



Trabalho 1: Buscas

SCCo230 - Inteligência Artificial

Antonio Carrilho Neto - Aluno UATI

Eduardo Souza Rocha - 11218692

Fábio Verardino de Oliveira - 12674547

Olavo Moraes Borges Pereira - 11297792

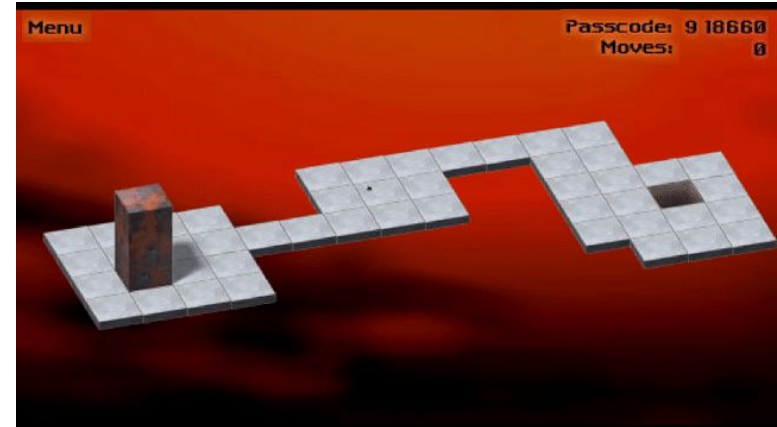


Introdução

- A tarefa de encontrar o conjunto de jogadas ótimo para um jogo é uma das aplicações mais comuns dos algoritmos de busca.
- Para demonstrar esse aplicação, implementamos um *solver* para o videogame *Bloxorz* (<https://bloxorz.io/>)
- Utilizamos mais de uma estratégia de busca, a fim de compará-las.

O Problema

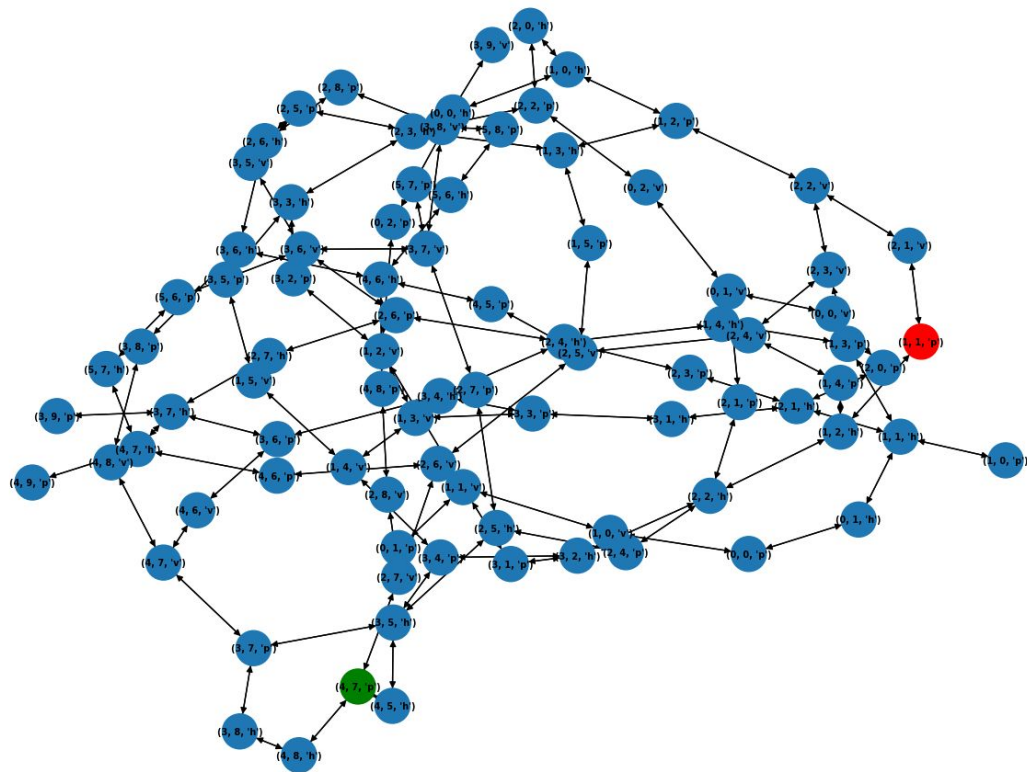
- O *Bloxorz* é um jogo simples, que consiste em tombar um paralelepípedo, levando ele até um ponto objetivo.
- O segredo são as manipulações da orientação do bloco (em pé ou deitado)





