

Manual Técnico

Introdução

O projeto Adji-Boto é uma aplicação desenvolvida em Common Lisp que utiliza o algoritmo de Alpha-Beta Pruning para resolver problemas estratégicos do jogo Oware. Este manual detalha a implementação do projeto, incluindo os algoritmos aplicados, as estruturas de dados utilizadas, as limitações identificadas e uma análise crítica do seu desempenho.

Objetivos do Projeto

O principal objetivo do Adji-Boto é fornecer uma solução eficiente para determinar jogadas otimizadas no jogo Oware, utilizando técnicas de inteligência artificial e algoritmos de pesquisa. Especificamente, o projeto pretende:

- Implementar o algoritmo Alpha-Beta Pruning para otimizar a tomada de decisões.
- Minimizar o número de nós analisados durante a pesquisa de estados.
- Oferecer uma ferramenta robusta para avaliar cenários complexos no jogo.

Algoritmo Implementado

O projeto recorre ao algoritmo de Alpha-Beta Pruning, uma melhoria do Minimax, que reduz significativamente o número de estados do jogo que precisam ser analisados. As principais funções desenvolvidas incluem:

1. `alfabeta`

- Explora o espaço de estados do jogo, aplicando cortes alfa e beta para evitar a análise de subárvores irrelevantes.
- Regista informações detalhadas sobre o número de nós analisados, cortes realizados e o tempo de execução.

2. Funções de Log

- `log-terminal-state`, `log-final-state` e `log-prune`: Responsáveis por registrar os eventos durante a execução do algoritmo, garantindo a rastreabilidade e a facilidade no debug.

Estruturas de Dados

O Adji-Boto utiliza listas para representar o tabuleiro e os nós do jogo. As principais estruturas são:

1. Tabuleiro

Representado como uma lista de listas, onde cada sublista contém os elementos de uma linha do tabuleiro:

```
((linha1-elementos) (linha2-elementos))
```

2. Nós do Jogo

Um nó no algoritmo é representado por uma estrutura que inclui o tabuleiro, a profundidade da pesquisa, o nó pai, as peças capturadas e o valor heurístico:

```
(entry depth parent-node captured-pieces heuristic)
```

Limitações do Projeto

Apesar dos bons resultados obtidos, o Adji-Boto apresenta algumas limitações:

- Limitação de Profundidade: Em cenários mais complexos (problemas E e F), o algoritmo sofreu um stack overflow, para uma profundidade de 5.
- Jogador vs CPU: Não foi implementada uma funcionalidade de jogador vs CPU.
- Limite de Tempo: Não foi implementada uma funcionalidade de limitação de tempo de execução do algoritmo.

Análise de Resultados

Os testes realizados com os problemas fornecidos, para uma profundidade de 5 demonstraram os seguintes resultados:

Problema Melhor Valor

A	-3
B	9
C	5
D	36
E	Overflow
F	Overflow
G	90

Estes resultados indicam que o algoritmo funciona corretamente na maioria dos cenários, mas enfrenta dificuldades em problemas complexos.

Conclusão

O projeto Adji-Boto mostra-se uma ferramenta poderosa para o jogo Oware, oferecendo decisões otimizadas através do Alpha-Beta Pruning. Contudo, há espaço para melhorias, especialmente na gestão de profundidade e na implementação de limites de tempo. Com ajustes adicionais, o Adji-Boto poderá atingir uma maior robustez e eficiência computacional.