# **Relatório do Trabalho Prático 2**

# **Resumo**

No âmbito da unidade curricular de Sistemas Inteligentes, foi desenvolvido um trabalho prático cujo objetivo principal consistiu na implementação de algoritmos de Teoria de Jogos aplicados ao **jogo do galo**. Para tal, foram implementados dois algoritmos de decisão: **MinMax** e **Alpha-Beta**.

O sistema permite que um jogador humano defronte o computador, bem como partidas totalmente automatizadas entre dois computadores. Adicionalmente, procedeu-se à análise do custo computacional (tempo de execução e utilização de memória) associado a cada algoritmo, de forma a discutir o seu desempenho.

1. **Descrição do Jogo**

O jogo do galo (tic-tac-toe) é um jogo de estratégia simples, jogado entre dois participantes num tabuleiro de 3x3 casas. O objetivo de cada jogador é conseguir alinhar três símbolos idênticos (X ou O) de forma horizontal, vertical ou diagonal.  
O jogo termina quando um dos jogadores alcança esse objetivo ou quando todas as casas do tabuleiro são preenchidas, resultando num empate.

## **Algoritmos Implementados**

* 1. **MinMax**

O algoritmo **MinMax** é uma técnica clássica de decisão utilizada em jogos de soma zero para dois jogadores. Parte do pressuposto de que ambos os jogadores agem de forma racional e procuram maximizar a sua utilidade.  
O jogador designado como "Max" tenta maximizar o valor da jogada, enquanto o "Min" procura minimizá-lo. A árvore de estados possíveis do jogo é explorada na sua totalidade até atingir estados terminais (vitória, derrota ou empate).

* 1. **Alpha-Beta**

O algoritmo **Alpha-Beta** é uma otimização do MinMax, que visa reduzir o número de estados explorados durante a pesquisa. Para tal, mantém dois valores:

* **Alpha**: representa o melhor valor encontrado até ao momento para o jogador Max.
* **Beta**: representa o melhor valor encontrado até ao momento para o jogador Min.

Sempre que se identifica que um determinado ramo não pode conduzir a uma decisão melhor do que uma já encontrada, esse ramo é descartado ("podado"), evitando assim processamento desnecessário.

1. **Discussão das Características dos Algoritmos**

|  |  |  |
| --- | --- | --- |
| Critério | MinMax | Alpha-Beta |
| Optimalidade | Garante sempre a jogada ótima. | Garante sempre a jogada ótima. |
| Completude | Completo para espaços de busca finitos. | Completo para espaços de busca finitos. |
| Complexidade Temporal | O(b^d), onde b é o fator de ramificação e d a profundidade | O(b^d), mas explora menos nós devido à poda |
| Complexidade Espacial | O(bd) | O(bd) |
| Vantagem Principal | Implementação simples | |  | | --- | |  |  |  | | --- | | Melhor desempenho em tempo e memória | |

1. **Estudo de Custo de Tempo e de Memória**

De modo a comparar o desempenho dos dois algoritmos, foram realizadas medições de:

* Tempo de execução por Jogo (em segundos)
* Memória máxima utilizada (em kilobytes)

Todos os testes foram realizados no modo Computador vs Computador, assegurando condições homogéneas para a avaliação de performance dos algoritmos, sem intervenção humana.

As medições foram efetuadas durante partidas completas entre dois computadores, recorrendo a várias instâncias de jogo.

|  |  |  |
| --- | --- | --- |
| Algoritmo | Tempo Médio por Jogo (s) | Memória Média por Jogo (KB) |
| MinMax | **43** | **15** |
| Alpha-Beta | **2** | **11** |

(Os valores apresentados são indicativos, arredondados e poderão variar ligeiramente entre execuções.)

1. **Discussão dos Resultados**

A análise dos dados recolhidos permitiu verificar que:

* Ambos os algoritmos garantem decisões ótimas, conduzindo à vitória ou ao empate, sem jogadas subótimas.
* O algoritmo MinMax, apesar de assegurar a escolha ótima, revelou ser extremamente ineficiente em termos de tempo, demorando em média **43 segundos** por jogo.
* O algoritmo Alpha-Beta, ao introduzir poda na árvore de pesquisa, foi substancialmente mais rápido, concluindo jogos em apenas **2 segundos** em média.
* Em termos de memória, embora a diferença seja menos expressiva, o Alpha-Beta também demonstrou ser mais eficiente, consumindo em média **11 KB** face aos **15 KB** registados com o MinMax.
* Esta grande diferença temporal evidencia a importância crítica da otimização Alpha-Beta, mesmo em jogos relativamente pequenos como o jogo do galo.
* A diferença de desempenho entre os dois algoritmos torna-se particularmente evidente nas fases iniciais do jogo, em que existem mais possibilidades de jogadas.

O recurso ao modo Computador vs. Computador para a realização dos testes permitiu assegurar a homogeneidade das condições de medição, eliminando variáveis externas que poderiam enviesar os resultados.

1. **Conclusões**

A implementação e análise comparativa dos algoritmos MinMax e Alpha-Beta permitiram obter as seguintes conclusões:

* Ambos os algoritmos garantem a execução de jogadas ótimas no jogo do galo.
* O algoritmo **Alpha-Beta** apresentou um desempenho drasticamente superior, reduzindo o tempo de execução de **43 segundos** para apenas **2 segundos** por jogo, mantendo a qualidade da decisão.
* Em termos de consumo de memória, o **Alpha-Beta** também se revelou mais eficiente, ainda que a diferença não seja tão acentuada quanto no tempo de execução.
* A utilização de poda é essencial para tornar viáveis algoritmos de decisão em jogos, mesmo em espaços de busca de dimensão moderada.

Assim, conclui-se que o algoritmo Alpha-Beta deve ser preferencialmente utilizado em jogos de decisão, tanto pela sua eficiência computacional como pela manutenção da optimalidade das soluções.

Este trabalho permitiu consolidar conhecimentos sobre a aplicação de algoritmos de decisão em jogos e a importância de técnicas de otimização para problemas de maior complexidade.