

# Proposta de solução do Problema de Dominação de Rainhas Utilizando ILS

## *Proposed solution to the Minimum Dominating Set of Queens Problem using ILS*

Maria Edoarda Vallim Fonseca

Thales Athayde Santos

2018, v-1.0.0

### Resumo

Neste artigo, propomos uma solução para o Problema de Dominação de Rainhas usando Busca Iterativa Local e comparamos nossos resultados com uma solução prévia utilizando Algoritmo Genético.

**Palavras-chave:** Problema de Dominação de Rainhas. Busca Iterativa Local. Metaheurística.

### Abstract

In this paper, we propose a solution to the Dominating Set of Queens Problem using Iterative Local Search and compare our results with a previous solution using Genetic Algorithm.

**Keywords:** Dominating Queen Problem. ILS. Metaheuristic.

**Data de submissão e aprovação:** elemento obrigatório. Indicar dia, mês e ano

**Identificação e disponibilidade:** elemento opcional. Pode ser indicado o endereço eletrônico, DOI, suportes e outras informações relativas ao acesso.

## 1 Introdução

O problema de dominação de rainhas é muito bem conhecido dentre os problemas de xadrez. Nele, dado um tabuleiro  $n \times n$ , temos  $n$  quadrados dispostos nas linhas e  $n$  quadrados nas colunas. Quando uma rainha  $Q$  é disposta no tabuleiro, ela domina a linha, a coluna, e as diagonais que passam pela sua posição. <sup>1</sup> O objetivo do problema é descobrir a disposição da menor quantidade de rainhas possível de forma à dominar todo o tabuleiro.

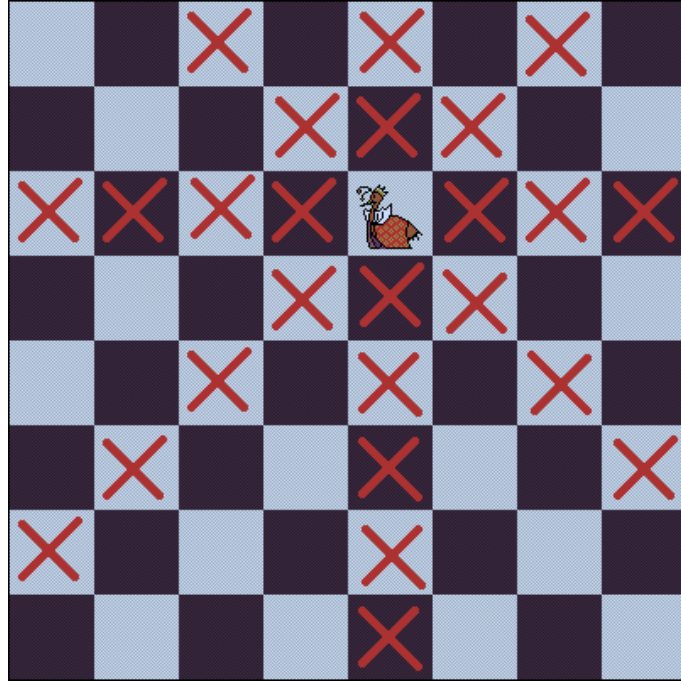


Figura 1 – Exemplo da linha de dominação de uma rainha em um tabuleiro de xadrez  $8 \times 8$

Algoritmos evolucionários provaram ter sucesso para resolver e otimizar uma grande variedade de problemas complexos, incluindo problemas combinatórios, como o estudado aqui, em um tempo computacional razoavelmente aceitável. (DOERR et al., 2011)

*Local Search*, ou Busca Local, é um método heurístico para resolver problemas computacionalmente difíceis. Busca local pode ser usada em problemas que possam ser formulados como achar a solução maximizando um critério entre várias soluções possíveis. Algoritmos de busca local movem de solução à solução no espaço de soluções possíveis aplicando mudanças locais, até uma solução dita ótima ser encontrada. (HOOS; STÜTZLE, 2004)

