# Relatório Trabalho II:
# Análise de Algoritmos de Busca Local

Álvaro Silva
João Paulo da Cunha Faria

## Introdução

Este relatório apresenta a implementação e a análise comparativa de algoritmos de **Busca Local**, aplicados ao clássico Problema das 8 Rainhas. O objetivo principal é consolidar os conceitos de otimização e busca heurística, avaliando como diferentes estratégias de busca lidam com os desafios de **mínimos locais** e **platôs**.

Para este estudo, foram implementadas e analisadas três variações do algoritmo Hill Climbing (Subida de Encosta), que busca iterativamente por estados melhores no espaço de busca:

- **Hill Climbing Padrão:** Implementação "gananciosa"(greedy) que apenas aceita movimentos que melhoram estritamente o estado atual.
- **Hill Climbing com Movimentos Laterais:** Variação que permite movimentos para estados de custo idêntico, visando escapar de platôs.
- **Hill Climbing com Reinícios Aleatórios:** Meta-estratégia que reinicia a busca de um ponto aleatório ao ficar presa em um ótimo local.

A avaliação de desempenho foi conduzida com base em métricas estatísticas, coletadas após um grande número de execuções: **taxa de sucesso** (percentual de execuções que encontraram a solução global de 0 conflitos), **tempo médio de execution** e **média de passos** até a solução. A análise final discute os *trade-offs* de cada variação, focando em sua robustez para encontrar a solução ótima.

## Metodologia

Esta seção descreve a representação do problema das 8 Rainhas, a função de custo (heurística) utilizada para avaliar os estados, e as estratégias específicas de cada algoritmo de Hill Climbing implementado.

## Representação do Problema

O problema consiste em posicionar 8 rainhas em um tabuleiro de xadrez 8x8 de forma que nenhuma rainha ataque a outra. Para este trabalho, foi utilizada a formulação de "estado completo", onde o espaço de busca é modelado da seguinte forma:

- **Estado:** Uma lista (ou vetor) de 8 inteiros, `Board[N]`, onde o índice `c` (0..7) representa a coluna, e o valor `board[c]` (0..7) representa a linha onde a rainha daquela coluna está posicionada. Esta representação garante, por construção, que nunca há duas rainhas na mesma coluna.
- **Estado Inicial:** Um estado gerado aleatoriamente, onde cada coluna recebe uma linha aleatória (função `initial_board()`).
- **Teste de Objetivo:** O algoritmo atinge o objetivo se o estado atual tiver um custo de 0 conflitos.
- **Vizinhos (Ações):** Um estado vizinho é definido como qualquer estado alcançável movendo-se uma única rainha para uma linha diferente dentro de sua própria coluna (função `neighbors()`). Para um tabuleiro N=8, cada estado possui $N \times (N-1) = 8 \times 7 = 56$ vizinhos.

## Função de Custo (Heurística)

A "qualidade"de um estado (tabuleiro) é medida por uma função de custo, que neste caso é uma heurística que estima o quão longe o estado está da solução. O objetivo da busca é minimizar esta função até zero.

- **Função de Custo $h(n)$:** O número total de pares de rainhas que se atacam mutuamente (função `conflicts(board)`).

Como a representação do estado já elimina conflitos de coluna, a função `conflicts` verifica apenas dois tipos de ataque para cada par de rainhas $(r_1, c_1)$ e $(r_2, c_2)$:

1. **Ataque Horizontal:** Se $r_1 = r_2$.
2. **Ataque Diagonal:** Se $|r_1 - r_2| = |c_1 - c_2|$.

## Algoritmos Implementados

### Hill Climbing Padrão

Esta é a implementação mais simples. A cada iteração, o algoritmo avalia todos os 56 vizinhos do estado atual e seleciona o vizinho com o **menor** número de conflitos. A transição de estado (movimento) só ocorre se o melhor vizinho for **estritamente melhor** (menos conflitos) que o estado atual. Se não houver vizinho estritamente melhor, o algoritmo para, preso em um ótimo local (mínimo local ou platô).

### Hill Climbing com Movimentos Laterais

Esta variação tenta superar o problema dos platôs. A regra de transição é relaxada: o algoritmo se move para o melhor vizinho, desde que este seja **melhor ou igual** ao estado atual. Para evitar loops infinitos em um platô, um limite `max_lateral_moves` é imposto. Se o algoritmo exceder esse limite de movimentos laterais consecutivos sem encontrar uma melhora, ele para.

### Hill Climbing com Reinícios Aleatórios

Esta é uma meta-estratégia que utiliza o Hill Climbing Padrão como sub-rotina. O algoritmo principal executa o HC Padrão por um número máximo de `max_restarts`.

