

Aplicación de algoritmo genético: N-queens

Orlando Hernandez Nuñez. Juan Alejandro Bernal Gallego

21 de Octubre del 2019

1 Introducción

En el presente trabajo se busca dar solución al problema de las N-queens mediante el uso de un algoritmo genético. A continuación, se mostrará la configuración de parámetros con las que fue ejecutado el algoritmo genético y los resultados obtenidos.

2 Problema

En el juego de ajedrez la reina amenaza a aquellas piezas que se encuentren en su misma fila, columna o diagonal. El problema de las 8 reinas (o n-queens ya que dependen del numero asignado) consiste en poner sobre un tablero de ajedrez ocho reinas sin que estas se amenacen entre ellas.

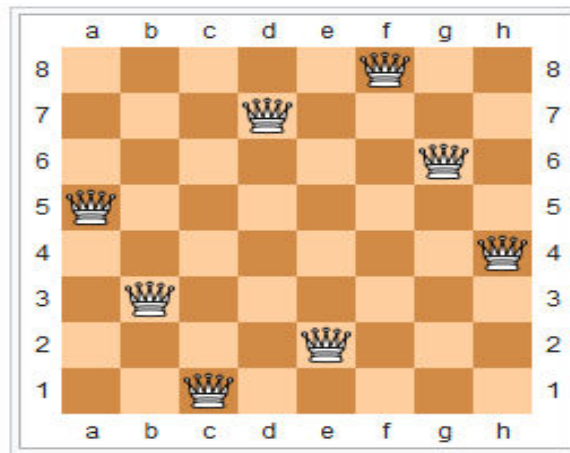


Fig. 1: Solución al problema de las N-queens

3 Algoritmo Genético

3.1 Definición de la codificación

Para representar cada uno de los cromosomas que integrarán la población de posibles soluciones al problema se eligió un vector de números enteros donde cada índice indica una fila del tablero. El valor de cada índice esta entre 0 y N-1 donde N es el número de reinas a distribuir en el tablero y este valor indicará en que columna se ubica la reina dentro del tablero.

Los cromosomas fueron codificados de manera que pudieran representar la ubicación en columnas y filas de cada reina. El gen representa la columna en la que se encuentra cada reina, y el índice del gen representa la posición de las reinas con respecto a las filas.

- Ejemplo cromosoma {1, 2, 0, 3}

0	0	0	1
1	0	0	0
0	0	1	0
0	1	0	0

3.2 Función de fitness

La función de fitness se encarga de contar el número de cruces en diagonales o en columnas.

3.2.1 Calcular cruces en diagonales

Se tiene una posible solución V y se establece que dos reinas en el tablero estan en la misma diagonal si y solo si:

$$\text{Para } i \in \{0, 1, 2, \dots, N-1\} \text{ y } j \in \{0, 1, 2, \dots, N-1\} \text{ y } i \neq j;$$

$$V[i] - i == V[j] - j$$

$$V[i] + i == V[j] + j$$

3.2.2 Calcular cruces en columnas

Se tiene una posible solución V de modo que para calcular si dos reinas estan en la misma columna se cuentan la cantidad de números repetidos que hayan dentro de V. Por ejemplo: Si tenemos [2, 3, 2, 7, 1, 6, 3, 7] se tiene que hay 4 cruces en columna.

Una vez calculadas estas dos cantidades se define la función de fitness de la siguiente forma:

$$fitness = \frac{1}{1 + nc} \quad (1)$$

Donde nc es el numero de cruces definido como la suma de la cantidad de cruces en diagonal y los cruces en columnas. Con la función de fitness definida de esta forma el objetivo será minimizar el número de cruces de modo que (1) se maximice.

3.3 Configuración de parámetros

Los parámetros elegidos para la ejecución del algoritmo genético se presentan a continuación:

$$P_{cruce} = 0.9 \quad (2)$$

$$P_{mutacion} = 0.001 \quad (3)$$

$$Tamano\ de\ poblacion = 20\ individuos \quad (4)$$

4 Resultados

A continuación, se presentan los resultados conseguidos con la configuración descrita previamente para el algoritmo genético.