Modelagem

- A fase é representada por uma matriz, cujos elementos indicam o tipo daquela coordenada (chão, vazio, objetivo, etc...)
- O problema em si é representado por um grafo direcionado, no qual cada nó representa um possível estado do cubo (posição e orientação)
- As arestas do grafo representam o movimento necessário para ir de uma posição a outra (cima, baixo, direita, esquerda)
- Assim, para cada fase, geramos um grafo de todos os possíveis movimentos



Verde: objetivo
Vermelho: início



Busca

As estratégias de busca escolhidas para serem implementadas pela equipe foram:

- Busca em profundidade
- Busca em largura
- Busca A*
 - Com algumas heurísticas diferentes



Resultados: Busca em Profundidade

- Amplamente utilizada
 - Fácil implementação
-
- Nesse problema, costuma visitar muitos nós
 - Não encontra, necessariamente, a solução ótima



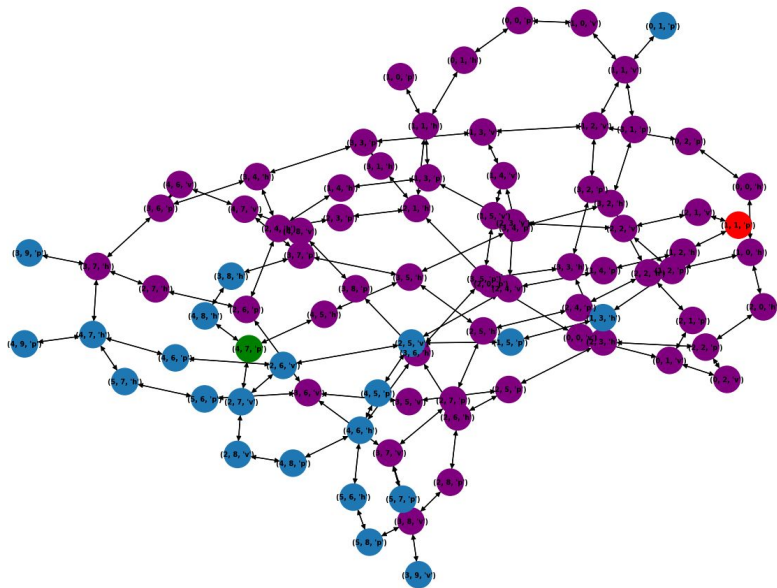
Resultados: Busca em Largura

- Checa todos os caminhos de tamanho 1, depois 2, assim por diante.
 - Característica desejável para esse problema (uma pequena manipulação no início pode mudar bastante os próximos passos)
- Visita aproximadamente o mesmo número de nós que a busca em profundidade
- Encontra a solução ótima!