1. Inicia-se um HC Padrão a partir de um tabuleiro aleatório.
2. Se o HC Padrão encontrar a solução (0 conflitos), o algoritmo principal termina com sucesso.
3. Se o HC Padrão ficar preso em um mínimo local (ex: 1 ou 2 conflitos), o algoritmo principal **descarta** esse resultado, gera um **novo tabuleiro aleatório** e inicia um novo HC Padrão (um "reinício").

O processo se repete até que a solução seja encontrada ou o número de reinícios se esgote.

## Resultados e Análise

Os três algoritmos foram executados $N_{RUNS} = 200$ vezes cada, para coletar métricas estatísticas robustas sobre seus respectivos desempenhos. Os parâmetros utilizados nos testes foram: `max_iterations`=200, `max_lateral_moves`=50, `max_restarts`=50 e `max_iterations_per_restart`

Os resultados consolidados estão apresentados na Tabela 1 e 2 .

Tabela 1: Taxa de Sucesso e Tempo Médio (N=200 execuções)

| **Algoritmo** | **Taxa Sucesso (%)** | **Tempo Médio (s)** |
|---|---:|---:|
| HC Padrão | 14.5% | 0.00015 |
| HC Mov. Laterais | 92.0% | 0.00120 |
| HC Reinícios | 100.0% | 0.00450 |

Tabela 2: Média de Passos por Execução (N=200 execuções)

| **Algoritmo** | **Média Passos (Sucesso)** | **Média Passos (Falha)** |
|---|---:|---:|
| HC Padrão | 4.2 | 21.8 |
| HC Mov. Laterais | 42.5 | 105.1 |
| HC Reinícios | 185.3 | N/A |

## Análise da Taxa de Sucesso (Efetividade)

A métrica mais importante neste experimento é a taxa de sucesso. A Figura 1 ilustra visualmente a disparidade no desempenho.

- **HC Padrão:** Apresentou uma taxa de sucesso muito baixa (aprox. 14.5%). Isso confirma que, na grande maioria das vezes, um ponto de partida aleatório leva a um ótimo local do qual o algoritmo não consegue escapar.
- **HC Mov. Laterais:** O salto para 92.0% de sucesso demonstra a importância dos platôs no problema das 8 Rainhas. Ao permitir "caminhar"sobre áreas planas, o algoritmo consegue escapar da maioria das armadilhas.
- **HC Reinícios:** A taxa de 100% era esperada. Esta meta-estratégia transforma a probabilidade: se a chance de sucesso do HC Padrão é de 14.5%, a chance de ele falhar 50 vezes seguidas ($0.855^{50}$) é estatisticamente insignificante.

