

TP6-1

Inteligencia Artificial

Universidad de Zaragoza

Marcos Ruiz García, 648045

19 de noviembre de 2015

1. Introducción

Se nos ha encargado realizar una serie de tareas para que el algoritmo de búsqueda Hill-Climbing permita realizar hasta 100 pasos cuando está en un llano para mejorar el porcentaje de éxito de manera notable.

2. Tareas

2.1. Tarea 1

Para la realización de la primera tarea se han realizado 1000 pruebas para sacar las estadísticas mostradas en la sección 2.1.1. Por cada iteración, para saber si se ha encontrado la solución se usa el siguiente booleano:

```
1 | solucionEncontrada = (search.getOutcome() == SearchOutcome.SOLUTION_FOUND);
```

Posteriormente se cuentan las veces que se ha encontrado la solución y se calcula el porcentaje de éxito. Además, para hallar el número de pasos, tanto en caso de encontrar la solución como en el caso contrario, se usa una instrucción similar a esta:

```
1 | pasos += Integer.parseInt(agent.getInstrumentation().getProperty("nodesExpanded"))  
   | - 1;
```

2.1.1. Salida

```
EJERCICIO1:  
HILL CLIMBING SEARCH:  
Porcentaje de éxito: 15 %  
Media de pasos en acierto: 4  
Media de pasos en fallo: 3
```

2.2. Tarea 2

Para la realización de la segunda tarea se han realizado 1000 pruebas como en la Tarea 1. En esta tarea se ha modificado el código del método:

```
1 | public List<Action> search(Problem p);
```

Se han añadido dos listas:

```

1 List<Node> listaNodosLlanos = new ArrayList<>();
2 List<Node> pendientesDeBorrar = new ArrayList<>();

```

Ambas sirven para eliminar los nodos repetidos dentro de una llanura y así evitar entrar en bucles dentro de esta. También se añadió la variable entera numLlanos para llevar la cuenta de nodos visitados cuando estamos en un llano la cual vuelve a 0 cada vez que salimos de un llano para prepararse al siguiente.

2.2.1. Salida

```

EJERCICIO2:
HILL CLIMBING SEARCH FLAT:
Porcentaje de éxito: 74 %
Media de pasos en acierto: 9
Media de pasos en fallo: 9

```

2.3. Tarea 3

Se desarrolló el código de la clase:

```

1 private static void nQueensHillClimbingSearchFlat(boolean ito);

```

Esta clase, tanto en el caso de que no entra en los llanos como en el caso de que sí, ejecuta en un bucle hasta que llega a una solución correcta. Tras esto, se puede observar que debido a no entrar en los llanos necesitamos numerosos intentos para llegar a una solución (eso sí, cuando encontramos una solución la tenemos muy cerca).

2.3.1. Salida

```

EJERCICIO3: REINICIAR HASTA ENCONTRAR LA SOLUCION

HILL CLIMBING SEARCH:
Action[name==moveQueenTo, location== ( 1 , 7 ) ]
Action[name==moveQueenTo, location== ( 4 , 0 ) ]
Action[name==moveQueenTo, location== ( 3 , 2 ) ]
Action[name==moveQueenTo, location== ( 6 , 1 ) ]
Action[name==moveQueenTo, location== ( 7 , 5 ) ]
Search Outcome=SOLUTION_FOUND
Final State=
----Q---
-----Q-
---Q-----
Q-----
--Q-----
-----Q
-----Q--
-Q-----

nodesExpanded : 6
nodesGenerated : 336
Porcentaje de éxito: 5 %
Pasos en acierto: 5
Media de pasos en fallo: 3

HILL CLIMBING SEARCH FLAT:
Action[name==moveQueenTo, location== ( 2 , 7 ) ]
Action[name==moveQueenTo, location== ( 1 , 4 ) ]
Action[name==moveQueenTo, location== ( 5 , 2 ) ]
Action[name==moveQueenTo, location== ( 3 , 3 ) ]
Action[name==moveQueenTo, location== ( 2 , 6 ) ]
Action[name==moveQueenTo, location== ( 6 , 7 ) ]
Action[name==moveQueenTo, location== ( 0 , 2 ) ]
Action[name==moveQueenTo, location== ( 2 , 1 ) ]
Action[name==moveQueenTo, location== ( 0 , 6 ) ]
Action[name==moveQueenTo, location== ( 3 , 5 ) ]
Action[name==moveQueenTo, location== ( 3 , 7 ) ]
Action[name==moveQueenTo, location== ( 6 , 3 ) ]

```

```

Action[name==moveQueenTo, location== ( 5 , 6 ) ]
Action[name==moveQueenTo, location== ( 0 , 2 ) ]
Search Outcome=SOLUTION_FOUND
Final State=
----Q---
--Q-----
Q-----
-----Q-
-Q-----
-----Q
-----Q--
---Q-----

nodesExpanded : 15
nodesGenerated : 840
Porcentaje de éxito: 100 %
Pasos en acierto: 14
No ha habido fallos

```

2.4. Tarea 4

La Tarea 4 consta de la realización de esta misma memoria.

3. Conclusión

Tras la realización de esta práctica nos hemos dado cuenta que el algoritmo de búsqueda Hill Climbing, aunque es muy rápido y la solución que encuentra la encuentra en un número de pasos muy reducido, al permitir a dicho algoritmo que pueda atravesar llanos de hasta 100 nodos, aumentamos drásticamente el porcentaje de éxito. Este aumento de porcentaje de éxito no es gratis, ya que las soluciones que encuentra suele ser en un mayor número de pasos.