

Heurística-Inteligencia Artificial

Heuristic -Artificial Intelligence

Autor: **Jarol Vidal Angulo**

IS&C, Universidad Tecnológica de Pereira, Pereira, Colombia

Correo-e: **Jarol.Vidal@utp.edu.co**

Resumen— Este documento presenta un resumen acerca de la heurística aplicada a la inteligencia artificial en este caso aplicándolo a el juego clásico de ajedrez donde se analizara las posibles jugadas para ganar la mayor ventaja al jugar una partida clásica de ajedrez utilizando la heurística, basándonos en la experiencia del jugador para proporcionar la correcta solución

Palabras clave— sistemas, inteligencia artificial, software, computación, heurística, grafo

Abstract—This document presents a summary of what is the heuristics applied to artificial intelligence in this case applying it to the classic game of chess where we will analyze the possible moves to win the greatest advantage when playing a classic game of chess using heuristics, based on the experience of the player to provide the correct solution.

Key Word— systems, artificial intelligence, software, computing, heuristics, graph



Figura 1. Tablero de ajedrez

I. INTRODUCCIÓN

En la rama de la inteligencia artificial existe un concepto llamado búsqueda heurística que nos permite resolver algún tipo de problema con conocimientos previos para este caso se planteara un problema acerca de la ventaja que se puede tener en ajedrez para favorecer a un jugador.

1.1 PLANTEAMIENTO DEL PROBLEMA

El ajedrez es un juego de estrategia en el que dos personas se desafían en un tablero cuadrado de 64 casillas y dos grupos de figuras 16 para cada jugador hay fichas blancas y negras estas se compone de: rey, dama o reina, dos alfiles, dos caballos, dos torres y ocho peones cada una de estas piezas tiene un movimiento determinado por ejemplo: el caballo se mueve en L en un recorrido de 4 celdas, el objetivo del juego es derrocar al rey del contrincante y para esto cada jugador debe intentar acercarse a la casilla ocupada por el rey y presionar hasta que quede desprotegida para hacer Jaque Mate y ganar la partida, a continuación se muestra un tablero de ajedrez

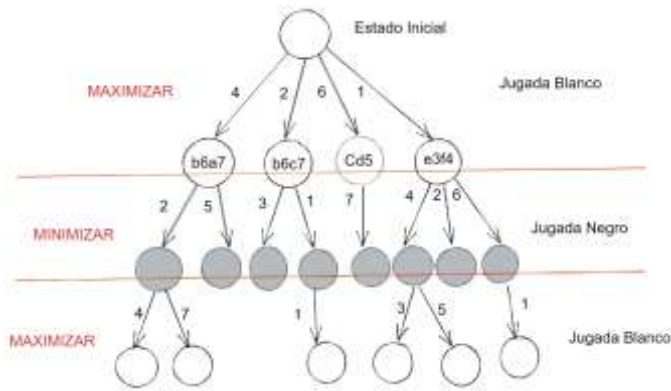
Para este caso se supone el hecho de que estamos jugando con las fichas blancas, como se puede intuir algo muy importante en este juego es la estrategia que se necesita para “predecir” el movimiento del contrincante en este caso las fichas negras por esto lo que se hace al momento de jugar es analizar que movimientos puede realizar el oponente con respecto al movimiento que se hace con las fichas blancas.

Por ejemplo, si solo se encuentra la reina protegiendo al rey de un posible ataque lo más lógico seria no moverla y mover otra ficha con la que podamos comer una ficha del contrincante, dificultar el Jaque Mate o poner en Jaque Mate a su rey.

El problema que se quiere plantear es la toma de decisiones acerca de una partida de ajedrez tomando los movimientos actuales y futuros que se puedan dar en una partida de ajedrez se debe ilustrar y ejemplificar las mejores jugadas que se pueden realizar anticipando los movimientos del contrincante asegurando siempre llevar la ventaja de los movimientos hasta hacer el Jaque Mate y ganar la partida del juego.

1.2 POSIBLE SOLUCION

Se plantea una posible solución donde se modela los movimientos de los dos jugadores haciendo uso de un árbol que es un tipo abstractos de datos donde se usa una estructura jerárquica donde hay una raíz, ramas, subárboles y hojas, además para este caso se hará uso de los niveles del árbol para así reconocer que jugador esta haciendo el movimiento cada nodo del árbol representa el movimiento que se puede hacer y también la arista o conexión entre nodos que tiene un peso que es un valor numérico que hace referencia a las unidades de beneficio del movimiento para determinado jugador para este caso se quiere que gane el jugador blanco, ejemplo:



Para este caso se procedió a realizar la codificación Python que es un lenguaje de programación interpretado multiparadigma ya que soporta la programación orientada a objetos de lo cual vamos a hacer uso para este caso donde haremos uso de un objeto llamado “nodo” para así generar el árbol de posibles movimientos.

Objeto:

Nodo
-String: id_nodo
-vector: pesos
-vector: hijos
-boolean: visita
-Getters
-Setters

Ahora aplicamos la búsqueda heurística para hallar una posible solución, en este caso se aplica este método ya que este es un problema que no cuenta con una solución exacta debido a algunas ambigüedades en el problema y además porque el costo computacional para este problema es muy alto, por lo tanto, para la codificación se definieron algunas reglas básicas para encontrar una solución las cuales son:

el jugador que se quiere que gane el juego es el primero en mover para este caso, por lo tanto, el primer nivel se maximiza el siguiente se minimiza el siguiente después de este se maximiza y así sucesivamente entendiendo que un nivel corresponde a los pesos entre un nodo a otro.

la maximización de un nivel se requiere revisar un nivel por debajo ya que ninguno de los pesos de los hijos de este puede superar la maximización.

Se lleva un recuento de los posibles movimientos con un valor de beneficio asignado posteriormente se analiza cual es el que tiene mayor beneficio para posiblemente ganar la partida de ajedrez con esto concluimos una solución con los movimientos que se deben hacer.

REFERENCIAS

Referencias en la Web:

[1]<https://www.jubilaciondefuturo.es/es/blog/el-ajedrez-un-deporte-que-estimula-la-mente.html>

[2]http://kali.azc.uam.mx/clc/03_docencia/posgrado/i_artificial/8_FuncionesHeuristicas.pdf

Diagramación en la Web:

[1]<https://excalidraw.com/>