

## INTRODUÇÃO

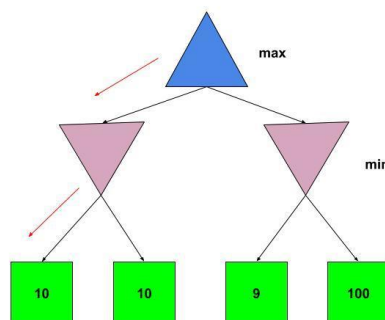
A publicação escolhida como alvo de pesquisa foi um artigo da organização Geeks For Geeks denominada “Expectimax Algorithm in Game Theory” cuja tradução seria algo como “Algoritmo Expectimax na Teoria de Jogos”. A publicação fornece um panorama geral sobre esse tipo de algoritmo, explicando como funciona o algoritmo de modo contrastante ao Min-Max, estabelece comparações entre os dois (vantagens e desvantagens), implementação em diferentes linguagens, análise de complexidade e uma breve apresentação de aplicações.

A proposta do autor com essa publicação é explicar Expect Min-Max por meio de um conhecimento prévio do Min-Max básico, as explicações são realizadas de maneira bem objetiva, certamente já destinada para um público experiente com programação e algoritmos de busca.

## ATIVIDADES DESENVOLVIDAS

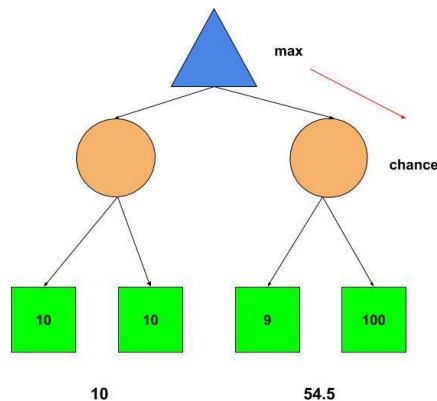
No primeiro tópico "Expectimax vs Minimax", o autor procura explicar o Expect Min-Max por meio das diferenças entre um e outro, as Figuras 1 e 2 ilustram seu pensamento, explicando que no Min-Max, a escolha pela esquerda é óbvia, enquanto no Expect Min-Max a chance de escolher a direita é mais provável.

Figura 1 - Árvore do Min-Max



A diferença substancial de um para outro é a consideração das ramificações que um nó têm, que a partir da média dos nós filhos, calcula-se a chance de expandir nesse nó ou não.

Figura 2 - Árvore do Expect Min-Max



O autor depois organiza em tópicos, vantagens e desvantagens do Expect Min-Max sobre o Min-Max básico, que em um balanço geral, o Expect Min-Max é ótimo para lidar com oponentes cuja ação é aleatória e é capaz de evitar ótimos locais. No entanto, ele apresenta bastantes desvantagens, o autor lista três deles: Não há garantia de solução ótima, toda árvore deve ser explorada, ou seja, o algoritmo pode ser lento; e cada variação de valor de árvore pode alterar a expansão dela.

O autor explica a lógica de seu algoritmo, tal como fornece implementação em diferentes linguagens, seu algoritmo é exatamente o exemplo dado acima, há apenas a maximização e a escolha por chance, o algoritmo dele retorna o valor da chance escolhida (maior).

A Figura 3 mostra a implementação, a condição de parada é chegar a um nó folha. Se a variável indicar que é momento de maximização, ele pega o de maior valor entre os nós filhos, se não for problema de maximização, adota o nó de chance como média entre os valores de filho e retorna como valor final.

Uma observação interessante é que o autor implementa esse algoritmo para uma árvore binária de nível de profundidade 2.

Figura 3 - Implementação Expectimax

```
// Getting expectimax
static float expectimax(Node node, boolean is_max)
{
    // Condition for Terminal node
    if (node.left == null
        && node.right == null) {
        return node.value;
    }

    // Maximizer node. Chooses the max from the
    // left and right sub-trees
    if (is_max) {
        return Math.max(
            expectimax(
                node.left, false),
            expectimax(node.right, false));
    }

    // Chance node. Returns the average of
    // the left and right sub-trees
    else {
        return (float) ((
            expectimax(node.left, true)
            + expectimax(node.right, true))
            / 2.0);
    }
}
```

O autor conclui com a função de complexidade de tempo e espaço do algoritmo, correspondendo ao mesmo que uma busca em profundidade:

Complexidade de Tempo:  $O(b^m)$

Complexidade de Espaço:  $O(b * m)$

Sendo  $b$  o número máximo de nós filhos e  $m$  o maior caminho a ser percorrido pela árvore.

A última observação dada pelo autor são as aplicações em jogos como Pac-Man e Campo Minado, em que o jogador age como agente MAX e minas e fantasmas agem como nós de chance. No caso do Pac-Man, a aplicação de estados terminais (vitória, derrota e empate), assim como função de avaliação (possíveis estados certa profundidade) são de interessante implementação para funcionamento.

## TABELA DE FORMULAÇÃO DO PROBLEMA

Apesar da simples implementação dessa publicação, ainda é possível ressaltar elementos presentes importantes quando se trata de busca competitiva, correspondente aos dados que preenchem a tabela 1.

Tabela 1 - Formulação do Problema

Aspecto do Minimax	Descrição	Implementação
Estados	Nó da árvore	Classe Node com valor e referência a dois filhos
Ações	Escolhe valor máximo ou valor de média entre seus nós filhos	Condicionais "If-Else" Presentes na Figura 3
Função de Avaliação	Não implementada diretamente, há somente uma avaliação que é no critério de parada (o nó ser folha)	Primeiro condicional "if" na Figura 3.
Corte Alfa-Beta	Expectimax age de maneira probabilística, não sendo certo podar certos valores por interferir na média dos nós de chance	Não implementado

## CONCLUSÃO

A publicação é útil para compreensão da concepção do algoritmo Expectimax, no entanto, não apresenta implementação para minimização do problema, não explica exatamente a ideia dos nós de chance e nem explicam diferentes cálculos que podem ser realizados sobre o nó de chance. É uma publicação que informa, porém creio que poderia ser mais bem formulada e completa, a publicação é um resumo muito sucinto que chega a remover assuntos importantes sobre o algoritmo.

## BIBLIOGRAFIA

GEEKS FOR GEEKS. Expectimax Algorithm in Game Theory. Geeks For Geeks, 2021. Disponível em: <<https://www.geeksforgeeks.org/expectimax-algorithm-in-game-theory/>>. Acesso em: 31 Maio 2024.