

# ExpectiMax

## Introduction

EXPECTIMAX est un algorithme de parcours d'arbre ou de graphe de jeu qu'on peut utiliser pour implémenter une Intelligence Artificielle résolvant les jeux discrets séquentiels non déterministes tels que Candy Crush et 2048.

Dans son utilisation principale, l'adversaire n'est pas supposé être un agent intelligent mais un phénomène aléatoire. Ainsi, au lieu des nœuds MIN, on a des nœuds EXPECT. En effet, EXPECT fait référence à l'*Expectation* ou l'*Espérance mathématique*.

On rappelle que l'espérance mathématique dans une expérience aléatoire est la valeur **moyenne** attendue de la variable aléatoire. Pour la calculer, on multiplie chaque valeur possible de la variable aléatoire par sa probabilité respective, puis on somme ces produits. Cela signifie que l'espérance mathématique est une **moyenne pondérée** des valeurs possibles de la variable aléatoire, où les poids sont les probabilités de chaque valeur.

## Algorithme

```
function expectimax(node, depth, isMaxPlayer):
    if node is a terminal node or depth = 0:
        return eval(node)
    if isMaxPlayer:
        bestValue = -infinity
        for child in node.children:
            value = expectimax(child, depth-1, False)
            bestValue = max(bestValue, value)
        return bestValue
    else:
        expectedValue = 0
        for child in node.children:
            probability = getProbability(child)
            value = expectimax(child, depth-1, True)
            expectedValue += probability * value
        return expectedValue
```

La fonction *getProbability* retourne la probabilité d'apparition du nœud en question selon les règles du jeu ou une étude faite sur le jeu.

## Remarques et conclusion

### Avantages :

1. **Gestion de l'incertitude** : Expectimax peut gérer des jeux ou des situations comportant des éléments de hasard, contrairement à l'algorithme Minimax, conçu pour les jeux déterministes.
2. **Adaptable à différents scénarios** : Expectimax peut être appliqué à diverses situations impliquant le hasard, comme la prise de décision en situation d'incertitude, le jeu ou même la gestion des risques.

### Inconvénients :

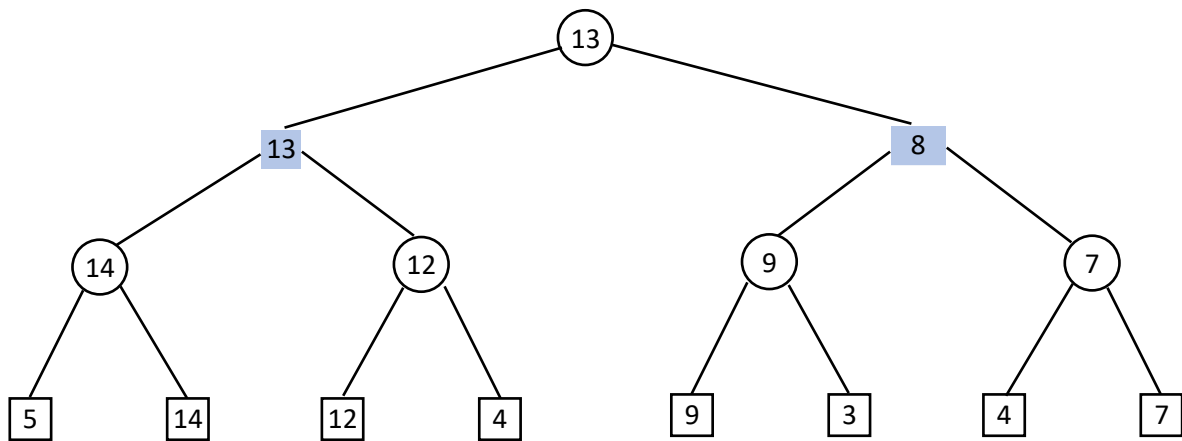
1. **Complexité computationnelle** : Expectimax a une complexité computationnelle élevée en raison de la nécessité de calculer les valeurs attendues à chaque nœud. Cela peut rendre l'algorithme lent et peu pratique pour les grands espaces de recherche ou la prise de décision en temps réel. De plus, on ne peut pas appliquer les coupes alpha et beta dans l'algorithme de parcours Expectimax.
2. **Performances dépendantes de probabilités précises** : La qualité des décisions prises par Expectimax dépend de la précision des estimations de probabilité pour chaque nœud. Des probabilités inexactes peuvent entraîner des décisions sous-optimales.

## Méthodes connexes à étudier

Il existe également d'autres méthodes telles que l'algorithme ExpectiMiniMax ou les méthodes de Monte Carlo.

## Exemples

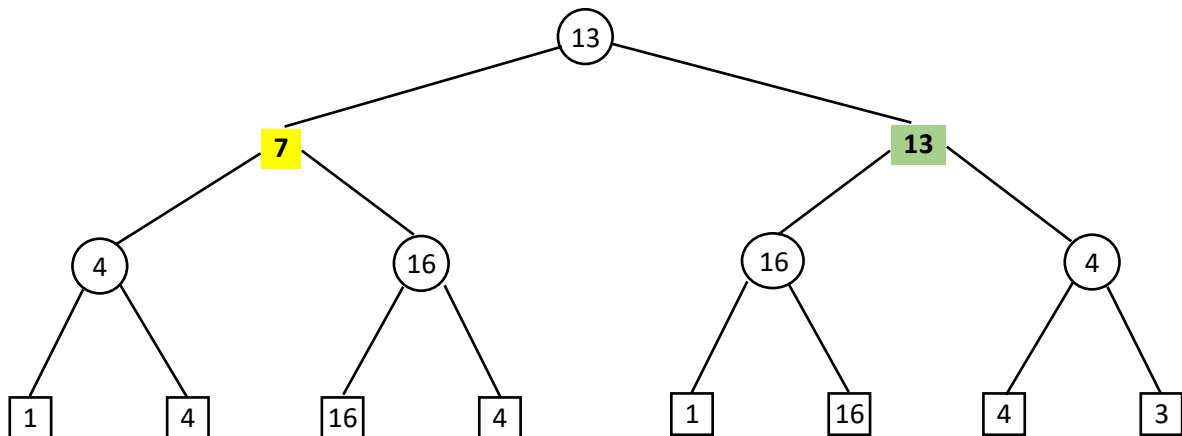
Exemple 1- Si les fils des nœuds de chances sont **équiprobables**



○ : Nœud **Max**  
 □ : Nœud **Expect**

Exemple 2 – Si le fils gauche d'un nœud de chance a 3 fois plus de chances d'apparaître que son fils droit

Donc :  $p_{gauche} = \frac{3}{4}$  et  $p_{droite} = \frac{1}{4}$



○ : Nœud **Max**  
 □ : Nœud **Expect**

$$\frac{3}{4} \times 4 + \frac{1}{4} \times 16 = 3 + 4 = 7$$

$$\frac{3}{4} \times 16 + \frac{1}{4} \times 4 = 12 + 1 = 13$$