



UNIVERSIDADE FEDERAL DO MARANHÃO
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA
BACHARELADO INTERDISCIPLINAR CIÊNCIA E TECNOLOGIA - BICT
INTELIGENCIA ARTIFICIAL

BEATRIZ PINHEIRO DE AZEVEDO - 2022023275

LUIS DA ASSUNÇÃO MAFRA MOURA - 2019050691

MARCELO ADRIEL CÂMARA ALMEIDA - 2020002392

MARIA EDUARDA PEREIRA LIMA - 2023033937

RELATÓRIO DO PROJETO 2:
MODELO DE BUSCA EM LARGURA

SÃO LUÍS - MA

2023

Sumário

Resumo.....	3
Introdução.....	4
Objetivos.....	4
Análise e Discussão.....	4
1. Estrutura do Código.....	4
2. Importação da Biblioteca.....	4
3. Função Principal.....	4
Passo 1: Inicialização da Fila.....	5
Passo 2: Laço Principal (Exploração de Estados).....	5
Passo 3: Iteração Sobre as Células do Tabuleiro.....	5
Passo 4: Simulação da Jogada.....	6
Passo 5: Verificação de Vitória.....	6
Passo 6: Inserção na Fila.....	6
Passo 7: Retorno Padrão.....	6
4. Pseudocódigo.....	7
5. Observações.....	7
6. Vantagens e Limitações.....	7
Vantagens:.....	7
Limitações:.....	7
Conclusão.....	8
Referências Bibliográficas.....	8

Resumo

Este relatório apresenta a análise de um algoritmo baseado na técnica de Busca em Largura (BFS) para o Jogo da Velha. O objetivo do código é encontrar a melhor jogada para o jogador atual, simulando todas as possibilidades no tabuleiro e identificando jogadas vencedoras de forma eficiente. Utilizando a estrutura de dados de fila, o algoritmo explora os estados do tabuleiro de maneira sistemática, garantindo que a primeira solução encontrada seja retornada.

A implementação foi detalhada em passos que incluem a inicialização da fila, a simulação de jogadas e a verificação de condições de vitória. A análise destaca as vantagens da abordagem, como sua exaustividade, e limitações, como a ausência de estratégias preditivas para jogadas futuras.

O trabalho demonstra a aplicabilidade de conceitos fundamentais de Inteligência Artificial e estruturas de dados em um cenário simples, mas relevante. Além disso, reforça a importância da exploração prática desses conceitos para o desenvolvimento de sistemas inteligentes. Sugestões para trabalhos futuros incluem a incorporação de estratégias heurísticas, ampliando o potencial do algoritmo para aplicações mais complexas.

Palavras-chave: Jogo da Velha, Busca em Largura, Inteligência Artificial, Estruturas de Dados, Algoritmos.

Introdução

O Jogo da Velha, um problema clássico de domínio finito e bem definido, é frequentemente utilizado como ferramenta didática para explorar algoritmos de busca e estratégias de decisão em Inteligência Artificial (IA). Este trabalho apresenta a análise de um algoritmo baseado na técnica de Busca em Largura (BFS) aplicado ao Jogo da Velha. A abordagem consiste em simular todas as possibilidades de jogadas a partir do estado atual do tabuleiro, utilizando uma estrutura de fila para garantir a exploração sistemática de estados. Esse método, embora exaustivo, assegura que a solução encontrada seja a mais próxima no espaço de busca, uma característica fundamental da BFS.

Ao detalhar o funcionamento do algoritmo, o trabalho busca evidenciar como conceitos fundamentais de IA e estruturas de dados podem ser utilizados para resolver problemas de forma eficiente. Além disso, destaca a importância da exploração prática desses conceitos como um caminho para compreender e desenvolver sistemas inteligentes aplicados a contextos mais complexos.

Objetivos

O objetivo do código é encontrar a melhor jogada para o jogador especificado, verificando todas as possibilidades disponíveis no tabuleiro. Se uma jogada vencedora for identificada, a função retorna as coordenadas dessa jogada. Caso nenhuma jogada vencedora seja possível, a função retorna None.

Análise e Discussão

1. Estrutura do Código

O código utiliza a estrutura de filas (queue) para explorar os estados do tabuleiro em um formato de busca por nível, onde:

- Cada estado do tabuleiro representa uma possibilidade de jogo.
- A função simula as jogadas do jogador e verifica se alguma delas leva à vitória.

2. Importação da Biblioteca

python

```
import queue
```

A biblioteca queue é usada para implementar uma fila, estrutura de dados necessária para a execução da Busca em Largura (BFS).

3. Função Principal

```
python
```

```
def encontrar_melhor_jogada(tabuleiro, jogador):
```

- Parâmetros:

- tabuleiro: Uma lista bidimensional (matriz 3x3) representando o estado atual do jogo.
- jogador: O símbolo do jogador atual ("X" ou "O").

- Retorno:

- Coordenadas (linha, coluna) da melhor jogada vencedora, se encontrada.
- None, caso não exista uma jogada que garanta a vitória imediata.

Passo 1: Inicialização da Fila

```
python
```

```
fila = queue.Queue()
```

```
fila.put(tabuleiro)
```

- Uma fila é criada para armazenar os possíveis estados do tabuleiro a serem explorados.
- O estado inicial do tabuleiro é colocado na fila.

