

Tarea 2

N-Queen Con Algoritmos Geneticos

Alumno: Gonzalo Uribe
Profesor: Alexandre Bergel Profesor 2
Auxiliar: Juan-Pablo Silva
Ayudantes: Alonso Reyes Feris
Gabriel Chandía G.

Fecha de entrega: 9 de noviembre de 2018
Santiago, Chile

1. Introducción

Se evaluó el rendimiento de un algoritmo genético en la resolución del problema N-Queen. Se presentarán a continuación los resultados obtenidos al variar sus parámetros.

2. Resultados

Se realizó un primer experimento en el que se evalúa el resultado de la función de fitness en el mejor candidato de cada generación durante el transcurso de las distintas generaciones.

Para resolver este problema se le resta fitness a una combinación de reinas cada vez que hay dos que se tocan, es por eso que en la figura 1 se ve una fitness negativa y termina el programa cuando esta llega a 0.

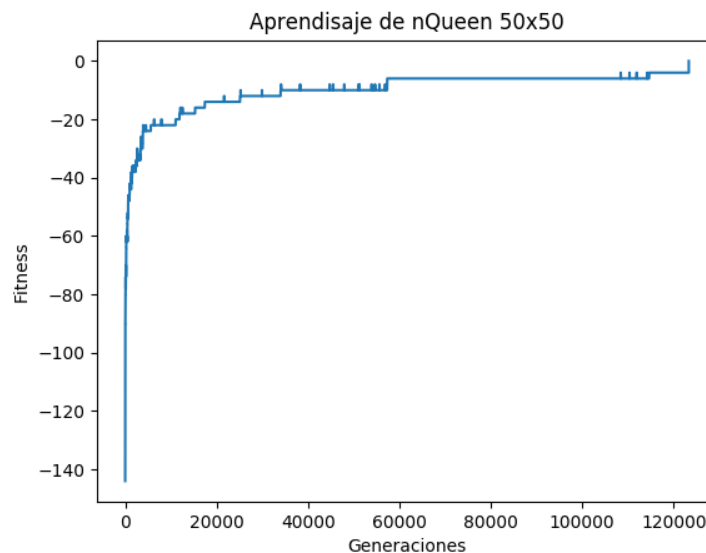


Figura 1: Fitness del mejor candidato de cada generación.

Como se puede ver en la figura 1, la fitness crece casi asintóticamente hasta alcanzar el resultado esperado.

2.1. Impacto del mutation rate

Es interesante ver los resultados obtenidos en la figura 2, pues al resolver el mismo problema con diferentes valores para el mutation rate se puede ver que el valor óptimo para este problema está entre 0.4 y 0.7.

Esto se explica porque un mutation rate muy bajo hace casi imposible el generar variaciones de los genes obtenidos inicialmente, mientras que uno muy alto no permite conservar los genes buenos de los mejores candidatos.

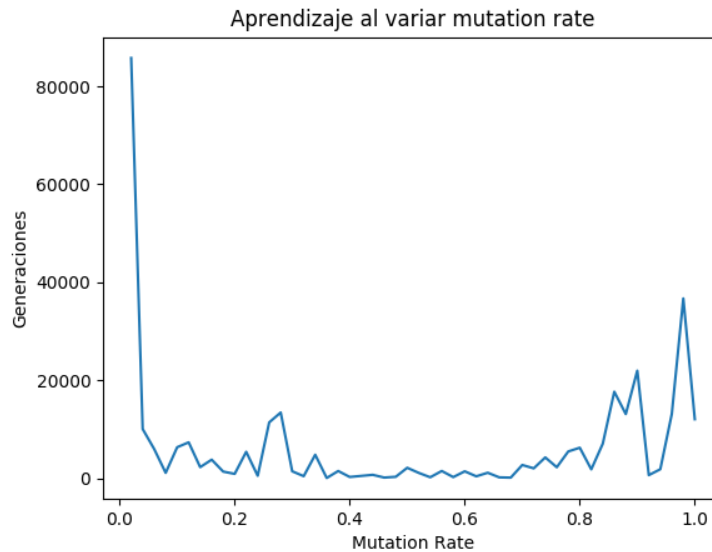


Figura 2: Generaciones que tarda el algoritmo en resolver N-Queen en un tablero de 8x8 al variar el mutation rate.

2.2. Variaciones del tamaño de las generaciones

Para evaluar el rendimiento del algoritmo al variar el tamaño de candidatos en cada generación no es buena idea utilizar como medida la cantidad de generaciones que se tarda en encontrar un resultado, pues obviamente la cantidad de generaciones disminuirá al aumentar su tamaño, pero mientras mas grande sea la generación mas cálculos tendrá que hacer el computador. Por esto mismo se utilizara el tiempo que tarda en resolverse el problema como medida.

Lo primero que llama la atención de la figura 3 es las diferentes lineas en las cuales se agrupan los puntos. En la mitad izquierda se pueden apreciar concentraciones en diferentes tiempos ocasionados por la cantidad de generaciones que tuvieron que hacerse para resolver el problema. Esto ocurre porque si no se genera la solución en una generación debe calcularse la siguiente completa.

También se puede ver en la mitad derecha que todos los puntos se agrupan en una nueva linea, esto es porque cuando la generación consiste en 2500 hijos se encuentra el resultado en la primera generación y no hay para que seguir avanzando.

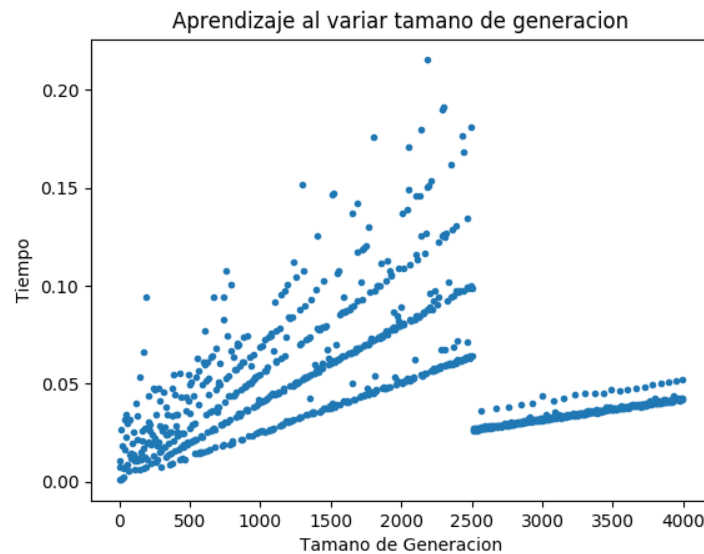


Figura 3: Tiempo de resolución de N-Queen con tablero de 5x5 en función del tamaño de las generaciones.

Se puede ver que los mejores resultados se obtienen con un tamaño entre 4 y 500, pues son los tiempos mas bajos. Si se escoge un tamaño mayor a 500 termina siendo mas rápido el buscar la solución por fuerza bruta, que es lo que hace el programa con generaciones de 2500 o mas elementos.

3. Anexo: Instrucciones de Ejecución

- Las instrucciones para ejecutar el proyecto se incluyen en el archivo README.txt
- El código fuente está en el repositorio <https://github.com/Mortup/RedesNeuronales>