TEC | Tecnológico de Costa Rica

Instituto Tecnológico de Costa Rica

Inteligencia Artificial – IC6200

Chines Checkers

Prof. María Auxiliadora Mora

Integrantes:

Emanuel Jiménez Sancho - 2017136727

Rodrigo Venegas Vega - 2017047627

II Semestre 2020

Contenido

Descripción del proyecto

ChineseCheckers es un proyecto que pone a disposición de un usuario el juego de damas chinas. El juego está pensado para un solo usuario que va a jugar contra una inteligencia artificial desarrollada por el grupo de estudiantes creadores del software. Mediante algunos algoritmos especificados por la profesora, la inteligencia artificial deberá suponer un verdadero desafío para el usuario que desee retar este software.

Agente

El agente, nombrado por el equipo como chineseIAchecker, tiene como objetivo principal jugar y ganar la mayor cantidad de partidas posibles contra diferentes usuarios. Al mismo tiempo, y si es posible, el agente debe adelantarse al menos tres jugadas para tomar una decisión. En pocas palabras, la IA es capaz de pensar a una profundidad de tres niveles.

Si ya empezó el juego y el agente no ha perdido ni ganado, busca la mejor jugada posible para llegar a la condición de victoria. Se muestra un cuadro de percepciones y acciones más adelante, donde (X,Y) refiere a una ficha en una posición cualquiera.

Percepciones	**Acciones**
[(X,Y), No-move-yet]	Incrementar la importancia de (X,Y)
[(X,Y), greater-weight]	Mover (X,Y) a su mejor posición

[(X,Y), can-jump]	Mover (X,Y) por encima de sus fichas vecinas
[(X,Y), can-arrive]	Mover pieza a su última posición en el tablero
[(X,Y), can-block]	Bloquear ficha enemiga

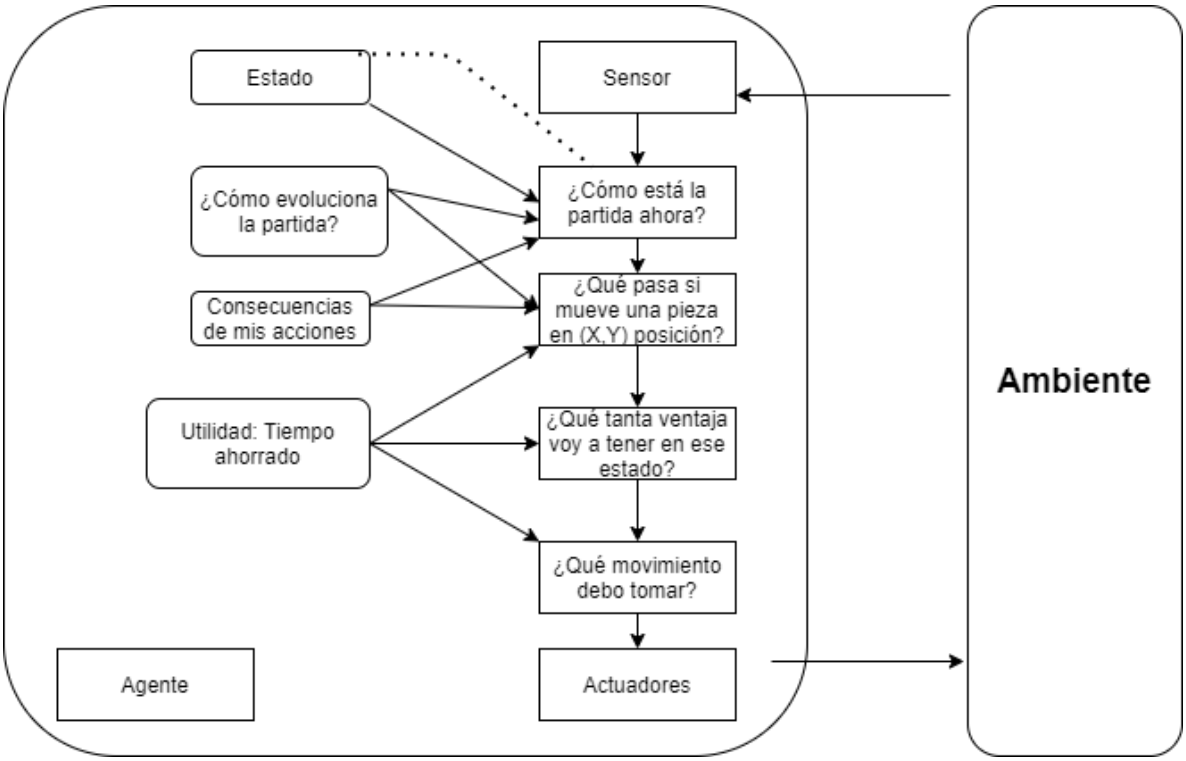
El entorno de la tarea (PEAS): la medida de desempeño, el entorno, los actuadores y los sensores:

