

# Prova 1 Bimestre Análise de Estratégias em IA

Vinicius Feld

RA: 2016204672

## 0.1 Introdução

Este artigo é baseado na análise de 4 problemas da programação: Cubo de Rubik, Missões canibais, Problema das N rainhas e Sudoku. O objetivo é apresentar o melhor algoritmo para resolução dos 4 problemas citados acima, realizar uma análise de funcionamento de cada algoritmo e como será aplicado o algoritmo em cada um dos problemas, no objetivo de resolvê-los da melhor forma possível.

## 0.2 Fundamentação Teórica

### 0.2.1 Componentes para Resolução dos Problemas

Para acharmos o melhor algoritmo para a resolução dos 4 problemas, precisamos analisar 5 componentes de cada problema:

- **Estado Inicial** : É o estado onde o agente começa, estado inicial.
- **Ações**: As ações que são executadas nesse problema (juntamente com as regras que devem ser seguidas para a execução de cada ação).
- **Modelo de transição**: Estado resultante

ao aplicar uma ação a um estado específico.

- **Teste de objetivo**: Determina qual estado é o estado objetivo do problema.
- **Custo de caminho**: Custo necessário do caminho de cada execução.

### 0.2.2 Algoritmos para Resolução dos Problemas

Para uma melhor solução de cada problema foi definido os seguintes algoritmos:

- **Cubo de Rubik** - Algoritmo A\*
- **Missionários e canibais** - Busca em Profundidade
- **Problema das n rainhas** - HillClimbing
- **Sudoku** - HillClimbing

### 0.2.3 Algoritmo A\*

É um algoritmo para Busca de Caminho. Ele busca o caminho em um grafo de um vértice inicial até um vértice final. Sua aplicação vai desde aplicativos para encontrar rotas de deslocamento entre localidades a resolução de problemas.

## 0.2.4 Busca em Profundidade

O algoritmo de busca em profundidade varre (= scans) um grafo e visita todos os seus vértices e todos os seus arcos. À medida que varre o grafo, o algoritmo pinta cada vértice de cinza e depois de preto. Quando descobre um novo vértice, o algoritmo pinta o vértice de cinza; quando termina de visitar todos os vizinhos do vértice, o algoritmo pinta o vértice de preto. Os vértices cinza são considerados ativos e os pretos são considerados mortos.

## 0.2.5 HillClimbing

É um algoritmo iterativo que começa com uma solução arbitrária para um problema, depois tenta encontrar uma solução melhor fazendo uma mudança incremental para a solução. Se a mudança produzir uma solução melhor, outra mudança incremental é feita para a nova solução, e assim por diante até que nenhuma outra melhoria seja encontrada.

## 0.3 Análise dos Problemas

### 0.3.1 Cubo de Rubik

O Cubo de Rubik consiste em 6 fases com 6 cores diferentes, com cada face sendo dividida em 9 partes 3x3. Para chegar a resolução do problema, é necessário que as 9 partes de cada face estejam com apenas 1 cor por face.

O algoritmo A\* é o melhor algoritmo para solucionar este problema, pois possui estratégias para resolução do problema que é conhecido como estado final do cubo mágico, além disso para o algoritmo A\* é irrelevante a quantidade de movimentos que serão executados para chegar ao estado final.

Modelagem do problema:

- **Estado inicial:** cubo embaralhado.
- **Ações:** girar as faces nos 12 possíveis sentidos.
- **Modelo de transição:** mover as peças no sentido diagonal ou vertical.
- **Teste de objetivo:** todas as faces com a mesma cor.
- **Custo de caminho:** 1

### 0.3.2 Missionários e canibais

O problema começa com 3 missionários e 3 canibais viajando juntos, com o objetivo de chegar as margens de um rio. A única forma de atravessar é usando um barco que possui capacidade de levar 2 pessoas por vez, onde em nenhum momento o número de canibais pode ser maior que o número de missionários nas duas margens do rio.

O algoritmo que melhor se encaixa como solução deste problema é de Busca em profundidade, pois para resolver este problema, é necessário verificar todos os pontos existentes, onde 2 canibis não podem ficar juntos com 1 missionário.

Modelagem do problema:

- **Estado inicial:** 3 missionários e 3 canibais juntos do mesmo lado do rio.
- **Ações:** mover até 2 pessoas para o lado esquerdo ou direito do rio através de um barco.
- **Modelo de transição:** ao movimentar, em nenhum momento numero de canibais pode ser superior ao de missionários nas margens do rio.
- **Teste de objetivo:** todos os missionários e canibais atravessaram o rio.

