



Autor: José Coronel Camacho

Asignatura: Modelos de Inteligencia Artificial

Centro: I.E.S La Marisma

Profesor/a: Águeda María López Moreno

Fecha de entrega: 01/11/2024

DESCRIPCIÓN BREVE

El problema de las N reinas consiste en colocar N reinas en un tablero de ajedrez de tamaño $N \times N$ de forma que ninguna pueda atacar a otra. Esto significa que no pueden estar en la misma fila, columna o diagonal. El objetivo es encontrar una manera de colocar todas las reinas sin que se amenacen entre sí.

Contenido

CÓDIGO EN PYTHON.....	2
EXPLICACIÓN GENÉRICA.....	2
CONCLUSIÓN.....	3

José Coronel Camacho

CÓDIGO EN PYTHON

Copia y pega el código en visual studio code y lo ejecuta con el comando python y el nombre de tu archivo, en mi caso python problemaNReinas.py.

```

1 # problemaNReinas.py
2 import random
3
4 # Configuración inicial
5 n = 8 # Tamaño del tablero y número de reinas
6
7 def intentar_solucion():
8     # Inicializa los marcadores de ataque
9     columna = [False] * n
10    diagonal_izquierda = [False] * (2 * n)
11    diagonal_derecha = [False] * (2 * n)
12
13    # Intentamos colocar las reinas fila por fila
14    for y in range(n):
15        # Generamos una lista aleatoria de columnas disponibles para esta fila
16        columnas_disponibles = [x for x in range(n) if not columna[x] and not diagonal_izquierda[x + y] and not diagonal_derecha[x - y + n - 1]]
17        random.shuffle(columnas_disponibles) # Baraja las columnas disponibles
18
19        # Si no hay columnas disponibles, indica que falló esta disposición
20        if not columnas_disponibles:
21            print(f"Reiniciando del tablero en la fila (y). No hay columnas válidas.")
22            return False
23
24        # Selecciona una columna aleatoria de las disponibles y coloca la reina
25        x = columnas_disponibles[0]
26        columna[x] = diagonal_izquierda[x + y] = diagonal_derecha[x - y + n - 1] = True
27        print(f"Reina colocada en fila (y), columna (x).")
28
29    # Si se logran colocar todas las reinas, se ha encontrado una solución
30    print("Se ha colocado una reina en cada fila sin conflictos.")
31    return True
32
33 # Realiza intentos hasta encontrar una solución válida
34 intentos = 1
35 while not intentar_solucion():
36     print(f"Intento (intentos) fallido. Reiniciando tablero completo.")
37     intentos += 1
38
39 # Indica que se ha encontrado una solución válida
40 print(f"¡Solución encontrada después de", intentos, "intentos!")
  
```

Reina colocada en fila 6, columna 0.
 Reinicio del tablero en la fila 7. No hay columnas válidas.
 Intento 13 fallido. Reiniciando tablero completo.
 Reina colocada en fila 0, columna 4.
 Reina colocada en fila 1, columna 6.
 Reina colocada en fila 2, columna 1.
 Reina colocada en fila 3, columna 5.
 Reina colocada en fila 4, columna 2.
 Reina colocada en fila 5, columna 0.
 Reina colocada en fila 6, columna 7.
 Reina colocada en fila 7, columna 3.
 Se ha colocado una reina en cada fila sin conflictos.
 ¡Solución encontrada después de 14 intentos!

EXPLICACIÓN GENÉRICA

Este código resuelve el problema de las N reinas en un tablero de ajedrez N x N mediante una estrategia de búsqueda aleatoria con reinicio. La idea central es colocar una reina en cada fila sin que se amenace a ninguna otra, eligiendo aleatoriamente entre las columnas válidas en cada fila. Si en algún momento una fila no tiene opciones válidas, el programa reinicia el tablero y comienza de nuevo desde la primera fila.

Durante el proceso:

1. Se elige aleatoriamente una columna en cada fila para colocar una reina que no esté amenazada por ninguna otra.
2. Se verifica que la columna y las diagonales no estén ocupadas, y si no hay conflictos, se coloca la reina.
3. Si no hay columnas válidas en una fila específica, el tablero se reinicia, y el programa vuelve a intentar desde la primera fila.
4. Este proceso se repite hasta que se encuentra una configuración en la que todas las reinas están colocadas sin conflictos.

El programa imprime mensajes de progreso que indican dónde se colocan las reinas, cuándo se reinicia el tablero y cuántos intentos se han realizado, manteniendo al usuario informado del estado actual de la búsqueda.

CONCLUSIÓN

Este código implementa una solución para el problema de las N reinas usando un método de búsqueda aleatoria que combina intentos sucesivos y retrocesos (backtracking con reinicio completo) hasta que se encuentra una configuración correcta. La estrategia de elegir columnas aleatoriamente permite explorar diferentes configuraciones posibles, y el reinicio total garantiza que, aunque pueda requerir varios intentos, eventualmente se encontrará una solución válida.