Medida de desempeño: 80. El agente cumple con su trabajo pero podría ser mejor a costa de un poco más de tiempo de respuesta.

Entorno: Tablero 10x10, casillas vacías, casillas ocupadas, fichas aliadas, fichas enemigas, filas, columnas, diagonales, “rosa” de jugadas. Entorno parcialmente observable.

Actuadores: Posibilidad de salto, posibilidad de bloqueo, peso acumulado, posibilidad de incrementar el peso, jugadas y pesos a futuro.

Sensores: El único elemento de hardware que forma parte de este agente es el procesador.



Eval

Eval trabaja con diferentes componentes para valorar el peso y la importancia de una jugada particular. Primeramente, se realiza un recorrido de todas las fichas aliadas y sus mejores jugadas, para hallar estas mejores jugadas se utiliza una matriz de pesos que funciona con un “Frío-Caliente” donde “frío” son las zonas alejadas del centro y “caliente” son las zonas con mejor posición en el tablero y más cercanas a la condición de victoria. Además, se incrementa el peso de aquellas fichas con alguna oportunidad de saltar sobre otras, esta es considerada una jugada importante.

También toma en cuanta, en todo momento, las jugadas del oponente. De esta forma se logra tanto aprovechar sus errores como bloquear y dificultar su juego.

Rendimiento del sistema

Para optimizar recursos en cuanto a memoria, se aprovecha la recursividad para hacer recorridos y hacer cálculos sin necesidad de crear y almacenar información en una estructura de datos. Sin embargo, se salta este paso y es posible prescindir del árbol si se trabaja únicamente con “backtracking” con valores que no es necesario recordar o se puede obviar directamente.

Además, el agente podría realizar siempre los mismos movimientos al inicio de una partida. Es decir, cuando no se han jugado demasiados movimientos, existen “jugadas triviales” que siempre van a ser correctas dado que el impacto de las mismas será mínimo a lo largo de la partida. Cabe recalcar que el número de jugadas triviales no debe ser muy grande, pero podría agilizar en gran medida el rendimiento del sistema.

Para agilizar la recursividad y no malgastar recursos, se hace una “poda” que descarta aquellas jugadas cullo peso nunca va a alcanzar a ser competente. Es decir, se ignoran aquellas jugadas que, por su poca utilidad, no alcanzarían a ser útiles para llegar a la victoria.

