

## PRACTICA INTELIGENCIA ARTIFICIAL “ALGORITMO GENETICO”

La finalidad de este trabajo es construir un planeador de rutas, capaz de encontrar la ruta más corta a la salida más cercana, para cada una de las “n” personas que se encuentran en un cuarto.

Este cuarto contiene obstáculos estáticos y varias salidas. La ruta encontrada para la persona “x” no debe llevarla a chocar con algún obstáculo ni con otra persona que a su vez se está moviendo en el cuarto.



La interface del programa tiene la siguiente simbología:

Cubos de colores y mas pequeños :	Personas
Cubos azules:	Obstáculos
Cuadros rojos:	Salidas

Los pasos de un algoritmo genético son los siguientes:

## 1.- INICIALIZACIÓN:

En este primer paso se crea aleatoriamente un conjunto de individuos (que están sobre el espacio S).

Inicialmente tenemos 20 individuos. Los individuos se representan como un vector, cuyos valores van del 0-7, y que significan lo siguiente:

- 0 -> Camina un paso a la izquierda
- 1 -> Camina un paso a la derecha
- 2 -> Camina un paso adelante
- 3 -> Camina un paso atrás
- 4-> Camina en diagonal izquierda hacia arriba
- 5-> Camina en diagonal izquierda hacia abajo
- 6-> Camina en diagonal derecha hacia abajo
- 7-> Camina en diagonal derecha hacia arriba
- 8 -> Quédate en tu lugar

Éstos serían dos ejemplos de individuos:

**1)** 2 - 1 - 2 - 0 - 7 - 4 - 3 - 6 - 3 - 3 - 3 - 0 - 5 - 0 - 3 - 3 - 3 - 1 - 1 - 1 - 3 -  
5 - 1 - 4 - 7 - 4 - 3 - 1 - 6 - 1 - 2 -

**2)** 0 - 4 - 1 - 7 - 5 - 3 - 1 - 1 - 3 - 3 - 3 - 7 - 5 - 1 - 2 - 5 - 6 - 4 -

Después el algoritmo itera en los siguientes pasos hasta que se cumpla alguna condición.

En este caso dicha condición es que evolucionen 10 generaciones.

## 2.- EVALUACIÓN:

La Función F es computada para cada individuo, ordenando la población del mejor individuo al peor. En el ejemplo, la función a optimizar es la longitud de los individuos. Es decir deseamos obtener un individuo con longitud mínima. Por ello, se ordenan en base a su longitud y en orden ascendente.

## 3.- SELECCIÓN:

Se selecciona con la regla de selección - explicada a continuación - una pareja de individuos. La regla utilizada es roulette wheel selection, que elige en base a probabilidad proporcional a su calificación:

- 1.- Se genera un número aleatorio  $r \in [0,1]$ .
- 2.- Se multiplica  $r$  por la suma de las calificaciones de la población (S), obteniéndose  $c = rS$ .
- 3.- Se establece la calificación acumulada ( $Ca$ ) y el índice en cero:  
 $Ca = 0, i=0$

4.- A la calificación acumulada se le suma la calificación del i-ésimo individuo:

$$Ca = Ca + \text{calif}(i)$$

5.- Si  $Ca > c$  entonces el i-esimo individuo es seleccionado.

6.- Si no, entonces se incrementa i y se regresa al paso 4.

Esta selección es con reemplazo, es decir, un mismo individuo puede ser seleccionado varias veces. Así que cada vez que se selecciona un individuo no se quita de la población de la que se está seleccionando, permanece en ella para que pueda ser elegido nuevamente.

## 4.- REPRODUCCIÓN:

Se generan nuevos individuos a partir de la pareja seleccionada en el paso 3.

El algoritmo de reproducción que se ha utilizado es el siguiente:

⇒ Una vez seleccionados 2 individuos  $p1$  y  $p2$  (llamados padres) se procede a cruzarlos.

⇒ Generamos un número aleatorio entre 1 y  $(\text{longitud de } p1)/2$ , llamado  $n1$ .

⇒ Generamos un número aleatorio entre  $(\text{longitud de } p2)/2$  y longitud de  $p2$ , llamado  $n2$ .

⇒ Calculamos la posición en la que se encuentra  $p1$  al dar  $n1$  pasos, y la llamamos  $\text{pasosp1}$ .

⇒ Calculamos la posición en la que se encuentra p2 al dar (longitud de p2)/2 pasos, y la llamamos pasosp2.

⇒ Generamos un camino aleatorio de pasosp1 a pasosp2, llamado camino\_intermedio. Este camino aleatorio verifica que no choque con ningún obstáculo predefinido (estático).

⇒ Copiamos los primeros n1 pasos de p1 en el nuevo individuo ni1.

⇒ Copiamos camino\_intermedio a ni1.

⇒ Copiamos los pasos de p2 de n2 a longitud p2.

Y de la misma manera generamos el segundo nuevo individuo, sólo que ahora tomamos el inicio de p2 y el final de p1.

Esta cruce nos garantiza que obtengamos individuos válidos, es decir que nos van a dar un camino que inicie en la posición original de la persona y finalice en alguna salida. Y que además no choque con ningún obstáculo.

Después de repetir 6 veces los pasos del 2 al 4, se obtiene una población final.

El algoritmo además de estos pasos, incorpora un quinto paso al que llamamos entrenamiento.

## **5.- ENTRENAMIENTO:**

Repite los pasos del 1 al 4 n veces. El primer algoritmo genético coloca en la población inicial del segundo algoritmo genético, a su mejor individuo (lo coloca 2 veces, para asegurarse de que va a preservarse en las siguientes generaciones), el segundo hace lo mismo con el tercero, y así sucesivamente hasta n.

Los algoritmos genéticos tienen una característica peculiar, que es que cada vez que se ejecuta nos da (muy probablemente) respuestas diferentes, y esto es porque la población inicial se genera aleatoriamente.

Estos entrenamientos nos permiten tomar el mejor individuo evolucionado del algoritmo genético anterior, y a su vez generar nuevos individuos. Esto es conocido también como elitismo. Con esto generamos  $n$  veces poblaciones aleatorias y  $(n-1)$  veces la población inicial al menos tiene un buen individuo. Este nuevo individuo al cruzarlo con algún otro buen individuo generado aleatoriamente puede ser que nos dé uno aún mejor.

Parámetros que deben ser especificados en el programa:

Nº de generaciones: cantidad de veces que se produce una nueva generación de individuos, utilizando las reglas anteriormente explicadas. Por lógica, cuantas más generaciones produzcamos, obtendremos mejores individuos.

Nº de individuos por generación: en este ejemplo, la cantidad de individuos de una generación a otra es constante, no varía.

Nº de entrenamientos: cantidad de veces que se lleva a cabo un entrenamiento entre diferentes generaciones de individuos.