Um dos problemas da Busca Local é que ela pode ficar presa em um mínimo local, sem conseguir melhorar seu resultado. Para contornar esse problema, utiliza-se *Iterative Local Search*, ou Busca Local Iterativa. Essa modificação consiste em iterar sobre chamadas da busca local, cada vez começando de um ponto diferente do conjunto de solução perturbando o mínimo local atual de modo que faça a solução chegar em outro ótimo local. Esta perturbação não pode ser muito forte nem muito fraca, pois corre o risco dela acabar encontrando o mesmo mínimo local ou servir como uma inicialização aleatória. (LOURENÇO; MARTIN; STÜTZLE, 2010)

Neste artigo, propomos uma solução baseada em *Iterative Local Search* para o Problema de Dominação de Rainhas. Nossos resultados serão comparados diretamente com o método descrito em (ALHARBI; VENKAT, 2017), que utiliza Algoritmo Genético e é o mais comumente encontrado para solucionar este problema. Depois disso, iremos debater os resultados alcançados.



Figura 2 – Exemplo de um tabuleiro de tamanho  $8 \times 8$  sendo completamente dominado por quatro rainhas

## 2 Motivação

Decidimos escolher o *Dominating Set of Queens Problem* por ter sido um dos temas abordados por um dos membros do grupo durante as apresentações de trabalho da disciplina, o que nos deu um certo grau de familiaridade com o assunto. Pesquisando mais à fundo, vimos que as soluções mais comuns para a solução do Problema de Dominação de Rainhas eram com *Backtracking* e Algoritmo Genético. Além disso, de acordo com (ALHARBI; VENKAT, 2017), existem muitos artigos procurando os limites superior e inferior do problema, mas não existe muito esforço de pesquisa na busca de soluções práticas para o problema.

Visto essa situação, decidimos propor uma solução baseada em *Iterative Local Search* para o problema e comparar os resultados com uma solução em Algoritmo Guloso.

## 3 Trabalhos Relacionados

Um problema semelhante, proposto em 1850, conhecido como problema das  $n$ -rainhas teve muitos esforços focados nele para sua solução. O problema das  $n$ -rainhas é descrito como: dado um tabuleiro  $n \times n$ , qual seria a disposição das rainhas de modo que nenhuma rainha consiga atacar a outra. Houve muita pesquisa em torno deste problema, e suas soluções utilizam desde teoria matemática até teoria dos grafos. O estudo desse tipo de problema pode beneficiar várias áreas como controle de tráfego, prevenção de *deadlocks* e armazenamento de memória paralela. (BELL; STEVENS, 2009)

Muitas soluções para o problema das  $n$  rainhas foram propostas, dentre elas backtracking, redes neurais e algoritmos evolucionários.

Em contrapartida, o problema de dominação de rainhas não recebeu tanta atenção dentre os pesquisadores das ciências da computação. Na realidade, várias pesquisas se preocuparam em pesquisar o problema porém só adotaram modelos matemáticos. (ALHARBI; VENKAT, 2017) Burger e Mynhart apontaram que o problema de dominação das rainhas é um dos mais difíceis problemas de xadrez. (BURGER; MYNHARDT, 2002) Esse problema é comumente definido como achar o menor número possível de rainhas em um tabuleiro  $n \times n$  que consigam a dominação total do tabuleiro. Esse número é conhecido como número de dominação e sua notação é  $\gamma(\mathbf{Qn})$ . Muitos pesquisadores focaram em descobrir os limites superior e inferior do problema desde a década de 70, e o número mínimo possível de rainhas para solucionar o problema ( $\gamma(\mathbf{Qn})$ ) foi calculado para diversos tamanhos de tabuleiro (vide tabela 1) (COCKAYNE, 1990; BURGER; MYNHARDT, 2002; WEAKLEY, 2002; GIBBONS; WEBB, 1997).