4.1 Elitista

Se ejecutó el algoritmo hasta alcanzar la iteración $i = 1000$ conservando en cada una de las siguientes generaciones el mejor individuo de la anterior. la solución obtenida fue [3, 7, 0, 4, 6, 1, 5, 2]

0	0	1	0	0	0	0	0
0	0	0	0	0	1	0	0
0	1	0	0	0	0	0	0
0	0	0	0	0	0	1	0
0	0	0	0	1	0	0	0
1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1
0	0	0	1	0	0	0	0

podemos observar en la figura (2) hay 3 curvas, el eje X son el numero de iteraciones y el eje Y representa el valor del fitness. En las 2 curvas online y best-so-far utilizando elitismo destaca un cambio repentino en el promedio

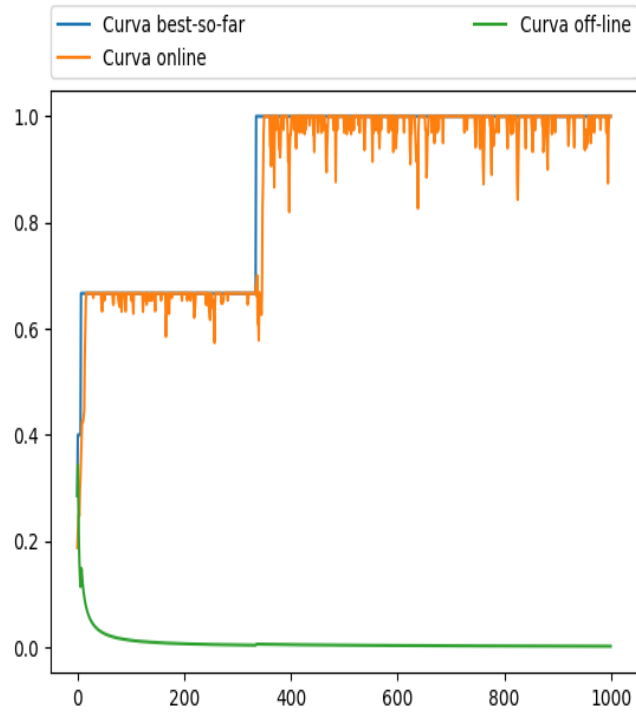


Fig. 2: Curvas de monitoreo ejecución con elitismo

de fitness de los individuos y los mejores fitness respectivamente alrededor de la iteración 300 y 400, mientras que la curva offline no presenta cambios tan grandes durante el proceso evolutivo.

4.2 Sin elitismo

Se ejecutó el algoritmo hasta alcanzar la iteración $i = 1500$ sin aplicar elitismo la solución obtenida fue [5, 2, 0, 7, 4, 1, 3, 6]

0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0
0	1	0	0	0	0	0	0
0	0	0	0	1	0	0	0
0	0	0	0	0	0	0	1
1	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0
0	0	0	0	0	1	0	0

En el caso de la figura (3) se observa que la curva online y best-so-far fluctúan menos en el transcurso de las generaciones, y la curva offline tiene una forma similar al de la figura (2), encontramos que sin el uso de elitismo se necesita de una mayor cantidad de generaciones para poder encontrar una solución.

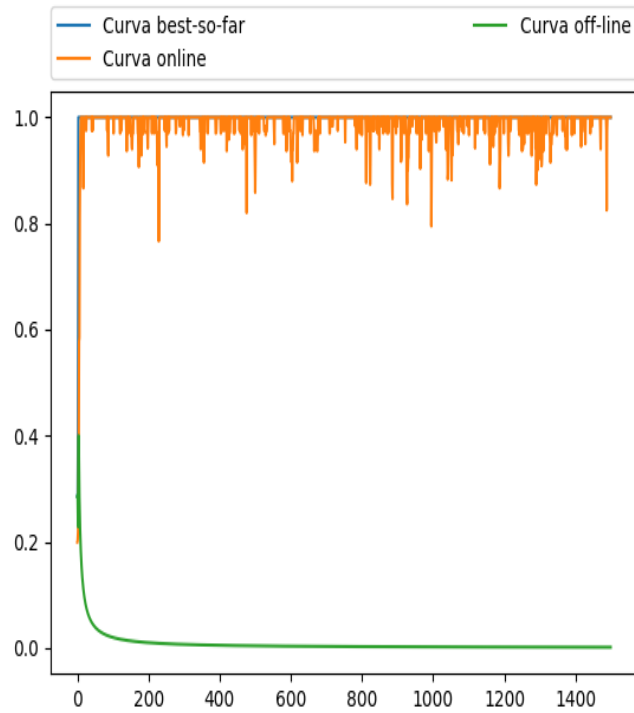


Fig. 3: Curvas de monitoreo ejecución sin elitismo