Resultados: Busca A*

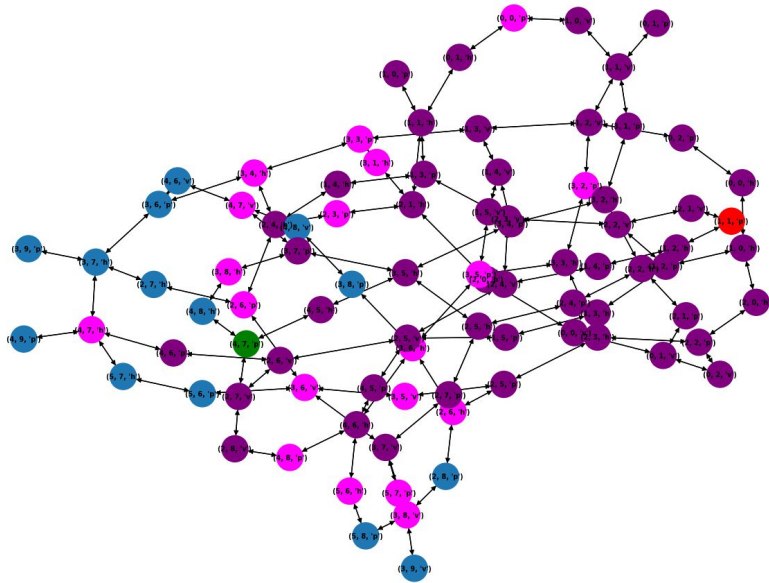
- Busca informada, leva em consideração uma função (heurística) de custo.
- Testamos com algumas heurísticas:
 - Distância de manhattan ao objetivo;
 - Metade da distância de manhattan ao objetivo;
 - Metade da distância euclidiana ao objetivo;
 - $h(n)=0$.
- Em geral, tão boa quanto ou melhor que a busca em largura (em nós visitados)
- Também encontra a solução ótima



Busca em profundidade:

- 67 nós visitados
- 50 passos na solução

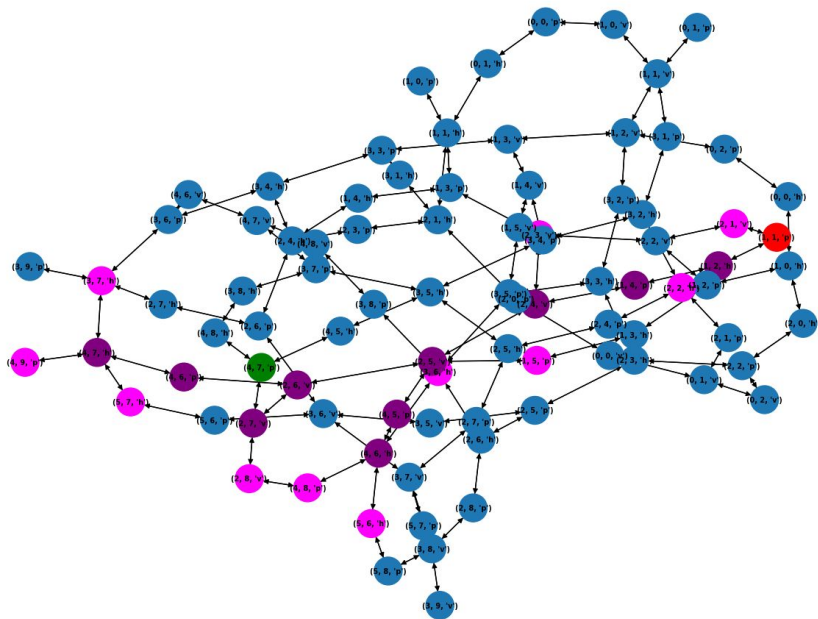
Verde: objetivo
Vermelho: início
Roxo: visitado
Azul: não visitado



Busca em largura:

- 56 nós visitados
- 7 passos na solução (ótima)

Verde: objetivo
Vermelho: início
Roxo: visitado
Azul: não visitado



Busca A* (manhattan):

- 28 nós visitados
- 7 passos na solução (ótima)

Verde: objetivo
 Vermelho: início
 Roxo: visitado
 Azul: não visitado



Fase 3		
Algoritmo de Busca	Tamanho da Solução	Número de Nós Visitados
Busca em Profundidade	49	100
Busca em Largura	19	97
Busca A* (Manhattan/2)	19	82
Busca A* (Euclidiana/2)	19	85
Busca A* (Trivial)	19	91
Busca A* (Manhattan)	19	68

Fim