

# Práctica 1. “Algoritmos genéticos con codificación binaria”

Dra. Miriam Pescador Rojas  
Escuela Superior de Cómputo (ESCOM-IPN)  
(email: mpescadorr@ipn.com)

21 de octubre de 2024

## 1 Objetivo.

El objetivo de esta práctica es que el alumno entienda el funcionamiento básico de los algoritmos genéticos para resolver problemas de optimización con codificación binaria.

Para cumplir con el objetivo general se proponen los siguientes objetivos particulares:

### 1.1 Objetivos específicos

- Entender el funcionamiento de los pasos que sigue un algoritmo genético.
- Estudiar diferentes estrategias de operadores de selección, cruza y mutación.
- Implementar un algoritmo genético para buscar soluciones de un nonograma (puzzle).

## 2 Introducción

Los algoritmos genéticos (GA por sus siglas en inglés) son meta-heurísticas de búsqueda que imitan el proceso de evolución natural, utilizando conceptos como la "Selección Natural" y la "Herencia Genética" introducidos inicialmente por las teorías de Charles Darwin.

Los algoritmos genéticos fueron desarrollados originalmente por John Holland en 1975, donde las posibles soluciones de un problema de optimización son representadas por individuos que conforman una población. Cada individuo se codifica a través de una cadena conocida como **cromosoma** la cual se compone de **genes**, estos representarán las variables de decisión del problema. La evaluación de cada individuo en la función objetivo indicará la aptitud de este para sobrevivir de una generación a otra. Por otro lado, los individuos de una población pueden reproducirse, combinando su información genética (**procedimiento de cruza o recombinación**) para crear descendientes (nuevas soluciones). Las soluciones hija pueden tener variaciones en su cromosoma a través de un procedimiento conocido como **mutación**. Finalmente, existe un proceso de selección de sobrevivientes donde se favorecerá la sobrevivencia de los individuos más aptos de la población, es decir, aquellas soluciones que optimicen en mayor medida a la función objetivo.

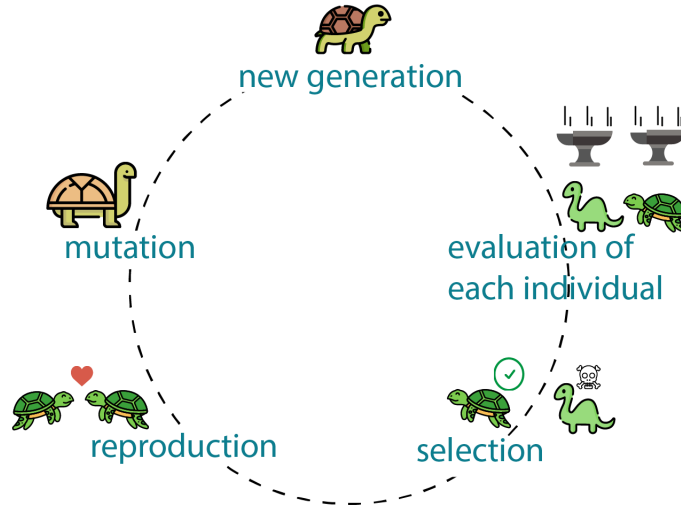


Figure 1: Proceso evolutivo de las especies

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 paro 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, cruza y mutación varían de acuerdo con la representación del problema, por lo que en esta práctica se resolverán problemas de optimización en espacios discretos usando codificación binaria.

Puede usar como referencia dos ejemplos de estructuras de algoritmo genético que se proporciona en el siguiente enlace.

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

## 2.1 Búsqueda de soluciones para un nonograma.

Un nonograma es un tipo de rompecabezas lógico que consiste en rellenar celdas dentro de una cuadrícula para revelar una imagen oculta.

El objetivo de resolver este tipo de rompecabezas es descubrir un patrón oculto basado en los números que aparecen en los bordes de la cuadrícula. Estos números indican cuántas celdas consecutivas deben rellenarse en cada fila y columna. Los jugadores deben deducir qué celdas rellenar y cuáles dejar vacías, usando lógica para resolver el puzzle.

