Agente Solucionador para o Problema das N-Rainhas Utilizando Forward Checking com *most-constrained-variable* e *least-constraining-value*

Natanael Maia
COPPE/UFRJ

PESC
Programa de Engenharia
de Sistemas e Computação

N-Rainhas

- Posicionar N rainhas num tabuleiro (NxN) de maneira que nenhuma rainha ataque outra rainha
- O problema original, conhecido como 8-rainhas possui 92 soluções, sendo apenas 12 distintas

PESC
Programa de Engenharia
de Sistemas e Computação

N-Rainhas

N	Total de Soluções	Soluções Distintas
1	1	1
2	0	0
3	0	0
4	2	1
5	10	2
6	4	1
7	40	6
8	**92**	**12**
9	352	46
10	724	92
11	2,680	341
12	14,200	1,787
13	73,712	9,233
14	365,596	45,752
15	2,279,184	285,053
16	14,772,512	1,846,955
17	95,815,104	11,977,939
18	666,090,624	83,263,591
19	4,968,057,848	621,012,754
20	39,029,188,884	4,878,666,808
21	314,666,222,712	39,333,324,973
22	2,691,008,701,644	336,376,244,042
23	24,233,937,684,440	3,029,242,658,210

PESC
Programa de Engenharia
de Sistemas e Computação

N-Rainhas

- Tipos de Formulação:
  - Formulação incremental
  - Formulação de estado completo

Formulação Incremental

- Utiliza *Forward Checking*
  - A cada rainha posicionada, os valores conflitantes são retirados dos domínios das outras rainhas não posicionadas. Se não for possível posicionar uma rainha (dominio vazio), realiza *backtracking*
- Heurísticas:
  - ***Most-constrained-variable****:* escolhe a variável com o domínio mais restrito
  - ***Least-constraining-value:*** escolhe o valor que restringe menos os domínios das outras variáveis

PESC
Programa de Engenharia de Sistemas e Computação

Forward Checking

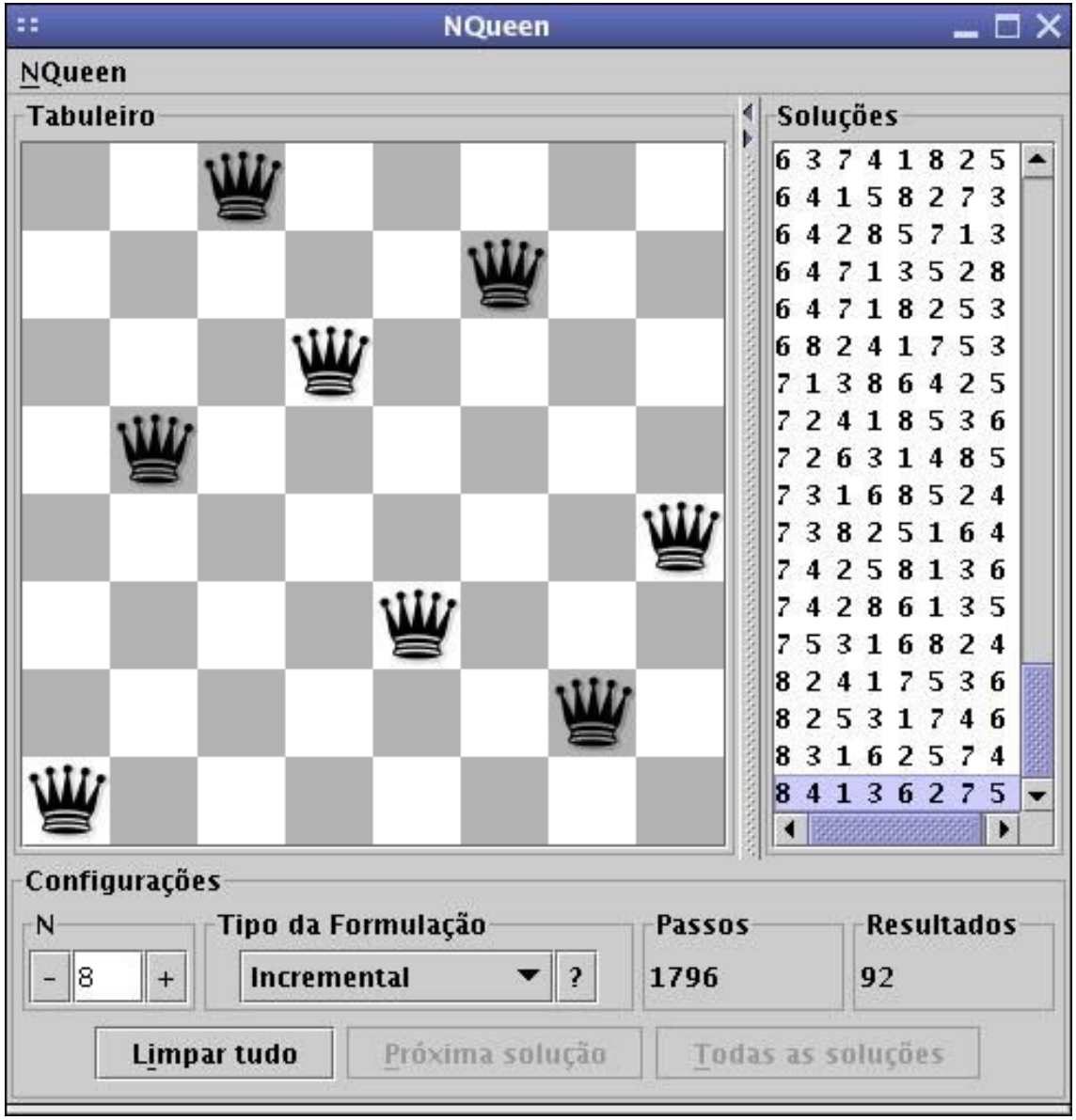
```
public Action[] search(Problem problem) {
    Node node = new Node(problem.getInitialState());
    nodes.insertElementsFront(expand(node));
    while (!nodes.isEmpty()) {
        node = (Node) nodes.removeFront();
        incrementSteps();
        if (problem.goalTest(node.getState())) {
            problem.setInitialState(node.getState());
            return node.getActionSequence();
        }
        nodes.insertElementsFront(expand(node));
    }
    return null;
}
```

