

# Introducción a la Computación Evolutiva

Dr. Carlos Artemio Coello Coello

Tarea No. 2

28 de mayo de 2009

## Implementación de un algoritmo genético simple

1. Escribir un programa en C/C++ que implemente un algoritmo genético simple (representación binaria, cruce de un punto, mutación uniforme y selección proporcional de ruleta, tal y como vimos en clase) y utilizarlo para **minimizar**

$$f(x_1, x_2, x_3, x_4) = [10(x_2 - x_1^2)]^2 + (1 - x_1)^2 + 90(x_4 - x_3^2)^2 + (1 - x_3)^2 + 10(x_2 + x_4 - 2)^2 + 0.1(x_2 - x_4)^2 \quad (1)$$

donde:  $-20.0 \leq x_j \leq 20.0$  ( $j = 1, \dots, 4$ ) usando números reales con una precisión de diez cifras después del punto decimal. Lo que debe entregarse es lo siguiente:

1. **(50 puntos)** El código fuente en C/C++ del programa, con comentarios. Deberá entregarse una versión impresa y una en diskette (o CD), incluyendo los archivos de TODAS las corridas de ejemplo efectuadas (estos archivos pueden abreviarse en caso de ser muy grandes). El programa deberá ser compilable y ejecutable usando GNU C/C++ bajo Linux y, de ser necesario, el autor deberá demostrar el programa personalmente al instructor. En cada generación deberá retenerse a la mejor solución (**elitismo**), y deberán imprimirse al menos las siguientes estadísticas **por generación**: media de aptitud de la población, aptitud máxima y mínima, número de cruces efectuadas, número de mutaciones efectuadas y cadena binaria correspondiente al mejor individuo (el que tenga la aptitud más alta). Adicionalmente, deberá imprimirse el mejor individuo global (es decir, contando todas las generaciones), su aptitud y su cadena binaria.
2. **(5 puntos)** Un análisis breve que indique cómo se efectúa el mapeo de las variables de binario a decimal y viceversa, incluyendo detalles de la codificación utilizada (por ejemplo, cómo se derivó el número de bits a utilizarse en las cadenas y qué tipo de decodificación se empleó).
3. **(5 puntos)** Estime el tamaño “intrínseco” del espacio de búsqueda para este problema. Es decir, dada la longitud de la cadena binaria que representa los valores posibles de  $x_j$  ( $j = 1, \dots, 4$ ), estime cuántas posibles soluciones pueden generarse. Dada una corrida promedio de las realizadas (con los parámetros que Ud. guste), indique qué porcentaje aproximado del espacio de búsqueda ha explorado el algoritmo genético para encontrar la solución.

4. **(10 puntos)** Una corrida con parámetros mínimos (una población de sólo 6 individuos, corriendo sólo 2 generaciones), a fin de ilustrar que la cruce, mutación y selección se efectuaron correctamente. En la corrida se mostrarán las cadenas binarias correspondientes a cada individuo, así como sus valores decodificados (es decir, los valores de  $x_j$  ( $j = 1, \dots, 4$ ) en cada caso), sus valores de aptitud, los puntos de cruce elegidos, los padres y los hijos producidos en cada caso, y los efectos producidos por la mutación (usar un porcentaje relativamente alto a fin de que se produzca mutación con una población pequeña). Asimismo, se deberá ilustrar el proceso de selección paso por paso, de manera análoga a como lo hicimos en clase (indicando los valores esperados para cada individuo, la aptitud promedio de la población y el número de copias de cada individuo tras el proceso de selección). Es importante hacer notar que todo este proceso SÓLO se incluirá en esta corrida con parámetros mínimos, ya que en las corridas normales se reportarán únicamente los datos pedidos en el primer punto.
5. **(30 puntos)** Una corrida de ejemplo (salida generada por el programa), y los resultados promediados de AL MENOS 20 corridas independientes en las que se usen los mismos parámetros (porcentaje de cruce y mutación, tamaño de la población y máximo número de generaciones) pero diferente semilla para generar números aleatorios. En el reporte deberá aparecer la mejor solución obtenida, la media de aptitud, la peor solución obtenida y la desviación estándar de las 20 corridas efectuadas. Deberán mostrarse los valores de  $x_j$  ( $j = 1, \dots, 4$ ) y de  $f(x_1, x_2)$  en cada caso. Asimismo, se incluirá la **gráfica de convergencia** correspondiente a la solución que esté en la mediana de los resultados obtenidos (de las 20 corridas). En esta gráfica se mostrará en el eje de las  $x$  el número de generaciones (de cero al máximo permisible) y en el eje de las  $y$  se mostrará la aptitud promedio de cada generación. TODAS las corridas efectuadas deberán aparecer en archivos diferentes en un diskette (o CD), además del código fuente del programa.
6. **(Bonificación : 10 puntos)** Se bonificará al que presente el conjunto de 20 corridas con el menor número total de evaluaciones de la función de aptitud que obtenga la solución óptima a este problema. Estas corridas se deberán presentar en diskette (o CD), pero la tabla que resume los resultados deberá incluirse en el reporte.

En la página web del curso se encuentra un programa (en GNU C para Unix) que implementa un algoritmo genético simple con representación binaria, selección mediante torneo, cruce de 2 puntos y mutación uniforme (en una variante que no vimos en clase). Este programa optimiza la función  $f(x) = x^2$ , aunque imprime toda la información de cada generación, incluyendo las cadenas correspondientes a cada individuo. Esta información (salvo la mencionada en el primer punto de la lista anterior) es redundante y no requiere incluirse en las corridas efectuadas por su programa. Es importante hacer ver también que puede desarrollar una implementación por cuenta propia, sin utilizar ningún elemento del código proporcionado.

**Fecha de entrega:** *Jueves 4 de junio 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.