

Introducción a la Computación Evolutiva

Dr. Carlos Artemio Coello Coello

Tarea No. 5

29 de junio de 2018

Manejo de restricciones

1. El objetivo de esta tarea es que implemente un algoritmo genético con codificación real y que experimente con una técnica para el manejo de restricciones.

A fin de dar mayor libertad a su creatividad, no se requerirá que implemente ningún tipo de técnica de manejo de restricciones en particular, de las que vimos en clase, sino que se le requiere que diseñe una que Ud. mismo proponga. Claro que es posible basarse en las técnicas que vimos en clase. Sin embargo, la técnica que proponga deberá distinguirse en algún aspecto fácilmente identificable de las que vimos en clase.

Lo que debe entregarse es lo siguiente:

1. **(60 puntos)** El código fuente en C/C++ de un programa que implemente un algoritmo genético con codificación real, utilizando cualquiera de las técnicas de cruce y mutación que vimos en clase para esta representación.¹ En los comentarios de su programa y en el reporte correspondiente deberá indicar claramente qué técnicas fueron las que decidió implementar. El código fuente deberá tener suficientes comentarios como para hacerlo fácilmente comprensible y deberá incluirse en un CD en donde también deberá ir el ejecutable correspondiente, así como TODAS las corridas que haya efectuado. La técnica de selección y los parámetros en general que utilice quedan a su libre elección, pero deberán mencionarse claramente en los comentarios del programa y en el reporte correspondiente. Se deberá mencionar en qué plataforma se probó el programa (incluyendo el compilador de C/C++ utilizado) y, de ser necesario, el autor deberá demostrar

¹Alternativamente, es válido utilizar operadores reportados en la literatura especializada, aunque no los hayamos visto en clase (en caso de optar por esta opción, es vital reportar la(s) fuente(s) consultada(s). Ver por ejemplo (estos artículos se encuentran disponibles en la página del curso):

F. Herrera, M. Lozano and J.L. Verdegay, "Tackling Real-Coded Genetic Algorithms: Operators and tools for the Behaviour Analysis", *Artificial Intelligence Review*, Vol. 12, pp. 265–319, 1998.

A.M. Sánchez, M. Lozano, P. Villar and F. Herrera, "Hybrid Crossover Operators with Multiple Descendents for Real-Coded Genetic Algorithms: Combining Neighborhood-based Crossover Operators", *International Journal of Intelligent Systems*, Vol. 24, No. 5, pp. 540–567, 2009.

C. García-Martínez, M. Lozano, F. Herrera, D. Molina and A.M. Sánchez, "Global and Local Real-Coded Genetic Algorithms Based on Parent-Centric Crossover Operators", *European Journal of Operational Research*, Vol. 185, No. 3, pp. 1088–1113, 2008.

Kalyanmoy Deb, Ashish Anand and Dhiraj Joshi, "A computationally efficient evolutionary algorithm for real-parameter optimization", *Evolutionary Computation*, Vol. 10, No. 4, pp. 371–395, 2002.

el programa personalmente al instructor. El programa deberá pedir al usuario (de forma interactiva y NO a través de la línea de comandos) los parámetros principales del algoritmo genético: tamaño de la población, porcentaje de cruza, porcentaje de mutación, número máximo de generaciones y nombre del archivo donde se almacenarán los datos de cada corrida. 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 cruzas efectuadas, número de mutaciones efectuadas y cadena real 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, su cadena de números reales, los valores de cada una de las variables decodificadas, el valor de la función objetivo y los valores de todas las restricciones, así como un mensaje que indique si la solución es factible o no.

2. **(10 puntos)** Un reporte donde se describa a detalle la forma en la que funciona la técnica propuesta, así como un breve análisis (informal) que indique sus mayores ventajas y desventajas (reales o probables). Asimismo, indique claramente en qué basó su idea y el razonamiento empleado para suponer que el tipo de técnica propuesta funcionaría.
3. **(30 puntos)** Una corrida de ejemplo por cada uno de los problemas incluidos en el punto 5, y los resultados promediados de AL MENOS 30 corridas independientes para cada problema (nótese que los resultados de estas 30 corridas deben aparecer en una tabla donde se indiquen los valores de las variables de decisión y de $f(x)$ para cada una de ellas) en las que se usen los mismos parámetros (porcentaje de cruza y mutación, tamaño de la población y máximo número de generaciones) pero diferente semilla para generar números aleatorios. En el reporte deberá aparecer una corrida representativa por problema, así como la mejor solución obtenida, la media de aptitud, la peor solución obtenida y la desviación estándar de las 30 corridas efectuadas. Deberán mostrarse los valores de todas las variables y de todas las restricciones, así como la $f(x)$ correspondiente en cada caso. Debe indicarse, para cada solución reportada, si ésta es factible o no. Asimismo, se incluirá la gráfica de convergencia correspondiente a la solución que esté en la mediana de los resultados obtenidos (de las 30 corridas) para cada problema. 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 máxima de cada generación. TODAS las corridas efectuadas deberán aparecer en archivos diferentes en un CD, además del código fuente del programa.
4. **(Bonificación : hasta 170 ó 340 puntos)** Se bonificará con diferentes puntajes (hasta un máximo de 170 puntos) a los que presenten un conjunto de corridas (al menos 30) en las que la mejor solución producida iguale (o supere) a la mejor solución presentada para cada uno de los problemas descritos a continuación. Los puntos asignados están en función del grado de dificultad de cada problema. Alternativamente, si muestra un conjunto de corridas cuya solución en la mediana iguale o supere a la mejor solución presentada en las tablas compara-

