Universidade de Évora
Departamento de Informática
Inteligência Artificial
Ano letivo 2019 - 2020

4º Trabalho Prático
N-Rainhas

Alunos:
Luís Ressonha - 35003
Rúben Teimas - 39868

Docente:
Paulo Quaresma

11 de Maio de 2020

Indice

1 Introdução

Este quarto trabalho prático tem como objetivo representar e resolver o problema das "**N-Raínhas**".

O problema das "**N-Rainhas**"é um problema de satisfação de restrições (*CSP*) no qual devem ser colocadas N rainhas num tabuleiro NxN sem se atacarem.
Uma rainha pode atacar qualquer outra rainha que se encontre na mesma linha, coluna ou diagonal que esta.

Este problema pode ser resolvido utilizando diferentes abordagens. Algumas delas são as permutações das rainhas no tabuleiro até que não existam ataques, a satisfação de restrições utilizando uma pesquisa por *backtracking* ou a satisfação de restrições utilizando pesquisa local (*Hill Climbing)*, tendo sido esta última utilizada por nós.

O objetivo do trabalho foi atingido com sucesso dado que conseguimos criar um programa que resolva o problema das **N-Rainhas** utilizando uma derivação do algortimo *Hill-Climbing*.

Para executar o trabalho é necessário entrar na ambiente Prolog, consultar o ficheiro rainhas e chamar o predicado resolve(N) em que N é o número de rainhas.

2 Desenvolvimento

2.1 Representação do problema

Para a representação do problema optámos por representar o tabuleiro como uma lista de tuplos.

Os tuplos da lista são compostos também eles por um tuplo e um valor, sendo este tuplo as coordenadas da posição e o valor o número de ataques.

(p(X,Y),H)

. Em que X representa a coluna, Y a linha e H a heurística, ou seja, o número de ataques que a rainha faz.

O estado inicial é uma representação do tabuleiro gerada de forma pseudo-aleatória, pois garante que no existem rainhas na mesma coluna, ficando assim por eliminar os conflitos nas linhas e nas diagonais.
O estado final é qualquer representação do tabuleiro que tenha, em todos os tuplos da lista, H=0, ou seja, nenhuma das rainhas tem ataques.

2.2 Abordagem ao problema

Para calcular a heurística de cada rainha criamos 2 predicados auxiliares: *diagonal/3* e *linha/2*.

Estes 2 predicados são utilizados em *calcularHeuristica/3* que calcula o número de ataques que uma rainha faz. Neste predicado optámos por usar um "truque", inicializando a heurística a -1 para que não conte consigo.
Como o predicado anterior só calcula a heurística para uma rainha criámos um predicado *heuristicaTabuleiro/3* que calcula a heurística de todas as rainhas.

Após calcularmos a heuristica do estado inicial aplicámos uma derivação do algoritmo Hill Climbing. Esta derivação foi feita sobre o algortimo Hill Climbing sem ciclos, fornecido pelo docente.

Inicialmente, para escolher a rainha a mover, tinhamos pensado escolher a que tivesse uma maior heurística, pois à partida iria convergir mais rápido, contudo, ao escolher sempre a rainha com maior heurística poderiamos ficar presos num minimo local.
Uma das formas de escapar a estes minimos locais seria, de vez em quando, escolher uma rainha aleatória para que o tabuleiro fosse ligeiramente alterado. Como o nosso conhecimento de Prolog não é muito vasto optámos por escolher sempre aleatoriamente uma rainha para mover, o que acaba por criar, à partida, um tempo de convergência mais lento.

No final da execução do programa printamos o tabuleiro enquanto lista e "graficamente", sendo os R a posição das rainhas.

O print grafico tem uma pequena particularidade: o tabuleiro é uma rotação da solução. Isto pode ser visto como um erro, mas considerámos uma questão de perspetiva..

2.3 Resultados obtidos

Para cada tamanho do tabuleiro fizemos a média temporal de 10 execuções.

Tamanho do Tabuleiro	Tempo Decorrido(segundos)
4x4	0,0092
5x5	0,0272
6x6	0,2694
7x7	0,0558
8x8	0,4766
9x9	0,424
10x10	1,3548
11x11	0,9192
12x12	3,324
13x13	5,9324
14x14	8,9076
15x15	8,5806
16x16	5,9542
17x17	11,0973
18x18	8,1563
19x19	10,5705
20x20	21,1285

3 Conclusão

Achamos que o objetivo final foi atingido dado que conseguimos resolver este problema com tempos aceitaveis. Em algumas ocasiões o programa "rebentou", mas foram bastante raras.

Este acontecimento deve-se à escolha aleatória da rainha a mover, o que nem sempre corre bem.