

# Ejercicios De Programación Genética - Computación Blanda

Genetic Programming Exercises – Soft Computing

Santiago Sánchez Pulgarín - 1004752250

Orlando Andrés Mejía Vélez - 1112793020

Valentina Gómez Isaza - 1088352305

*Risaralda, Universidad Tecnológica de Pereira, Pereira, Colombia*

**Resumen**— En este artículo se explicarán 2 ejercicios sobre genética de la estructura poblacional (GEP), uno de ellos sobre un algoritmo genético y el otro es un problema de mayoría.

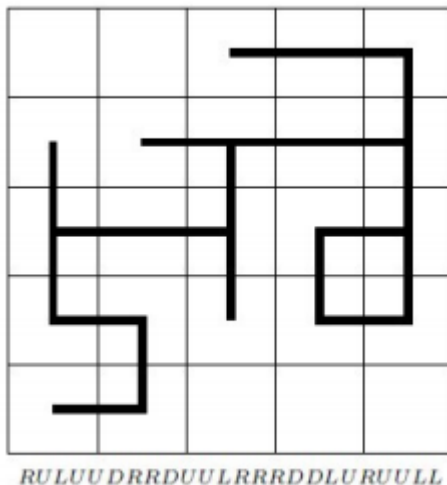
**Palabras clave**— GEP, algoritmo genético, problema de la mayoría

**Abstract—** In this article 2 exercises on genetics of the population structure (GEP) will be explained, one of them about a genetic algorithm and the other is a majority problem.

**Key Word** — GEP, genetic algorithm, majority problem.

I.

Un *paseo* es una sucesión de movimientos sobre una malla entre celdas que comparten un lado. Si ninguna celda se visita 2 veces, el paseo **no se cruza**. Si toda celda se visita el paseo es **óptimo**. Los paseos se pueden codificar como cadenas sobre el alfabeto {U, D, L, R} que se interpretan como los sucesivos movimientos del paseo, que empiezan en la esquina inferior izquierda. Cuando un movimiento intenta salir de la malla es ignorado.

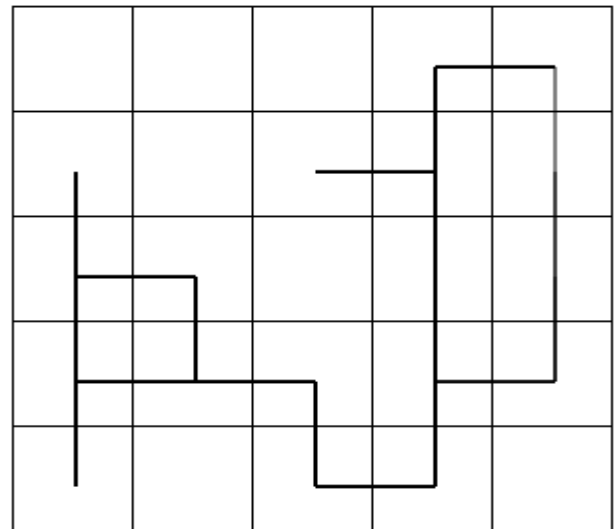


**Figura I.** Un paseo en una malla 5x5

En una malla 5x5:

- i. Defina una función para evaluar el desempeño de cada cromosoma.
- ii. Cree aleatoriamente una población inicial de 5 cromosomas.
- iii. Haga una selección elitista o por ruleta con esa población.
- iv. Defina un operador de cruzamiento entre pares de cromosomas y úselo para obtener una población intermedia.
- v. Defina un operador de mutación y aplíquelo a la población intermedia para tener una nueva generación de cromosomas.

### SOLUCIÓN



Lo anterior es la gráfica del mejor cromosoma con una puntuación de 18. Mas adelante explicaremos como llegamos a él.

- i. Este primer punto se resolvió a partir de la función `function getPoints(crhromosome)`

En el que, a partir de recibir un cromosoma, esta cuenta 1 punto si va hacia arriba (Up), abajo (Down), izquierda (Left), derecha

(Right), y no cuenta puntos si a la hora de moverse sobre pasa el límite del cromosoma. El conteo de puntos se acaba cuando se supera el número de caracteres que posee el cromosoma.