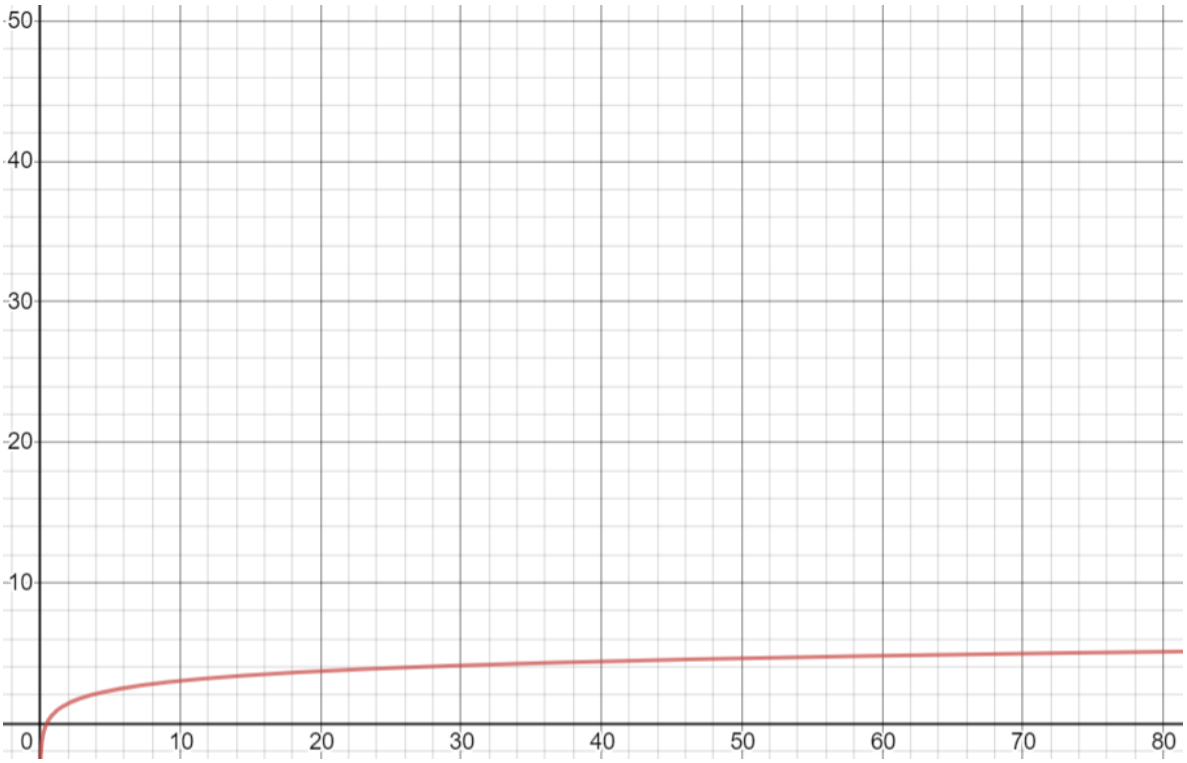
Tabla y Gráfico de rendimiento

Prueba	Eficacia	Tiempo (ms)
Jugada 1	0	1.9
Jugada 2	0	1.8
Jugada 3	1	1.9
Jugada 4	2	1.9

Jugada 5	2	3.4
Jugada 6	2	5.1
Jugada 7	2	3.4
Jugada 8	2	4.7
Jugada 9	2	3.4
Jugada 10	2	4.7
Jugada 11	2	5.6
Jugada 12	3	3.2
Jugada 13	3	3.4
Jugada 14	3	3.7
Jugada 15	3	12.8
Jugada 16	3	14.6
Jugada 17	4	15.2
Jugada 18	4	13.8
Jugada 19	4	14.6
Jugada 20	4	18.1
Jugada 21	4	15.2
Jugada 22	4	13.1
Jugada 23	4	10.3
Jugada 24	4	11.2
Jugada 25	2	4.2
Jugada 26	1	3.6
Jugada 27	0	1.8

En base a las pruebas realizadas, se tomó una muestra (27 pruebas no sesgadas) con las cuales se pudo hallar la relación de incremento de la eficacia de una predicción o jugada del agente en contraste con el tiempo que tomó hallar dicha respuesta. Se observa que la eficacia de jugada incrementa en relación con el tiempo que tomó encontrarla, esto se debe a que las piezas se encuentran cada vez más dispersas y el algoritmo tiene más jugadas por analizar.

Sin embargo, aunque es difícil de ver en la gráfica, llega un punto en el que no vale la pena esperar mucho tiempo para una jugada que, a pesar de ser óptima, puede ser preferible esperar menos para alcanzar un resultado parecido en cuanto a eficacia.



Bibliografía

Racket Documentation. (2020). Retrieved 26 October 2020, from https://docs.racket-lang.org/