N Queens - Hill Climbing

Sistemas Inteligentes - Doc. Víctor Manuel de la Cueva

José Manuel Beauregard Méndez

A01021716

Manual de Usuarios

Este manual de usuario contiene toda la información necesaria para correr y probar la funcionalidad de la entrega. Esta implementación utiliza librerías que ya vienen pre instaladas en la versión Python 3.0 o superior. Es importante mencionar que el resultado que arroja el algoritmo no es la única solución y pueden existir 1 o más combinaciones posible, este algoritmo regresa el primer máximo local que se encuentra.

A continuación se mostrará un ejemplo de casi de éxito:

```
 → py main.py
**********  Solution found in round 16692  **********
[0, 0, 0, 1, 0, 0, 0, 0]
[0, 0, 0, 0, 0, 0, 1, 0]
[1, 0, 0, 0, 0, 0, 0, 0]
[0, 0, 0, 0, 0, 0, 0, 1]
[0, 0, 0, 0, 1, 0, 0, 0]
[0, 1, 0, 0, 0, 0, 0, 0]
[0, 0, 0, 0, 0, 1, 0, 0]
[0, 0, 1, 0, 0, 0, 0, 0]
```

El nombre del archivo es opcional, solo necesita seguir el siguiente formato:

```
from Proyecto3 import busquedaHC

N = 8     # o el número de reinas que tenga el problema
lateral = False # o false
M = 38000     # o el número máximo que se desee reiniciar
busquedaHC(N, lateral, M)
```

La función busquedaHC recibe tres argumentos:

- N: número que representa la cantidad de reinas asi como el tamaño del tablero siendo NxN
- lateral: bandera booleana que dicta si se cambia de estado en caso de que la heursitica nueva sea igual a la pasada
- M: número de veces que se deberá repetir el código, reiniciado de manera aleatoria. Entre mayor sea N se recomiendo que M crezca para que se pueda encontrar una solución

Existen dos posibles tipos de resultados:

- El tablero con las quinas en su lugar sin atacarse entre sí. (Las quinas son representadas con 1)
- Un mensaje de aviso diciendo que no se encontró alguna solución