- ii. Los cromosomas aleatorios se crearon a partir de la función `function CreateChromosome` :

donde el tamaño del cromosoma es 25, escoge un numero aleatorio entre el 1 y el 4, y dependiendo que numero salga, el cromosoma se va llenado de las letras U, D, R y L.

UDRRUDRRLDDDRUDDLLRULDLUD

LLURUULDURRRDRDUDLDDUDDRLD

LRLRUDRULRDLULLLLDDDLULLLUL

RDLDRUULRLDUDUDRRDLRURULD

LLDLDDRURLLLLDLRLRUL

Estos son los 5 cromosomas aleatorios que surgieron.

- iii. El cromosoma elitista (el de mejor puntuación) fue el segundo (LLURUULDURRRDRDUDLDDUDDRLD).

- iv. El operador de cruzamiento que escogimos fue intercambiar el primer y el ultimo carácter del cromosoma.

Si nuestro cromosoma es este LLURUULDURRRDRDUDLDDUDDRLD y le aplicamos el operador de cruzamiento quedaría así:

**DLURUULDURRRDRDUDLDDUDDRL**L (Este sería el nuevo cromosoma cruzado).

- v. El operador de mutación que escogimos fue el siguiente:

Dividir el cromosoma elitista y la ruleta a la mitad, cogemos la segunda mitad de cada cromosoma y lo ponemos en el otro.

Si estos son nuestros cromosomas LLURUULDURRRDRDUDLDDUDDRLD – UDRRUDRRLDDDRUDDLLRULDLUD y les aplicamos el operador de mutación quedaría así:

Paso 1

LLURUULDURRRD **RDUDLDDUDDRLD**  
UDRRUDRRLDDDRU **UDDLRLULDLUD**

Paso 2

Los 2 nuevos cromosomas quedarían de la siguiente manera:

LLURUULDURRRD**UDDLRLULDLUD**  
UDRRUDRRLDDDRU**RDUDLDDUDDRLD**

## II.

La figura I muestra, en GEP, una población de cromosomas como 2 genes para el problema de la mayoría. Los genes están ligados por la función OR. La función que mide el desempeño de un cromosoma es el numero de aciertos de la función lógica que el cromosoma expresa.

```
Generation N: 0
01234560123456
NaObaacOAbbcc- [0]
AaNcbbaNcOaacc- [1]
OONcbbbbNcbcbca- [2]
ANNcaacNcObaab- [3]
AbObcbcbOAcacac- [4]
```

**Figura II.** Cromosomas para el problema de la mayoría

- Evalúe el desempeño de algunos cromosomas.
- Seleccione aleatoriamente dos cromosomas y haga cruce en un punto. Mire que otros operadores de cruzamiento se pueden aplicar.
- Seleccione aleatoriamente dos cromosomas y haga una mutación por inversión. Mire que otros operadores de mutación se pueden aplicar. Muestre el fenotipo resultante.

### SOLUCIÓN

- Hemos evaluado el desempeño de los cromosomas y los resultados han sido los siguientes:

```
NaObaacOAbbcc-[0] = 4
AaNcbbaNcOaacc-[1] = 2
OONcbbbbNcbcbca-[2] = 2
ANNcaacNcObaab-[3] = 2
```

- Para el cruzamiento y la mutación, utilizamos la función `crossMutateChromosome` para generar un cruce en un punto de los dos cromosomas aleatorios. Se consideró usar cruce en dos puntos y corte y empalme, pero debido a la falta de recursos de hardware para probar el entrenamiento., se optó por dejar el cruce en un punto.
- Para el operador de mutación, por recomendación de compañeros, se utilizó inversión para mutar los cromosomas.

## III. CONCLUSIONES

Este tipo de algoritmos nos permiten resolver problemas de manera eficaz y eficiente.

El primer ejercicio nos da la facultad de sacar el mejor resultado entre muchos que son estudiados.