sizes cromosomas para formar parte de la nueva población. Con el fin de asegurar que el mejor miembro de la

población anterior θ_{old_best} (que recordemos se había almacenado en el paso 3) sobrevive éste se almacena en la posición $pop_size + 1$ de la nueva población.

Paso 4.3: Aplicar el operador de cruce en un solo punto

Recuérdese que la *probabilidad de cruce* se denomina p_c . Este operador se puede implementar de la siguiente forma:

- 1) Se deben seleccionar miembros de la población para reproducirse, para ello por cada cromosoma θ_i $i=1,\dots, pop_size$ se genera un número real aleatorio r comprendido en el rango $[0,1]$. Si $r < p_c$ entonces el cromosoma θ_i es seleccionado para reproducirse.

Por ejemplo supóngase que se tiene una población de $pop_size=6$ individuos o cromosomas: $[\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6, \theta_{old_best}]$ se genera un número aleatorio r para cada uno de ellos: $[0.1234, 0.5623, 0.8956, 0.7584, 0.0256, 0.9106]$ excepto para el individuo θ_{old_best} , si la probabilidad de cruce es $p_c=0.8$ entonces los cromosomas seleccionados para reproducirse son $[\theta_1, \theta_2, \theta_4, \theta_5]$.

- 2) Se procede entonces a aplicar la operación de cruce en un sólo punto entre parejas de individuos seleccionados, el punto de cruce se elige aleatoriamente.

Siguiendo con el ejemplo anterior, los cromosomas seleccionados han sido $[\theta_1, \theta_2, \theta_4, \theta_5]$, se forman dos parejas para reproducirse $\{\theta_1, \theta_2\}$ y $\{\theta_4, \theta_5\}$. Supóngase que cada cromosoma consta de 10 genes.

Por ejemplo para la pareja $\{\theta_1, \theta_2\}$, se selecciona la posición de cruce es decir hay que generar un número entero positivo aleatorio pos comprendido entre $[1,9]$. Supóngase que ha salido $pos=3$ entonces el cruce se realiza de la siguiente forma:

PADRES:

$$\theta_1 = [a_1, a_2, a_3, | a_4, a_5, a_6, a_7, a_8, a_9, a_{10}]$$

$$\theta_2 = [b_1, b_2, b_3, | b_4, b_5, b_6, b_7, b_8, b_9, b_{10}]$$

HIJOS:

$$\theta'_1 = [a_1, a_2, a_3, | b_4, b_5, b_6, b_7, b_8, b_9, b_{10}]$$

$$\theta'_2 = [b_1, b_2, b_3, | a_4, a_5, a_6, a_7, a_8, a_9, a_{10}]$$

De la misma forma se procede con la pareja $\{\theta_4, \theta_5\}$, en ese caso supóngase que $pos=2$ se generarían dos hijos $\{\theta'_4, \theta'_5\}$.

3) Se sustituye en la población los padres por los hijos. Es decir, siguiendo con el ejemplo, la población estaría ahora formada por:

$$[\theta'_1, \theta'_2, \theta_3, \theta'_4, \theta'_5, \theta_6, \theta_{old_best}]$$

Paso 4.4: *Aplicar el operador de mutación uniforme*

Para cada cromosoma de la población obtenida en el paso 4.3 se aplica el operador de mutación uniforme, cuyo comportamiento viene definido por el valor de la probabilidad de mutación p_m . Este operador se implementa de la siguiente forma:

- 1) Dado un cromosoma $\theta_i = [x_1, \dots, x_n]$, se genera para cada gen x_k del cromosoma un número real aleatorio r comprendido entre $[0, 1]$.
- 2) Si $r < p_m$ entonces el cromosoma x_k debe ser mutado, es decir, debe ser reemplazado por un número real aleatorio $x_k^{ale} \in [x_k^{\min}, x_k^{\max}]$.

Paso 4.5: *Evaluación de la población actual*

Se realiza de igual forma que en el paso 2, recuérdese que solo deben evaluarse los cromosomas θ_i $i=1, \dots, pop_size$.

Paso 4.7: *Elitismo*

La técnica de elitismo, se implementa de la siguiente forma:

- 1) Se localiza la posición del mejor cromosoma θ_{best} de la población actual, es decir, aquel que presenta el mayor valor de la función de coste J , que se denominará por J_{best} .