Entre las reglas básicas se tiene:

- Los números en cada fila o columna muestran el tamaño de los bloques consecutivos de celdas que deben ser rellenadas.
- Si hay más de un número, indica que hay varios bloques, con al menos una celda vacía entre ellos.
- El desafío es combinar la información de las filas y columnas para completar el rompecabezas correctamente.
- Cuando el puzzle se resuelve correctamente, el resultado es una imagen (pixelada), que pueden ser figuras simples hasta escenas complejas.

## 3 Consigna 1. Implementación de algoritmo genético con codificación binaria.

Implemente un algoritmo genético con codificación binaria para buscar la solución de un nonograma de tamaño  $N \times N$ . La solución que implemente debe considerar los aspectos que se describen en cada una de las siguientes subsecciones.

### 3.1 Función objetivo

Minimizar la función objetivo, dada por la siguiente expresión:

$$\min f(X) = f_1(X) + f_2(X) \quad (1)$$

Donde:

$$f_1(X) = \sum_{i=1}^N \sum_{j=1}^Q \|r_{ij} - x_{ij}\| \quad (2)$$

$$f_2(X) = \sum_{i=1}^N \sum_{j=1}^Q \|c_{ij} - x_{ij}\| \quad (3)$$

$r_{ij}$  y  $c_{ij}$  son las reglas a cumplir por las filas y columnas, respectivamente, de la cuadrícula.  $N$  es el tamaño del nonograma.  $Q$  es el número máximo posible de 1's consecutivos que puede tener el nonograma.

Suponga el siguiente ejemplo de nonograma, donde las reglas son:

$$r = [(0, 0, 3), (0, 0, 5), (1, 1, 1), (0, 0, 5), (0, 1, 1)] \text{ y } c = [(0, 0, 3), (0, 2, 2), (0, 0, 4), (0, 2, 2), (0, 0, 3)]$$

Para este ejemplo  $N = 5$  y  $Q = 3$ .

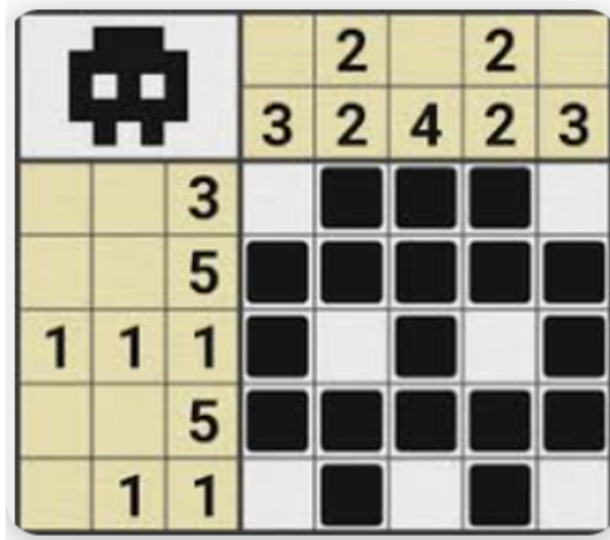


Figure 2: Caption

Si se tiene un individuo X codificado como :

$X = [0, 0, 0, 0, 1,$   
 $1, 1, 1, 0, 0,$   
 $0, 0, 0, 1, 0,$   
 $0, 0, 1, 1, 0,$   
 $1, 0, 1, 0, 1]$

En términos de reglas esa cadena de 1's y 0's se puede decodificar como:

$r_x = [(0, 0, 1), (0, 0, 3), (0, 0, 1), (0, 0, 2), (1, 1, 1)]$  y  $c_x = [(0, 2, 1), (0, 0, 2), (0, 1, 2), (0, 0, 2), (0, 1, 1)]$ , al aplicar las ecuaciones tenemos que el valor de la función objetivo es  $f(X) = 11 + 14 = 25$ .

La implementación del código solicitado debe considerar 1 función de codificación y otra de decodificación para interpretar las reglas y visualizar la solución. Muestre pruebas sencillas del funcionamiento correcto de este procedimiento.