- **Custo de caminho:** 1

### 0.3.3 Problema das n rainhas

O problema consiste em dispor 8 rainhas em um tabuleiro 8x8, de maneira que nenhuma das 8 rainhas seja atacada pela outra. É necessário que uma mesma rainha não esteja na mesma linha, coluna ou diagonal de outra rainha.

O algoritmo que apresenta uma melhor solução para este problema é o HillClimbing, pois não é necessário chegar a um resultado com um menor custo possível. O objetivo é encontrar uma resolução correta para o problema, que pode ir variando de acordo com a criação.

Modelagem do problema:

- **Estado inicial:** tabuleiro vazio.
- **Ações:** colocar uma peça na posição x e y.
- **Modelo de transição:** colocar a peça em uma posição não ocupada e não atacada ou seja, mesma linha, coluna ou diagonal.
- **Teste de objetivo:** todas as 8 rainhas estão alocadas de forma que nenhuma esteja na mesma linha, coluna ou diagonal.
- **Custo de caminho:** 1

### 0.3.4 Sudoku

O Sudoku é um quebra-cabeças que é baseado na inserção lógica de números. O tabuleiro possui um formato 9x9, possuindo 9 quadrantes, onde é possível inserir números de 1 a 9 em cada quadrante. Para a resolução do problema, é necessário preencher todos esses quadrantes com números de 1 a

9, totalizando 81 no total, porém com certas condições, sendo elas:

- O mesmo número não pode se repetir em cada quadrante.
- Não é permitido inserir o mesmo número na horizontal do tabuleiro.
- Não é permitido preencher o mesmo número na vertical do tabuleiro.
- Os números permitidos são de 1 a 9.

A melhor solução para este problema novamente é o algoritmo HillClimbing, pois não é necessário chegar a um resultado com um menor custo possível, e sim encontrar uma resolução correta para o problema, que pode ser variado de acordo com a criação.

- **Estado inicial:** tabuleiro 9x9 parcialmente preenchido.
- **Ações:** inserir um número de 1 a 9 em uma posição x e y vazia.
- **Modelo de transição:** inserir o número de forma que o mesmo número não se repita na mesma linha, coluna ou quadrante 3x3.
- **Teste de objetivo:** tabuleiro completamente preenchido onde o mesmo número não se repete na mesma linha, coluna ou quadrante 3x3.
- **Custo de caminho:** 1

## 0.4 Considerações Finais

Nesse artigo foi possível observar 4 problemas clássicos, são eles: Cubo de Rubik, Missionários e Canibais, Problema das N rainhas e Sudoku. Observamos também como funciona cada um desses problemas e como

podemos resolver os mesmos com alguns algoritmos (Algoritmo A\*, Busca em profundidade, HillClimbing). Todos os algoritmos foram escolhidos para se encaixar melhor e trazer uma melhor solução para cada um dos problemas apresentandos.

## REFERÊNCIAS

José Roberto Gorski. *HillClimbing*. 2019. Disponível em: <<https://www.iaparatodos.com.br/post/algor%C3%ADmos-de-busca-e-otimiza%C3%A7%C3%A3o-hill-climbing#:~:text=Na%20an%C3%A1lise%20num%C3%A9rica%2C%20a%20Subida,mudan%C3%A7a%20incremental%20para%20a%20solu%C3%A7%C3%A3o>>.

Paulo Feofiloff. *Busca em profundidade*. 2020. Disponível em: <[http://https://www.ime.usp.br/~pf/analise\\_de\\_algoritmos/aulas/dfs.html](http://https://www.ime.usp.br/~pf/analise_de_algoritmos/aulas/dfs.html)>.

Raul Joaquim. *Problema das Oito Rainhas*. 2010. Disponível em: <<http://rauljcs.jimdofree.com/arquivos-ufrn/aedii/problema-das-oito-rainhas>>.

Thaís. *Os Missionarios e os Canibais*. 2010. Disponível em: <<http://eu-thais.blogspot.com/2010/08/os-missionarios-e-os-canibais.html>>.

Wikipédia. *Sudoku*. 2020. Disponível em: <<http://wpt.wikipedia.org/wiki/Sudoku>>.

(Raul Joaquim, 2010)

(Thaís, 2010)

(José Roberto Gorski, 2019)

(Wikipédia, 2020)

(Paulo Feofiloff, 2020)