

Práctica 2. “Algoritmos genéticos: codificación binaria”

Dra. Miriam Pescador Rojas
Escuela Superior de Cómputo (ESCOM-IPN)
Laboratorio de Inteligencia Artificial
Cómputo Evolutivo, CIC-IPN
(email: mpescadorr@ipn.com)

16 de marzo de 2023

1 Objetivo.

El objetivo de esta práctica es que el alumno entienda el funcionamiento básico de los algoritmos genéticos con codificación binaria, mediante la implementación de diferentes estrategias de operadores de selección, cruce y mutación.

2 Introducción

El esquema general de un algoritmo evolutivo puede verse en el algoritmo 1.

Algorithm 1: Esquema general de un algoritmo evolutivo

```
1  $G \leftarrow 0$ ;  
2 Inicializar  $P_g$ ;  
3 Evaluar ( $(P_g)$  en la función objetivo  $f(\bar{x})$ );  
4 begin  
5   while Criterio de determinación no satisfecho do  
6      $padres \leftarrow seleccionarPadres()$ ;  
7      $P'_g \leftarrow operadorRecombinacion(padres)$ ;  
8      $P''_g \leftarrow OperadorMutacion(P'_g)$ ;  
9      $Evaluar(P''_g)$ ;  
10     $P_g + 1 \leftarrow seleccionar(P_g \cup P''_g)$ ;  
11     $G \geq G + 1$ ;  
12  end  
13 end
```

Donde los operadores de selección, cruce y mutación varían de acuerdo con la representación del problema, por lo que en esta práctica se resolverán problemas de optimización continua usando codificación binaria.

Tome como referencia la estructura de algoritmo genético que se proporciona en el siguiente enlace.

<https://colab.research.google.com/drive/1XdXnGJl5wKR5xYLh27ZnWtHDCf4YW0UW?usp=sharing>

3 Consignas

1. Funciones de prueba.

Resuelva los siguientes 3 problemas de optimización mono-objetivo (para un vector de 10 variables de decisión):

■ Esfera

$$f_{esfera}(x) = \sum_{i=1}^n x_i^2 \quad (1)$$

donde $n = 10$ y $-5.12 \leq x_i \leq 5.12$

■ Rosenbrock

$$f_{rosenbrock}(x) = \sum_{i=1}^{n-1} 100(x_i^2 - x_{i+1})^2 + (1 - x_i)^2 \quad (2)$$

donde $n = 10$ y $-2.048 \leq x_i \leq 2.048$

■ Ackley

$$f_{ackley}(x) = -20 \exp \left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2} \right) - \exp \left(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i) \right) + 20 + e \quad (3)$$

donde $n = 10$ y $-32.768 \leq x_i \leq 32.768$

Figure 1: Funciones de prueba de optimización mono-objetivo

2. **Codificación binaria.** Implemente un algoritmo genético con codificación binaria que considere los límites superior e inferior de cada problema y 4 decimales de precisión. Recuerde que para calcular el tamaño del cromosoma debe usar la siguiente formula por cada variable de decisión .

$$numDigitos = \log_2(limSuperior \times 10^{numDecimales} - limInferior \times 10^{numDecimales}) \quad (1)$$

Para más detalles, revise las páginas 93 a la 97 de los apuntes del Dr. Carlos Coello

En el código debe implementar una función de codificación y una de decodificación. Muestre pruebas sencillas del funcionamiento correcto de este procedimiento.

3. **Estrategia de selección de padres** Lea el capítulo 6 Técnicas de selección de los apuntes (página 113) Y elija uno de los siguientes mecanismos de selección proporcional .

- La ruleta.
- Sobrante estocástico sin reemplazo.
- Sobrante estocástico con reemplazo.
- Universal Estocástica.
- Muestreo determinístico.

Argumente el por qué de su selección.

4. Estrategia de selección de sobrevivientes

Lea la sección 6.2 de técnicas de selección de los apuntes (página 124) y elija uno de los siguientes mecanismos.

- Selección mediante torneo (binario determinista o probabilista)
- Selección del estado uniforme
- Selección disruptiva
- Jerarquías no lineales

Argumente el por qué de su selección.

5. Operador de cruza. Realice la implementación de cruza de 2 puntos y cruza uniforme (páginas 135-139).

6. Operador de mutación. Implemente el operador de mutación donde con cierta probabilidad uno de los alelos del cromosoma puede cambiar de valor. Por ejemplo de 0 a 1 o viceversa.