### 3.2 Operadores genéticos

- **Operador de cruza.** Realice la implementación de dos operadores de recombinación, a) cruza de 2 puntos y b) cruza uniforme (páginas 135-139).
- **Operador de mutación.** Implemente el operador de mutación donde con cierta probabilidad cada uno de los alelos del cromosoma puede cambiar de valor. Por ejemplo, para el caso de codificación binaria un 0 cambia a 1 o viceversa.
- **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.
  - La ruleta.
  - Sobrante estocástico sin reemplazo.
  - Sobrante estocástico con reemplazo.
  - Universal Estocástica.

- Selección mediante torneo (binario, n-ario, determinista)

Argumente el por qué de su selección.

- **Estrategia de selección de sobrevivientes**

Se considerará una estrategia de selección extintiva donde deberá ordenar a la población actual de padres y descendientes de acuerdo a su aptitud (de mayor a menor) y sobrevivirán los  $n$  mejores individuos, manteniendo fijo el tamaño de la población en cada generación del algoritmo. Este **procedimiento de elitismo** es importante para evitar fluctuaciones en la convergencia hacia el valor óptimo global del problema.

- **Criterio de Paro.** Como criterio de paro del algoritmo emplearemos dos condiciones

**Condición 1.** Encontrar el nonograma que cumpla con todas las reglas preestablecidas.

**Condición 2.** Si el mejor y el peor individuo son muy parecidos entre sí, es decir.

$$|f(\vec{x}_{worst}) - f(\vec{x}_{best})| \leq \epsilon \quad (4)$$

Una vez que se cumpla cualquiera de las dos condiciones se detendrá el proceso de búsqueda del algoritmo genético.

## 4 Consiga 2. Pruebas y resultados

- **Configuración de parámetros numéricos del algoritmo.** Se sugiere alguna combinación de los siguientes valores de parámetros numéricos, elija la que de mejores resultados o proponga una nueva argumentando el por qué vario de esa manera. No es válido usar poblaciones mayores a 1000 individuos.

- Tamaño de la población = 200, 500 individuos
- Porcentaje de cruza = {0.6, 0.9}
- Porcentaje de mutación = {0.1, 0.3, 0.5}

Resuelva los siguientes ejemplos de pruebas.

- **Tabla de resultados.** Realice 20 ejecuciones independientes (con diferente semilla para generación de aleatorios) por cada versión de algoritmo y cada problema, obtenga como medidas de tendencia central lo siguiente:
  - El promedio del número de evaluaciones de la función objetivo que se generaron por cada ejecución.
  - Indique cuántas de las 20 ejecuciones si llegaron al valor óptimo antes de perder diversidad en la población.
  - El mínimo, máximo, promedio y desviación estándar del valor de la función objetivo alcanzado.
- **Gráficas de convergencia.** Para cada función objetivo, grafique la evolución del algoritmo para los casos: mejor, peor y mediana. Considere el siguiente ejemplo, y que en el eje de las  $y$  irá el valor de la función objetivo del mejor individuo en la población por cada generación.
- **Discusión de resultados.** Realice una discusión de resultados por cada problema de optimización, resumiendo a qué estrategia le fue mejor y para qué problema, de ideas de como mejorar la convergencia en los casos donde no le fue bien.
- Entregar reporte con conclusiones individuales.



[illegible]

Figure 4: Nonograma de prueba 2

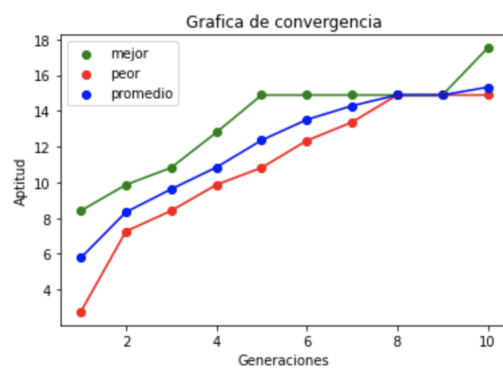


Figure 5: Ejemplo de gráfica de convergencia