

Instituto Tecnológico de Aeronáutica

CTC-17

Projeto 2: buscas de melhorias iterativas e satisfação de restrições

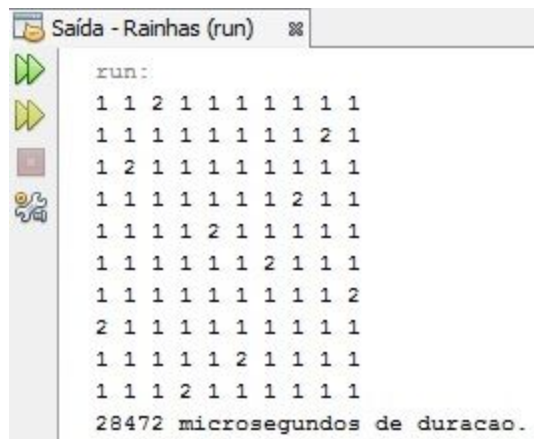
1. Nomes

Aluno: João Vitor Marques

Professor: Paulo André.

2. Resultados obtidos

2.1- Foi implementado o algoritmo Hill Climbing para solucionar o problemas das N rainhas. Cada '2' representa uma rainha e cada '1' representa casa sendo atacada por rainha. Obtiveram-se os seguintes resultados:



```
run:
1 1 2 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 2 1
1 2 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 2 1 1
1 1 1 1 2 1 1 1 1 1
1 1 1 1 1 1 2 1 1 1
1 1 1 1 1 1 1 1 2 1
2 1 1 1 1 1 1 1 1 1
1 1 1 1 1 2 1 1 1 1
1 1 1 2 1 1 1 1 1 1
28472 microsegundos de duracao.
```

Figura 1: 10 rainhas. 28 ms de duração

```

Saída - Rainhas (run)
run:
1 1 1 2 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 2 1 1 1
1 1 1 1 2 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 2 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 2 1 1
1 1 1 1 1 2 1 1 1 1 1 1 1 1 1
1 1 2 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 2 1
1 1 1 1 1 1 1 1 2 1 1 1 1 1 1
1 1 1 1 1 1 2 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 2
2 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 2 1 1 1 1 1 1 1
1 2 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 2 1 1 1 1 1 1
2218022 microsegundos de duracao.

```

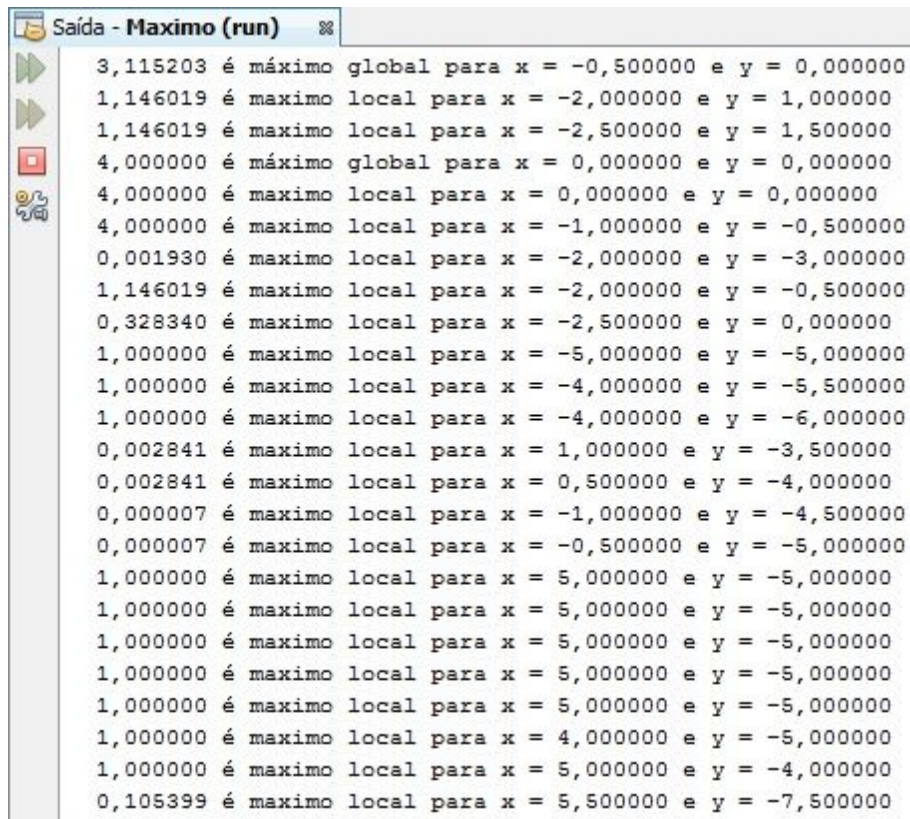
Figura 2: 15 rainhas. 2.22 segundos de duração

```

Saída - Rainhas (run)
run:
1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1
1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1
1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1
1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2
2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1
1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1
1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
522235338 microsegundos de duracao.

```

Figura 3: 20 rainhas. 8 min e 42 segundos de duração.