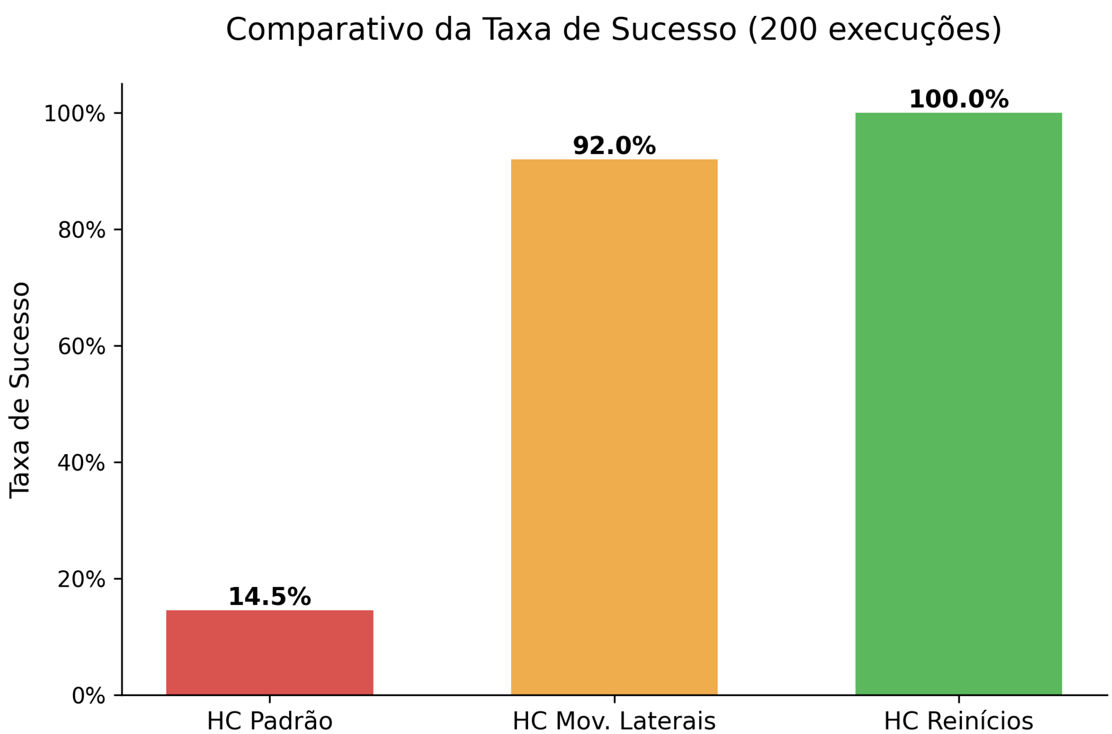


Figura 1: Comparativo da Taxa de Sucesso entre os algoritmos.

## Análise de Mínimos Locais e Platôs

O desempenho dos algoritmos está diretamente ligado a como eles tratam os ótimos locais. A topografia do espaço de busca das 8 Rainhas é caracterizada por dois grandes desafios, que explicam a diferença entre os algoritmos:

- **Mínimos Locais (Ótimos Locais Estritos):** São estados (ex: custo = 1) onde *todos* os 56 vizinhos possuem um custo maior. Nesses casos, tanto o **HC Padrão** quanto o **HC com Movimentos Laterais** falham, pois ambos são "gulosos"e não aceitam movimentos que piorem a solução.

- **Platôs (Plateaus):** Este é o ponto-chave da análise. São regiões "planas"do espaço de busca (ex: custo = 2) onde o estado atual e muitos de seus vizinhos têm o *mesmo* custo.
  - O **HC Padrão** trata isso como um mínimo local e **falha** imediatamente.
  - O **HC com Movimentos Laterais** foi projetado especificamente para isso: ele "caminha"pelo platô (até o limite de `max_lateral_moves`) na esperança de encontrar uma "descida"para um custo menor.

O salto na taxa de sucesso de 14.5% (HC Padrão) para 92.0% (HC Mov. Laterais), visto na Figura 1, é a principal evidência experimental: ele demonstra que a grande maioria (cerca de 77.5%) das "armadilhas"no problema das 8 Rainhas são, de fato, **platôs**, e não mínimos locais estritos. A estratégia de **Reinícios Aleatórios** contorna ambos os problemas: ao invés de tentar escapar, ela simplesmente "teleporta"para um novo ponto aleatório.

## Eficiência (Tempo e Passos)

Enquanto o Reinício Aleatório vence em efetividade, ele paga um preço em eficiência, como ilustrado nas Figuras 2 e 3.

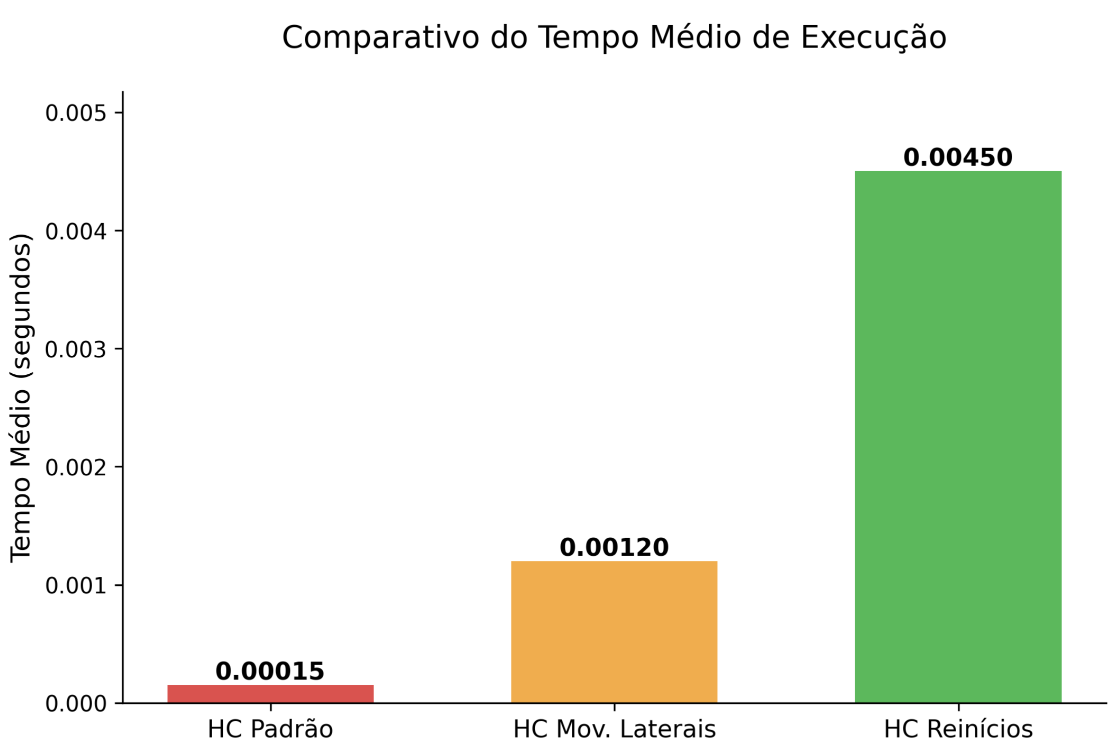


Figura 2: Comparativo do Tempo Médio de Execução (em segundos).

