

# Práctica 5. “Algoritmos genéticos: codificación entera para permutaciones”

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

8 de diciembre de 2023

## 1 Objetivo.

El objetivo de esta práctica es que el alumno entienda el funcionamiento básico de los algoritmos genéticos con codificación entera para permutaciones, aplicados al caso de encontrar cuadrados mágicos.

## 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 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, cruce y mutación varían de acuerdo con la representación del problema, por lo que en esta práctica se resolverán problemas de búsqueda de solución en un espacio discreto donde la codificación de las soluciones es para permutaciones.

### 3 Problema: búsqueda de soluciones para cuadrados mágicos

En esta práctica se analizará el problema de generar un cuadrado mágico, el cual se trata de una matriz de  $n$  filas por  $n$  columnas que contiene una serie de números enteros consecutivos del 1 a  $n^2$ , de forma tal que la suma de los números por columnas, filas y diagonales principales sea la misma.

La figura 1 muestra ejemplos de cuadrados mágicos de tamaños  $3 \times 3$  y  $4 \times 4$ , para el primer caso la suma que se espera por cada fila, columna y diagonales principales es igual a 15 mientras que para el segundo caso es 34. Este valor se le conoce como constante mágica  $k$  y puede calcularse usando la ecuación 1.

6	1	8
7	5	3
2	9	4

16	3	2	13
5	10	11	8
9	6	7	12
4	15	14	1

Figure 1: Ejemplos de cuadrados mágicos.

$$k = \frac{n(n^2 + 1)}{2} \quad (1)$$

donde  $n$  es el tamaño del cuadrado mágico y define el número filas y columnas.

Una de las áreas de aplicación de los cuadrados mágicos ha sido para componer música. Músicos de la India usaron el cuadrado mágico en el ritmo, más que en las notas ya que estos son útiles en ciclos de tiempo y ritmo aditivo. En este caso, no son importantes las propiedades habituales de un cuadrado, sino la relación del número central con la suma total de todos los números del cuadrado mágico.

Otra aplicación fue su extensión en el juego de sudoku, creado en 1979 en Japon, se trata de un rompecabezas de lógica donde el objetivo es rellenar una cuadrícula de  $9 \times 9$  celdas dividida en subcuadrículas de  $3 \times 3$ . Cada celda contendrá un número del 1 al 9 sin repetir valores en cada fila, columna y subcuadrículas.

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

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

A diferencia de la práctica anterior, se requieren de otro tipo de operadores evolutivos que mantengan la estructura del cromosoma para representar una permutación ya que de lo contrario los operadores implementados anteriormente generarían soluciones no válidas que requieran una reparación.

De acuerdo con la asignación al equipo de trabajo implemente las técnicas correspondientes.

#### 3.1 Operadores de cruce para permutaciones

- Order Crossover (OX)
- Partially Mapped Crossover (PMX)
- Position-based Crossover
- Order-based Crossover

- Cycle Crossover (CX)

### 3.2 Operadores de mutación para permutaciones

- Order

## 4 Consignas

### 1. Funciones de prueba.

Resuelva el problema de encontrar cuadrados mágicos usando las siguientes propuestas de funciones objetivo:

- **Conteo de éxitos.** Contar el número de filas, columnas y diagonales principales que cumplen con la constante mágica, aquellas que lo cumplan incrementarán la variable *exito* definida por la ecuación 2. El valor mínimo de esta función es 0 y el máximo es  $2n + 2$ , por lo que el problema se define como maximizar el número de éxitos dependiendo del tamaño del cuadrado mágico.

$$\max \sum_{i=1}^{2n+2} exito_i \quad (2)$$

donde  $exito \in \{0, 1\}$ .

- **Error mínimo a la constante mágica.** En esta propuesta se calcula la diferencia entre el valor de la constante mágica y el valor de la suma de cada fila, columna y diagonales principales. Esta función se puede definir por la ecuación 3. El mejor valor de esta función será 0, mientras que un número mayor a cero define el error en la aproximación del cuadrado.

$$\min \sum_{i=1}^{2n+2} |k - s_i| \quad (3)$$

donde  $k$  es la constante mágica y  $s_i$  se refiere al valor de la suma de la fila, columna o diagonal según sea el caso.

2. **Codificación entera.** Implemente un algoritmo genético con codificación entera. Defina una lista o arreglo con los elementos de cada casilla del cuadrado, como se muestra en la figura 2.