7. Mecanismo de reparación del individuo.

Tanto los operadores de cruza como mutación pueden generar soluciones no factibles, es decir que no se encuentren dentro de los límites inferior o superior de cada variable como lo definen las restricciones iniciales del problema. Por lo que debe implementarse una estrategia de reparación, por ejemplo si la solución tiene un valor menor al límite inferior o mayor al límite superior igualar a los límites correspondientes según sea el caso.

8. Procedimiento de elitismo. Es importante considerar que por cada generación del algoritmo genético debe retenerse al mejor individuo de la población actual, para evitar fluctuaciones en la convergencia del algoritmo.

9. Criterio de Paro. Como criterio de paro del algoritmo emplearemos la siguiente expresión:

$$f(\vec{x}_{best}) - f(\vec{x}^*) = \epsilon \quad (2)$$

donde $f(\vec{x}^*)$ es el valor óptimo global conocido de la función, $f(\vec{x}_{best})$ la evaluación del mejor individuo de la población en la función objetivo y $\epsilon = 0.001$. Una vez que se cumpla la ecuación anterior se detendrá el proceso de búsqueda del algoritmo genético.

10. Configuración de parámetros numéricos del algoritmo. Considere los siguientes valores de prueba para cada parámetro numérico:

- Tamaño de la población = $\{20, 50, 100, 200\}$ individuos
- Porcentaje de cruza = $\{0.5, 0.7, 0.9\}$
- Porcentaje de mutación = $\{0.01, 0.1, 0.25\}$

11. Tabla de resultados. Realice 30 ejecuciones independientes (con diferente semilla para generación de aleatorios) por cada versión de algoritmo y cada problema, obtenga como medidas de tendencia central el mínimo, máximo y promedio del número de evaluaciones de la función objetivo para alcanzar el valor óptimo deseado. Un ejemplo de reporte resultados puede ser la siguiente tabla.

Parámetros numéricos			Estadísticas		
tamaño población	porcentaje cruza	porcentaje mutación	mínimo	maximo	promedio
20	0.5	0.01			
		0.1			
		0.25			
	0.7	0.01			
		0.1			
		0.25			
50	0.5	0.01			
		0.1			
		0.25			
....

12. **Graficas de convergencia.** De acuerdo con el ejemplo proporcionado muestre la grafica de convergencia para la versión de algoritmo que encuentra el valor óptimo de la función en el menor número de evaluaciones de la función objetivo. Considere el siguiente ejemplo.

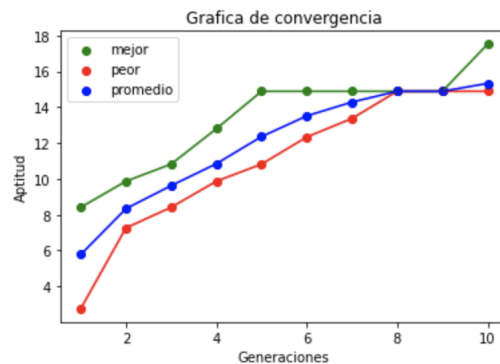


Figure 2: Ejemplo de grafica de convergencia

13. **Discusión de resultados.** Realice una discusión de resultados por cada problema de optimización, contestando a las siguientes preguntas:

- ¿Cuál fue la versión de algoritmo genético que convergió (en promedio) más rápido a la solución óptima conocida?
- ¿Cuál es el tipo de cruza que obtuvo mejores resultados? ¿Por qué?
- ¿Qué tamaño de población obtuvo mejores resultados?
- ¿Cuál de los parámetros configurados en el algoritmo genético considera que es más sensible en el proceso de búsqueda? (puede ser un parámetro numérico, categórico o ambos).
- Considera importante realizar pruebas con otros valores de parámetros, si o no y ¿por qué?. Si su respuesta es afirmativa cómo diseñaría las pruebas de estos experimentos.

Agregue conclusiones generales en su reporte.

Nota. El lenguaje de programación es el de su preferencia. Si utiliza python debiera mostrar en un notebook la ejecución de las pruebas, de lo contrario mostrar en el reporte las pruebas básicas que muestren el correcto funcionamiento del algoritmo genético.

La fecha límite de entrega sin penalización es el **viernes 31 de marzo de 2023 a las 10:00 pm**. El desarrollo de la práctica es individual o por parejas. El envío de la práctica (reporte y código) es a través de la plataforma correspondiente y habrá penalización de 10% sobre la calificación obtenido por cada día de retraso.