Tabela 1 – Número mínimo de rainhas para dominação total  $\gamma(\mathbf{Qn})$  de um tabuleiro de tamanho  $n$ .

n	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
$\gamma(\mathbf{Qn})$	1	1	1	3	3	4	4	5	5	5	5	7	7	$\leq 8$	$\leq 9$	$\leq 9$	9	9

## 4 Methodology

O *Local Search* foi implementado com 1 de distância pela nossa implementação estar utilizando uma matriz e não um vetor contendo todas as posições do tabuleiro. Embora usar um *Local Search* com distâncias maiores gere resultados melhores, isso teria a consequência negativa de aumentar exponencialmente o tempo computacional da execução do algoritmo.

Pseudocódigo do ILS

```

Data: rainhas, tabuleiro
Result: how to write algorithm with LATEX2e
initialization;
listaMovimentos = [0..7];
melhorResultado;
while rainhas do
    embaralha(rainhas);
    if understand then
        go to next section;
        current section becomes this one;
    else
        go back to the beginning of current section;
    end
end

```

**Algorithm 1:** Pseudocódigo do algoritmo de ILS utilizado

## 5 Resultados

Nossos testes foram rodados em uma máquina Intel Core i5-7200U com 8GB de RAM, usando o sistema operacional Manjaro Linux com o pacote gráfico KDE Plasma. A

**Data:** rainhas, tabuleiro  
**Result:** how to write algorithm with L<sup>A</sup>T<sub>E</sub>X2e  
initialization;  
listaMovimentos = [0..7];  
melhorResultado;  
**while** *rainhas* **do**  
    embaralha(rainhas);  
    **if** *understand* **then**  
        go to next section;  
        current section becomes this one;  
    **else**  
        go back to the beginning of current section;  
    **end**  
**end**

**Algorithm 2:** Pseudocódigo do Algoritmo Genético utilizado

linguagem de programação utilizada foi Python 3.7.1.

## 6 Conclusão

Text...

## Referências

- ALHARBI, S.; VENKAT, I. A genetic algorithm based approach for solving the minimum dominating set of queens problem. *Journal of Optimization*, Hindawi, v. 2017, 2017. Citado 3 vezes nas páginas 2, 3 e 4.
- BELL, J.; STEVENS, B. A survey of known results and research areas for n-queens. *Discrete Mathematics*, Elsevier, v. 309, n. 1, p. 1–31, 2009. Citado na página 3.
- BURGER, A. P.; MYNHARDT, C. M. An upper bound for the minimum number of queens covering the  $n \times n$  chessboard. *Discrete Applied Mathematics*, Elsevier, v. 121, n. 1-3, p. 51–60, 2002. Citado na página 4.
- COCKAYNE, E. J. Chessboard domination problems. *Discrete Mathematics*, Elsevier, v. 86, n. 1-3, p. 13–20, 1990. Citado na página 4.
- DOERR, B. et al. Evolutionary algorithms and dynamic programming. *Theoretical computer science*, Elsevier, v. 412, n. 43, p. 6020–6035, 2011. Citado na página 2.
- GIBBONS, P. B.; WEBB, J. Some new results for the queens domination problem. *Australasian Journal of Combinatorics*, CENTRE FOR COMBINATORICS, v. 15, p. 145–160, 1997. Citado na página 4.
- HOOS, H. H.; STÜTZLE, T. *Stochastic local search: Foundations and applications*. [S.l.]: Elsevier, 2004. Citado na página 2.

LOURENÇO, H. R.; MARTIN, O. C.; STÜTZLE, T. Iterated local search: Framework and applications. In: *Handbook of metaheuristics*. [S.l.]: Springer, 2010. p. 363–397. Citado na página [2](#).

WEAKLEY, W. D. Upper bounds for domination numbers of the queen's graph. *Discrete mathematics*, North-Holland, v. 242, n. 1-3, p. 229–243, 2002. Citado na página [4](#).