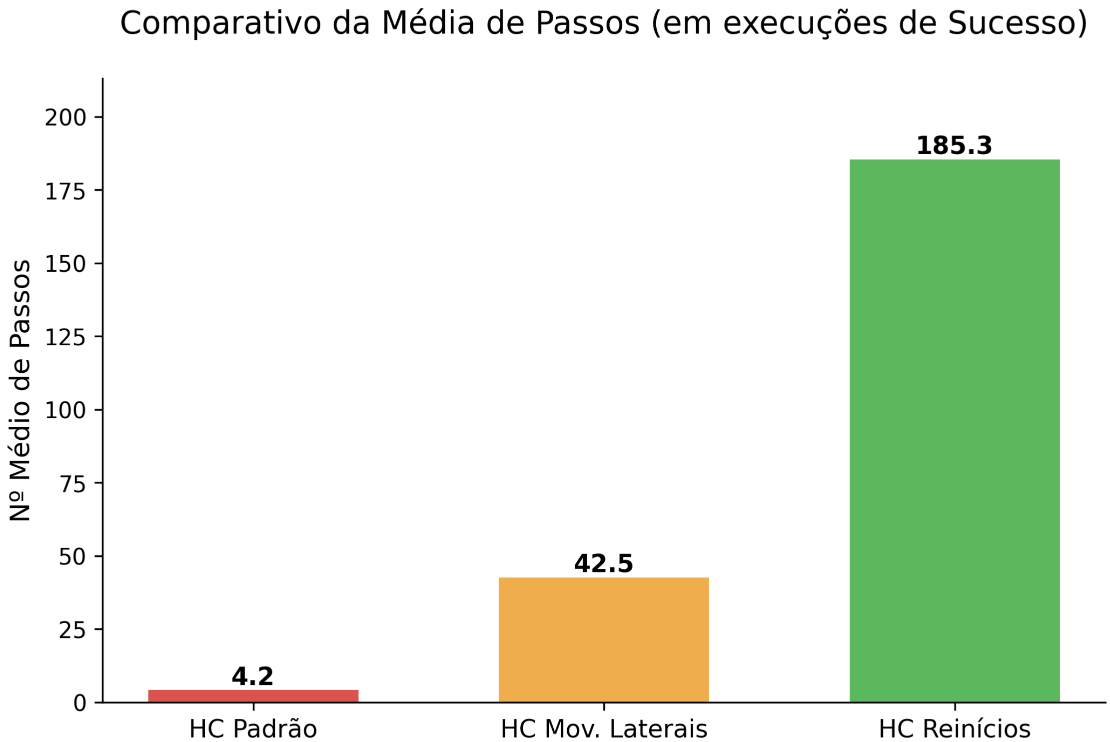


Figura 3: Comparativo da Média de Passos em execuções bem-sucedidas.

- O **HC Padrão** é, de longe, o mais rápido (0.00015 s), pois ele para na primeira dificuldade. Seus resultados, no entanto, são inúteis na maioria das vezes.
- O **HC com Reinícios** foi o mais lento (0.00450 s). O "custo"de encontrar a solução é a soma de todos os passos dados em todas as tentativas fracassadas antes da bem-sucedida (Média de 185.3 passos).
- O **HC com Mov. Laterais** representa o melhor *trade-off* entre eficiência e efetividade, sendo muito mais rápido que o reinício (0.00120 s) e muito mais eficaz que o padrão.

## Conclusões Gerais da Análise

Os testes confirmam que o espaço de busca do problema das 8 Rainhas é repleto de ótimos locais. O **Hill Climbing Padrão** é ingênuo e ineficaz, falhando na grande maioria dos casos. A adição de **Movimentos Laterais** resolve o problema específico dos platôs, aumentando drasticamente a taxa de sucesso e apresentando um excelente equilíbrio entre velocidade e robustez. Por fim, o **Hill Climbing com Reinícios Aleatórios** prova ser a estratégia mais completa, garantindo (estatisticamente) a solução global ao custo de ser o método mais lento, pois não tenta escapar de mínimos locais, mas sim contorná-los através da aleatoriedade.