INTELIGENCIA ARTIFICIAL

Memoria de la práctica 3

Grupo Jueves B

Hecho por
GUILLERMO BAJO LABORDA, 842748@unizar.es

Universidad Zaragoza
1542

Escuela de Ingeniería y Arquitectura Universidad Zaragoza

Introducción y resumen

En esta tercera práctica de la asignatura de Inteligencia Artificial, abordamos problemas de búsqueda local y optimización. Utilizamos el paquete *aima.core.search.local* para explorar algoritmos como Hill-Climbing, Simulated Annealing y Genetic Algorithm, aplicándolos al clásico problema de las 8 reinas. Partimos del código preexistente en la clase *NQueensDemo* del paquete *aima.gui.demo.search*, donde ya se encontraban implementadas estas técnicas.

Se han implementado diversos métodos que realizan un determinado número de veces búsquedas como la de Hill-Climbing o Simulated Annealing a partir de estados iniciales del tablero aleatorios, devolviendo por pantalla las estadísticas correspondientes a las búsquedas, para analizar los resultados obtenidos en términos de eficacia y eficiencia en la resolución del problema.

Trazas del problema

Las trazas correspondientes a la ejecución del programa NQueensDemo son las siguientes:

```
NQueens HillClimbing con 1000 estados iniciales diferentes -->
Fallos: 84,30%
Coste medio fallos: 3,05
Exitos: 15,70%
Coste medio exitos: 4,01


Search Outcome=SOLUTION_FOUND
Final State=
Q-------
------Q-
----Q---
-------Q
-Q------
---Q----
-----Q--
--Q-----

Numero de intentos: 14
Coste medio fallos: 3,38
Coste exito: 48
Coste medio exito: 4


NQueens Simulated Annealing con 1000 estados iniciales diferentes -->
Parametros Scheduler: Scheduler (10, 0,1, 500)
Fallos: 47,90%
Coste medio fallos: 57,85
Exitos: 52,10%
Coste medio exitos: 45,44

Search Outcome=SOLUTION_FOUND
Final State=
-Q------
------Q-
----Q---
-------Q
Q-------
---Q----
-----Q--
--Q-----

Numero de intentos: 1
Fallos: 0
Coste exito: 48
```

```
GeneticAlgorithm
Parámetros iniciales:    Población: 50, Probabilidad mutación: 0.15
Mejor individuo=
----Q---
Q-------
-------Q
---Q----
-Q------
------Q-
--Q-----
-----Q-

Tamaño tablero = 8
Fitness = 28.0
Es objetivo = true
Tamaño de población = 50
Iteraciones = 784
Tiempo = 4751ms.
```

A continuación, se modificarán algunos de los parámetros de algunas funciones para observar cómo varían los resultados.

Primeramente, procedemos a modificar el tiempo en el que tardará en decrecer la temperatura:

```
NQueensDemo Simulated Annealing con 1000 estados iniciales diferentes -->
Parametros Scheduler: Scheduler (10, 0,1, 1000)
Fallos: 22,90%
Coste medio fallos: 81,82
Exitos: 77,10%
Coste medio exitos: 51,97
```

Se puede observar que el porcentaje de aciertos aumenta, aunque el coste medio de los fallos también lo hace.

Respecto al Genetic Algorithm, primero disminuimos la temperatura para ver cómo los resultados varían:

```
GeneticAlgorithm
Parámetros iniciales:    Población: 10, Probabilidad mutación: 0.15
Mejor individuo=
Q-------
----Q---
-------Q
-----Q--
--Q-----
------Q-
-Q------
---Q----
```

```
Tamaño tablero       = 8
Fitness              = 28.0
Es objetivo          = true
Tamaño de población = 10
Iteraciones          = 8276
Tiempo               = 2195ms.
```

Como se puede observar, las iteraciones aumentan significativamente.

En cambio, si aumentamos la población:

```
GeneticAlgorithm
Parámetros iniciales:    Población: 100, Probabilidad mutación: 0.15
Mejor individuo=
-----Q--
---Q----
Q-------
----Q---
-------Q
-Q------
------Q-
--Q-----

Tamaño tablero = 8
Fitness = 28.0
Es objetivo = true
Tamaño de población = 100
Iteraciones = 94
Tiempo = 2436ms.
```

Como es de esperar, las iteraciones disminuyen.