

Algoritmo Genético

En serio me gustó trabajar con el algoritmo genético, pues me di cuenta de una nueva manera para encontrar soluciones o optimizar problemas.

Cuando estaba a punto de correr el código la primera vez, no creí que fuera a funcionar, o pensaba que se iba a tardar mucho, pero la verdad me lleve una sorpresa, en seguida me dio una solución, y me emocioné. Aunque ya comparando después con otros algoritmos quedé un poco decepcionado, porque a pesar de que fuera relativamente rápido, no era el mejor, pues comparándolo en tiempo con el de búsqueda por profundidad limitada, es más lento, no por mucho, pero lo es, además ya se hablará en los resultados de otro algoritmo implementado.

La forma que tiene cada individuo está representada por un arreglo, por ejemplo el arreglo [4,3,5,2,1,7,0,6], el cual representa en la matriz una reina en (0,4), otra reina en (1,3), otra en (2,5), otra en (3,2), otra en (4,1) y así sucesivamente.

La forma en la que implemente el algoritmo genético fue la siguiente:

1) Crear población:

Se inicia con una población de 4 individuos totalmente aleatorios.

2) Selección:

Determinamos cual individuo es el mejor (el que tenga menos ataques), y cual es el peor (el que tenga más ataques). Eso lo determinamos un con método creado llamado getFitness().

Después eliminamos de la población al peor individuo.

3) Reproducción:

Una vez teniendo al mejor individuo lo reproducimos con los otros dos individuos, generando así 4 individuos nuevos siendo su padre el mejor de la población (teniendo en cuenta que cada reproducción genera dos hijos). O sea, [0,1,2,3,4,5,6,7] X [5,3,1,2,0,7,6,4]=[0,1,2,3,0,7,6,4] y [5,3,1,2,4,5,6,7].

4) Mutación:

Finalmente aplicamos una mutación a 3 o 2 de los individuos (una mutación aleatoria) y repetimos el procedimiento con la nueva población hasta encontrar una solución.

El algoritmo Totalmente aleatorio simplemente se sacaban arreglos con elementos de número del 0-7, siendo estos individuos, y se verificaba si es una solución, en caso que no lo fuera se volvió a repetir el procedimiento.

Por último se hizo un algoritmo aleatorio inteligente, el cual crea un arreglo inicial=[0,1,2,3,4,5,6,7], y lo que va haciendo es revolver aleatoriamente el arreglo, como si fuera una baraja de poker, y lo padre aquí es que no se repiten números, lo cual es de mucha ayuda pues si se repitiera algún número significaría que no puede ser solución. Este algoritmo aleatorio inteligente resultó ser muy bueno en las pruebas.

Resultados:

Se hicieron varias pruebas, donde se determinaba el tiempo de ejecución, o individualmente en cuantas generaciones (cuantas veces se repetía el ciclo), para encontrar una solución. Y para tener resultados más precisos, lo que se hizo es sacar un promedio de correr el programa 1000 veces, excepto en el totalmente aleatorio pues era muy lento, con este se repitió 100 veces, y los resultados fueron los siguientes:

Promedio Algoritmo genético: 1600

Promedio Aleatorio inteligente: 439

Promedio Totalmente aleatorio : 184949

En base a los resultados tenemos que el algoritmo aleatorio inteligente es aproximadamente 3 veces más rápido que el genético, pero el genético es aproximadamente 100 veces más rápido que el algoritmo totalmente aleatorio.

La verdad si quedé un poco decepcionado porque el aleatorio inteligente fue mejor que el genético, pero comparándolo con uno totalmente aleatorio vemos que realmente funciona el genético, y quizá aumentando la población o haciendo algunas pequeñas modificaciones por ahí, podríamos mejorarlo.