tivas correspondientes, la bonificación será del doble de lo indicado para cada problema, hasta llegar a un máximo de **340 puntos** (estas 2 bonificaciones no son acumulables). Estas corridas se deberán presentar en CD, pero la tabla que resume los resultados deberá incluirse en el reporte. En cada caso, debe indicarse claramente cuáles fueron los parámetros usados para el AG (tamaño de población, porcentaje de cruce y mutación y número máximo de generaciones).

5. **Funciones de prueba:** Los siguientes 3 problemas serán los que tendrá que utilizar para probar su programa:

• **Problema 1 (20 puntos):** Maximizar:

$$f(\vec{x}) = \left| \frac{\sum_{i=1}^n \cos^4(x_i) - 2 \prod_{i=1}^n \cos^2(x_i)}{\sqrt{\sum_{i=1}^n ix_i^2}} \right| \quad (1)$$

sujeta a:

$$\begin{aligned} g_1(\vec{x}) &= 0.75 - \prod_{i=1}^n x_i \leq 0 \\ g_2(\vec{x}) &= \sum_{i=1}^n x_i - 7.5n \leq 0 \end{aligned} \quad (2)$$

donde $n = 20$ y $0 \leq x_i \leq 10$ ($i = 1, \dots, n$). Se desconoce el óptimo global de esta función. La mejor solución reportada en la literatura para este problema es: $f(\vec{x}^*) = 0.803619$. La restricción g_1 casi está activa ($g_1 = -10^{-8}$).

• **Problema 2 (50 puntos):** Minimizar:

$$f(\vec{x}) = 3x_1 + 0.000001x_1^3 + 2x_2 + (0.000002/3)x_2^3 \quad (3)$$

sujeta a:

$$\begin{aligned} g_1(\vec{x}) &= -x_4 + x_3 - 0.55 \leq 0 \\ g_2(\vec{x}) &= -x_3 + x_4 - 0.55 \leq 0 \\ h_3(\vec{x}) &= 1000 \sin(-x_3 - 0.25) + 1000 \sin(-x_4 - 0.25) + 894.8 - x_1 = 0 \\ h_4(\vec{x}) &= 1000 \sin(x_3 - 0.25) + 1000 \sin(x_3 - x_4 - 0.25) + 894.8 - x_2 = 0 \\ h_5(\vec{x}) &= 1000 \sin(x_4 - 0.25) + 1000 \sin(x_4 - x_3 - 0.25) + 1294.8 = 0 \end{aligned} \quad (4)$$

donde $0 \leq x_1 \leq 1200$, $0 \leq x_2 \leq 1200$, $-0.55 \leq x_3 \leq 0.55$, y $-0.55 \leq x_4 \leq 0.55$. La mejor solución conocida es: $x^* = (679.9453, 1026.067, 0.1188764, -0.3962336)$ donde $f(x^*) = 5126.4981$.

- **Problema 3 (100 puntos):** Minimizar:

$$f(\vec{x}) = e^{x_1 x_2 x_3 x_4 x_5} \quad (5)$$

sujeta a:

$$\begin{aligned} h_1(\vec{x}) &= x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_5^2 - 10 = 0 \\ h_2(\vec{x}) &= x_2 x_3 - 5 x_4 x_5 = 0 \\ h_3(\vec{x}) &= x_1^3 + x_2^3 + 1 = 0 \end{aligned} \quad (6)$$

donde $-2.3 \leq x_i \leq 2.3$ ($i = 1, 2$) y $-3.2 \leq x_i \leq 3.2$ ($i = 3, 4, 5$). El óptimo global es $x^* = (-1.717143, 1.595709, 1.827247, -0.7636413, -0.763645)$ donde $f(x^*) = 0.0539498$.

Lectura

1. Lea el siguiente artículo:

Larry J. Eshelman, **The CHC Adaptive Search Algorithm: How to Have Safe Search When Engaging in Nontraditional Genetic Recombination**, in Gregory J.E. Rawlins (editor), *Foundations of Genetic Algorithms*, pp. 265–283, Morgan Kaufmann Publishers, San Mateo, California, USA, 1991.

Conteste lo siguiente:

(10 puntos) ¿En qué aspectos difiere CHC de un algoritmo genético tradicional? ¿Cuál fue la motivación principal del autor para proponer este algoritmo? Explique con sus propias palabras.

(5 puntos) ¿Cómo funciona el operador HUX? ¿Cuál es su motivación principal? Explique con sus propias palabras.

(5 puntos) ¿Por qué argumenta el autor que CHC no tiene la propiedad del paralelismo implícito sino una versión más débil de éste? Explique con sus propias palabras.

Fecha de entrega: *Viernes 6 de julio 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. No se recibirán tareas entregadas más de 10 días hábiles después de la fecha antes indicada (se le recuerda que la entrega de TODAS las tareas es requisito para tener derecho a examen).