Passo 2: Laço Principal (Exploração de Estados)

```
python
```

```
while not fila.empty():
```

```
    estado_atual = fila.get()
```

- O laço while continua até que a fila esteja vazia.
- A cada iteração, um estado do tabuleiro (estado_atual) é retirado da fila para ser processado.

Passo 3: Iteração Sobre as Células do Tabuleiro

python

```
for linha in range(3):  
    for coluna in range(3):  
        if estado_atual[linha][coluna] == " ":
```

- Duas estruturas de laço (for linha e for coluna) percorrem todas as posições no tabuleiro.
- Se uma célula estiver vazia (" "), ela é considerada como uma jogada potencial.

Passo 4: Simulação da Jogada

python

```
tabuleiro_simulado = [linha[:] for linha in estado_atual]  
tabuleiro_simulado[linha][coluna] = jogador
```

- Uma cópia do tabuleiro atual é criada para simular a jogada do jogador na posição vazia (linha, coluna).
- O símbolo do jogador (jogador) é colocado na posição simulada.

Passo 5: Verificação de Vitória

python

```
if verificar_vitoria(tabuleiro_simulado, jogador):  
    return linha, coluna
```

- A função auxiliar verificar_vitoria é chamada para verificar se a jogada simulada resulta em uma vitória.
- Caso a jogada seja vencedora, a função retorna as coordenadas (linha, coluna) imediatamente.

Passo 6: Inserção na Fila

python

```
fila.put(tabuleiro_simulado)
```

- Se a jogada simulada não for vencedora, o estado resultante (tabuleiro_simulado) é colocado na fila para ser explorado em iterações futuras.

Passo 7: Retorno Padrão

```
python  
return None
```

- Se a fila for esvaziada sem que uma jogada vencedora tenha sido encontrada, a função retorna None.

4. Pseudocódigo

O funcionamento pode ser resumido no seguinte pseudocódigo:

1. Inicializar uma fila com o estado inicial do tabuleiro.
2. Enquanto houver estados na fila:
 - Remover o próximo estado.
 - Para cada célula vazia no tabuleiro:
 - Simular a jogada do jogador na célula.
 - Verificar se a jogada é vencedora:
 - Se for, retornar as coordenadas.
 - Caso contrário, adicionar o estado simulado à fila.
3. Se nenhuma jogada vencedora for encontrada, retornar None.

5. Observações

- A função verificar_vitoria não está implementada no código fornecido, mas é essencial para determinar se o jogador venceu após uma jogada.
- O código presume que o tabuleiro é uma matriz 3x3, característica do Jogo da Velha.

6. Vantagens e Limitações

Vantagens:

- Explora todas as possibilidades de jogada de forma sistemática.
- Garante que a primeira jogada vencedora encontrada será retornada.

Limitações:

- Não otimiza para jogadas estratégicas além da vitória imediata (não avalia bloqueios ou futuras ameaças).
- O uso de BFS pode se tornar ineficiente em jogos com tabuleiros maiores ou mais complexos.

Conclusão

A implementação e análise do algoritmo apresentado para o Jogo da Velha demonstram a aplicação prática de conceitos fundamentais de Inteligência Artificial e estrutura de dados. Por meio da utilização da Busca em Largura (BFS), foi possível explorar todas as possibilidades de jogadas no tabuleiro de forma sistemática e eficiente, garantindo a identificação de jogadas vencedoras para o jogador atual.

A principal vantagem do método utilizado está na sua exaustividade, característica que assegura a busca por soluções completas dentro do espaço limitado de estados do jogo. No entanto, como destacado, a ausência de estratégias preditivas para jogadas futuras ou ações do adversário limita a aplicabilidade do algoritmo em situações mais complexas. Essa limitação é comum em algoritmos que priorizam a solução de problemas imediatos, conforme discutido por Rezende (2011) e Ziviani (2004).

Apesar dessas limitações, a simplicidade do Jogo da Velha oferece um cenário ideal para explorar estruturas de dados como filas e técnicas de busca sistemáticas, conforme abordado por Alves e Silva (2018). A experiência reforça a importância do entendimento teórico aliado à prática na construção de sistemas inteligentes, como destacado na literatura sobre aprendizado de máquina (Barreto & Souza, 2011).

O trabalho evidencia que técnicas fundamentais de IA são ferramentas poderosas mesmo em cenários aparentemente simples. Futuras extensões do algoritmo poderiam incluir estratégias heurísticas e análise preditiva, conforme discutido em sistemas de aprendizado e busca otimizados. Dessa forma, a abordagem poderia ser adaptada para outros jogos ou problemas mais complexos, ampliando ainda mais seu potencial acadêmico e computacional.

Referências Bibliográficas

1. ALVES, M. L. M.; SILVA, L. M. **Inteligência Artificial: Fundamentos e Aplicações**. Rio de Janeiro: LTC Editora, 2018.
2. BARRETO, G. A.; SOUZA, A. C. **Aprendizado de Máquina: Uma Abordagem Prática com Aplicações em Python**. Rio de Janeiro: Ciência Moderna, 2011.
3. ZIVIANI, N. **Projeto de Algoritmos: Com Implementações em Pascal e C**. São Paulo: Thomson Learning, 2004.
4. REZENDE, S. O. **Sistemas Inteligentes: Fundamentos e Aplicações**. Barueri: Manole, 2011.