PESC
Programa de Engenharia
de Sistemas e Computação

Forward Checking

- Soluções:

  most-constrained-variable



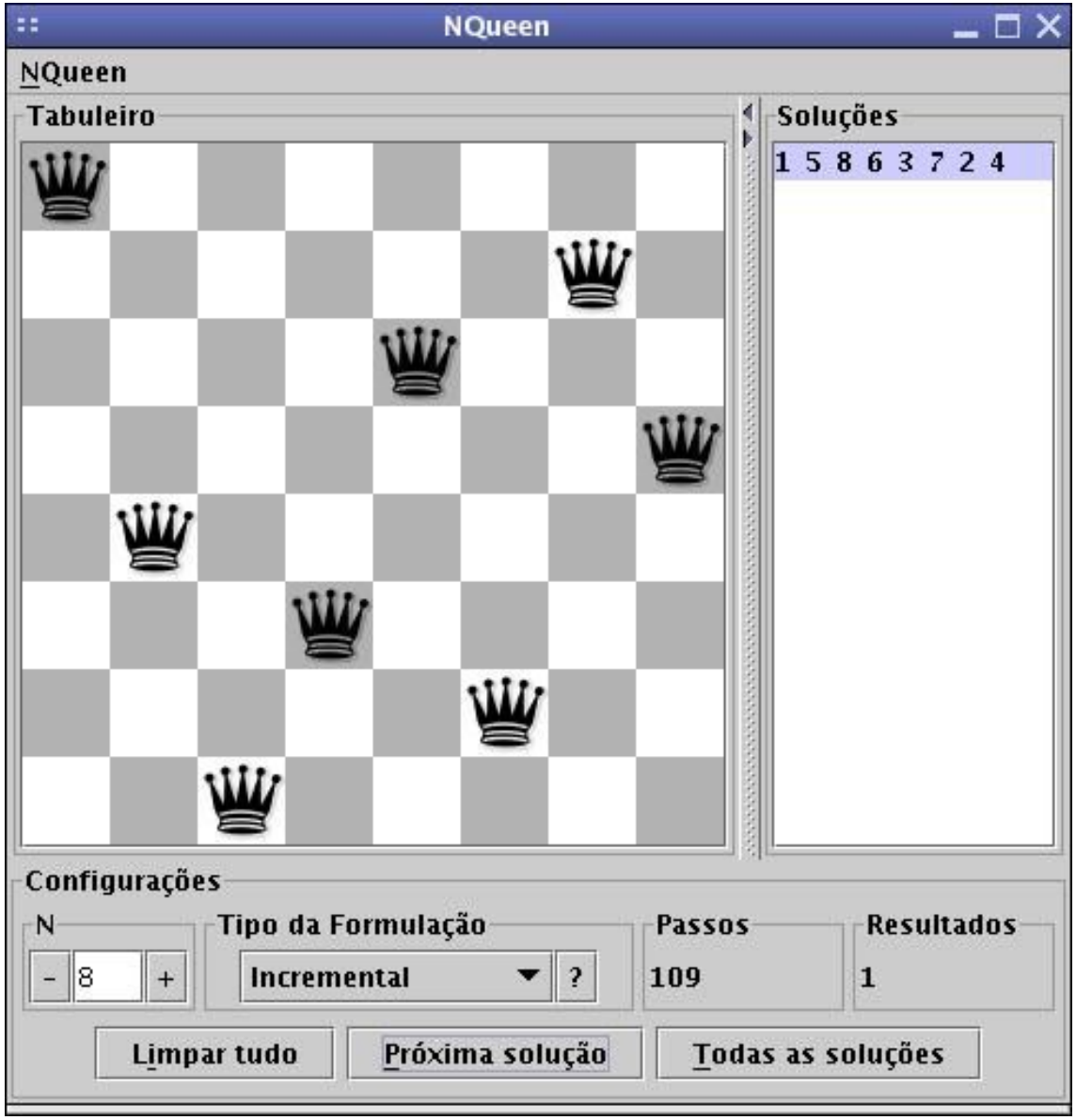
PESC
Programa de Engenharia
de Sistemas e Computação