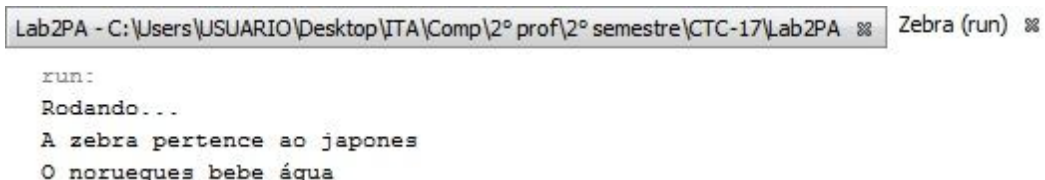


```
Saída - Maximo (run) %
3,115203 é máximo global para x = -0,500000 e y = 0,000000
1,146019 é maximo local para x = -2,000000 e y = 1,000000
1,146019 é maximo local para x = -2,500000 e y = 1,500000
4,000000 é máximo global para x = 0,000000 e y = 0,000000
4,000000 é maximo local para x = 0,000000 e y = 0,000000
4,000000 é maximo local para x = -1,000000 e y = -0,500000
0,001930 é maximo local para x = -2,000000 e y = -3,000000
1,146019 é maximo local para x = -2,000000 e y = -0,500000
0,328340 é maximo local para x = -2,500000 e y = 0,000000
1,000000 é maximo local para x = -5,000000 e y = -5,000000
1,000000 é maximo local para x = -4,000000 e y = -5,500000
1,000000 é maximo local para x = -4,000000 e y = -6,000000
0,002841 é maximo local para x = 1,000000 e y = -3,500000
0,002841 é maximo local para x = 0,500000 e y = -4,000000
0,000007 é maximo local para x = -1,000000 e y = -4,500000
0,000007 é maximo local para x = -0,500000 e y = -5,000000
1,000000 é maximo local para x = 5,000000 e y = -5,000000
1,000000 é maximo local para x = 5,000000 e y = -5,000000
1,000000 é maximo local para x = 5,000000 e y = -5,000000
1,000000 é maximo local para x = 5,000000 e y = -5,000000
1,000000 é maximo local para x = 5,000000 e y = -5,000000
1,000000 é maximo local para x = 4,000000 e y = -5,000000
1,000000 é maximo local para x = 5,000000 e y = -4,000000
0,105399 é maximo local para x = 5,500000 e y = -7,500000
```

Figura 5: resultado do algoritmo Têmpera Simulada

São encontrados vários máximos globais, mas o verdadeiro é o último encontrado. O algoritmo aplica os conceitos de aleatoriedade de escolha e foi rodado por 4 horas.

2.3 Atendendo a todas as restrições propostas, obteve-se o seguinte resultado aplicando-se o algoritmo de satisfação de restrições e backtracking:



```
Lab2PA - C:\Users\USUARIO\Desktop\ITA\Comp\2º prof\2º semestre\CTC-17\Lab2PA % Zebra (run) %
run:
Rodando...
A zebra pertence ao japones
O noruegues bebe água
```

Figura 6: resultado do problema da zebra.

3. Conclusões

3.1 O algoritmo Hill Climbing se mostrou um tanto quanto ruim para problemas com grande número de variáveis. Simples de implementar, mas ineficientes para grandes amostras.

3.2 Não necessariamente o algoritmo de Têmpera Simulada gera uma solução correta, pois adiciona-se certa aleatoriedade em suas escolhas além de que as soluções que o algoritmo encontra depende do tempo que o programa ficou rodando.

3.3 O algoritmo de backtracking seria mais eficiente se existissem mais restrições do tipo “o espanhol é dono do cachorro” que já definem o local da variável “cachorro” em “espanhol”, pois assim, seria necessário expandir menos filhos daquele nó.

4. Descrição da implementação

Todos os exercícios foram implementados em Java com a IDE NetBeans 8.0.2

4.1 Duas classes implementadas: “Rainhas”, que roda o algoritmo Hill climbing e a função Main e “Tabuleiro” que implementa o construtor para tabuleiros de tamanho variável e métodos que , por exemplo, checam se o tabuleiro contém rainhas que não se atacam. O algoritmo parte de um estado inicial e muda as rainhas de lugar (somente muda as linhas, as colunas permanecem as mesmas) até se chegar em um estado final, que acontece quando nenhuma rainha é atacada por outra.

4.2 Pela simplicidade da questão, foi implementada apenas uma classe com um método, que retorna o valor da função dado um ponto cartesiano (x, y). Para se verificar os “vizinhos” do ponto utilizava-se a passo de 0.5. Se esse passo aumentasse, o algoritmo ia perder precisão, se diminuísse, ele ganharia precisão, mas precisaria de um tempo maior para gerar resultados plausíveis.

4.3 Foram implementadas duas classes: “Zebra”, que contém a função principal e os métodos de satisfação de restrições e “Pessoa”, que armazena os parâmetros em questão(Nacionalidade, cor da casa, posição da casa, cigarro, bebida e animal). O método “atendeRestrições” da classe “Zebra” verifica se a solução cumpre os requisitos para se chegar a um resultado correto e o método “backtracking” é o responsável por alocar cada variável para cada morador.

Para simplificar a implementação usou-se a seguinte constituição:

Para cada atributo, existe uma correspondência:

- nacionalidade: 0-inglês, 1-espanhol, 2-norueguês, 3-ucraniano, 4-japones
- posição: 0-a primeira casa a esquerda, 4- a última casa a direita
- cor:0-vermelha, 1-amarelo, 2-azul, 3-verde, 4-marfim
- cigarro: 0-kool, 1-chesterfield, 2-winston, 3-lucky strike, 4-parliament
- bebida:0-água, 1-suco de laranja, 2-chá, 3-café, 4-leite
- animal:0-zebra, 1-cachorro, 2-raposa, 3-caramujos, 4-cavalo