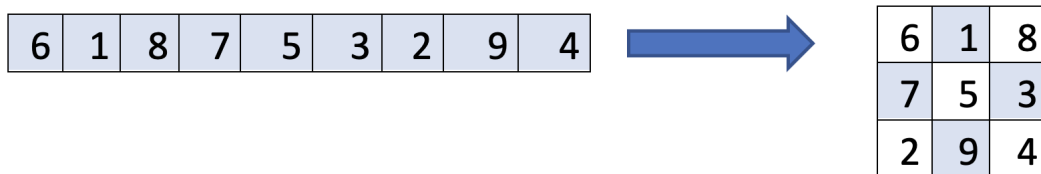


Figure 2: Representación de un cuadrado mágico de 3X3 en una lista

3. **Estrategia de selección de padres.** Elija alguno de los métodos vistos anteriormente, puede usar el mismo mecanismo que utilizó en la práctica anterior.

4. **Estrategia de selección de sobrevivientes.**

Se considerará la estrategia de selección extintiva donde deberá ordenar a la población (de padres y descendientes), en cada generación de acuerdo a su aptitud (de mayor a menor) y sobrevivirán los  $m$  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.

5. **Operador de cruza.** Implementa la técnica de cruza para permutaciones asignada.

- OrderCrossover(OX)
- Partially Mapped Crossover (PMX)
- Position-basedCrossover
- Order-basedCrossover CycleCrossover(CX)

Para mayor detalle consulte las páginas 145 a la 149 de los apuntes de Cómputo evolutivo del Dr. Carlos Coello.

6. **Operador de mutación.** Implementa la técnica de mutación para permutaciones asignada.

- Mutación por Inserción
- Mutación por Desplazamiento
- Mutación por Intercambio Recíproco
- Mutación Heurística

Para mayor detalle consulte las páginas 159 y 160 de los apuntes de Cómputo evolutivo del Dr. Carlos Coello.

7. **Criterio de Paro.** El criterio de paro del algoritmo será alcanzar el valor mínimo o máximo de las funciones objetivo del mejor individuo en la población, es decir que se encontró al menos un cuadrado mágico.

8. **Configuración de parámetros numéricos del algoritmo.** Para la configuración de los parámetros tales como el tamaño de la población, el porcentaje de cruza, el porcentaje de mutación, realice pruebas y elija un conjunto de valores que obtenga mejores resultados. Comente la forma en qué determinó estos valores.

9. **Tabla de resultados.**

Realice 20 ejecuciones independientes (con diferente semilla para generación de aleatorios) del algoritmo por cada función objetivo y obtenga como medidas de tendencia central lo siguiente:

- El mínimo, máximo, promedio y desviación estandar del número de evaluaciones de la función objetivo.
- El mínimo, máximo, promedio y desviación estandar del número de generaciones empleadas por el algoritmo.

Resuelva el problema para tamaños de cuadrados mágicos  $n = 5$  y  $n = 10$ . Reporte los resultados en tablas comparativas por cada problema usando las diferentes configuraciones de parámetros.

10. **Graficas 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.

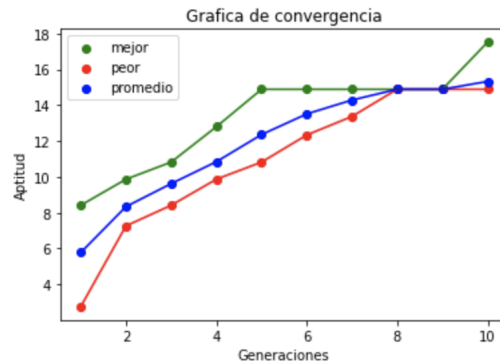


Figure 3: Ejemplo de grafica de convergencia

11. **Punto extra.** Aquellas personas que entreguen a tiempo, y acoplen alguna estrategia diferente de los temas expuestos, tal como búsqueda tabú en el algoritmo y demuestren mejora para llegar a la solución del problema tendrán un punto extra en la calificación final del curso.
12. **Discusión de resultados.** Realice una discusión de resultados por cada problema de optimización, contestando a las siguientes preguntas:
  - ¿Cuál fue la función objetivo que encontró más rápido el cuadrado mágico?
  - ¿Cómo modificaría el algoritmo para encontrar todos los posibles cuadrados mágicos? (recuerde que existe más de una solución)
  - ¿Modificó los parámetros de su algoritmo para los diferentes tamaños del cuadrado mágico ( $n = 5$  y  $n = 10$ )? Si su respuesta es afirmativa ¿cuáles?.
  - Mencione en promedio cuánto tardó en ejecutarse el algoritmo por cada caso de prueba.

Agregue conclusiones individuales describiendo como fue diseñando su algoritmo y los resultados obtenidos

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. No enviar enlaces con el código, tiene que ser la versión final del código y el documento PDF del reporte sin comprimir.

La fecha límite de entrega sin penalización es el **miércoles 20 de diciembre de 2023 a las 23:00 pm**. El desarrollo de la práctica es individual o por equipos. El envío de la práctica (reporte y código) es a través de la plataforma correspondiente y habrá penalización de 10% sobre la calificación obtenida por cada día de retraso.