Forward Checking

- Primeira solução:

  most-constrained-variable

- 109 passos



PESC
Programa de Engenharia
de Sistemas e Computação

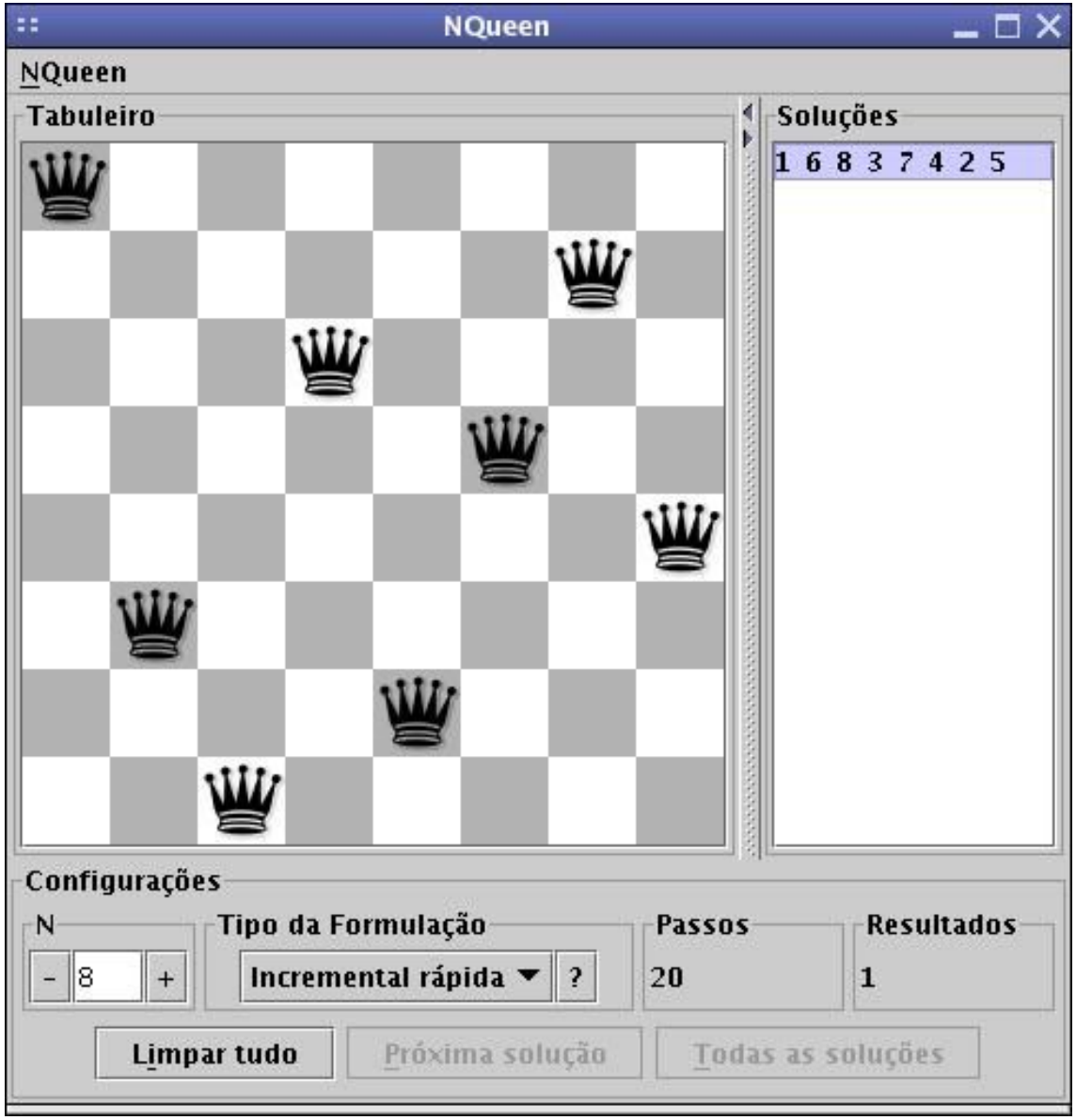
Forward Checking

- Primeira solução:

  most-constrained-variable

  least-constraining-value

- 20 passos



PESC
Programa de Engenharia de Sistemas e Computação

Comparativo

- Calculo do número de passos e tempo gasto para encontrar uma solução para cada N variando de 4 até 80
  - Utilizando somente *most-constrained-variable*:
    - 310.121 passos em aprox. 1m15s
  - Utilizando *most-constrained-variable* e *least-constraining-value:*
    - 19.401 passos em aprox. 4s
    - Atingiu N=155 em 30minutos

PESC
Programa de Engenharia
de Sistemas e Computação

Download

- http://www.cos.ufrj.br/~ntmaia/nqueen.zip