2) Se localiza la posición del peor cromosoma θ_{worst} de la población actual, es decir, aquel que presenta el menor valor de la función de coste J , que se denominará por J_{worst} .

3) Se compara el valor J_{best} con el valor J_{oldbest} del mejor cromosoma de la generación anterior que recordase se encuentra almacenado en la posición $\text{pop_size}+1$.

- Si $J_{\text{best}} \geq J_{\text{old_best}}$, se sustituye el mejor cromosoma de la generación anterior por el mejor cromosoma de la generación actual. Es decir, $\theta_{\text{old_best}} = \theta_{\text{best}}$ y $J_{\text{old_best}} = J_{\text{best}}$.

- En caso contrario hay que sustituir el peor cromosoma de la generación actual por el mejor cromosoma de la generación anterior. Es decir, $\theta_{\text{worst}} = \theta_{\text{old_best}}$ y $J_{\text{worst}} = J_{\text{old_best}}$.

Se vuelve entonces al comienzo del bucle (paso 4).

Cuando se sale de este bucle el algoritmo llega a su fin y nos devuelve el mejor individuo o cromosoma encontrado a lo largo de todas las generaciones, así como el valor de la función de coste que produce dicho individuo.

Es importante tener en cuenta que a la hora de resolver un problema con un AG, este debe ser corrido varias veces (*eras*). La solución se tomará como la mejor obtenida en todas las *eras*.

3. INTERACTIVIDAD DEL USUARIO CON EL PROGRAMA

3.1 Entrada de datos

El alumno/a tiene libertad para implementar la entrada de datos como considere más oportuno, aunque se debe intentar que se realice de forma sencilla y que presente una estética agradable al usuario. A continuación se listan los parámetros que el usuario debe introducir para poder ejecutarse el algoritmo descrito en la sección 2.

- La *función matemática* J se debe introducir como un string o en un fichero ASCII.
- El *espacio o rango* S de valores para las variables de la función J (rango de valores para los cromosomas).

- *Número de eras (eras)*. Es decir, el número de veces seguidas que se ejecutará el AG. Este parámetro es un número entero que estará comprendido entre 1 y 100.
- *Número máximo de generaciones (maxgens)*. Es decir, el número máximo de generaciones que se deben generar para dar por finalizada una era del AG. Este parámetro es un número entero que estará comprendido entre 1 y 1000.
- *Tamaño de la población (pop_size)*. Es decir, el número de cromosomas de que consta cada población. Este parámetro es un número entero que estará comprendido entre 1 y 1000.
- Probabilidad de cruce (pc). Este parámetro es un número real que estará comprendido entre 0 y 1.
- Probabilidad de mutación (pm). Este parámetro es un número real que estará comprendido entre 0 y 1.

3.2 Salida de datos

El alumno/a tiene libertad para implementar la salida de datos como considere más oportuno, aunque se debe intentar que se realice de forma sencilla y que presente una estética agradable al usuario. A continuación se listan algunos parámetros (opcionales) que el programa puede presentar por pantalla durante su tiempo de ejecución:

- Tiempo de ejecución del programa
- Estima del tiempo de ejecución que resta.
- Número de Era actual.
 - Número de Generación actual.
 - El mejor valor obtenido de la función de coste J en la generación pasada.
 - El valor medio de \bar{J} en la generación pasada
 - Desviación estándar de J en la generación pasada.
 - El mejor cromosoma obtenida en la generación pasada.
 - Mejora porcentual obtenido en el mejor valor de J en la generación pasada.

- Mejor valor de J a lo largo de las distintas eras.
- Valor medio del mejor valor de J obtenido en las distintas eras.
- Mejor cromosoma obtenido a lo largo de las distintas eras.

Al finalizar el programa se debe haber almacenado en un fichero ASCII el valor del mejor valor de la función de coste J obtenido en cada generación de cada era. Además se debe almacenar el mejor cromosoma obtenido en cada era.

El alumno debe ponderar la viabilidad de que el programa vaya realizando representaciones gráficas de los datos anteriormente citados durante la ejecución del programa y/o al finalizar el mismo.

3.3 Terminación del programa

Se deben implementar los mecanismos adecuados que permitan al usuario finalizar el programa cuando considere oportuno pudiendo disponer de la información obtenida hasta dicho momento.