

# Proyecto 1, Inteligencia Artificial

Miguel Ángel Askar Rodríguez - 201355842

Danny Fernando Cruz Arango - 2014549949

miguel.askar@correounivalle.edu.co

danny.cruz@correounivalle.edu.co

## Resumen

El sudoku como juego, representa una dificultad bastante alta para los humanos; por lo tanto, es un excelente proyecto para conocer los alcances de la inteligencia artificial, específicamente, de los agentes inteligentes. A través de este documento se observa cómo se diseña una estrategia para resolverlo, así como la dificultad con la que es capaz de trabajar y la definición en sí del nivel de la misma.

## Abstract

The Sudoku as a game, represents a though challenge for humans; by the way, is an excellent project to know artificial intelligence approaches. Through this paper, it shows how to design a strategy to figure the Sudoku out, as well the difficult which the agent can work and itself definition.

## Palabras clave

Sudoku, Inteligencia, artificial, preferente, amplitud, búsqueda, no informada, agente.

## 1. Selección de estrategia

Se seleccionó búsqueda no informada, preferente por profundidad. Debido a que se creará un árbol con los diferentes estados y se seleccionará, para la búsqueda, alterar la primera casilla (modificable y con las opciones de llenado más bajas) que se encuentre en el tablero, es decir, la que esté a la izquierda en el árbol.

## 2. Estado

El estado consiste en el tablero con las casillas con menor probabilidad almacenadas y con las probabilidades para la primera casilla (más arriba y más a la izquierda en el tablero). Esto con el fin de no tener que almacenar todas las probabilidades y sólo guardar los nodos hermanos del que se usará para continuar la búsqueda.

El estado contiene los siguientes ítems:

- Tablero Actual.

- Casillas con menor probabilidad.

- Opciones de la primera casilla con menor probabilidad.

Cuando se actualiza el estado, este va modificado con la casilla alterada con la probabilidad correspondiente y se vuelven a procesar las casillas con menor probabilidad, así como las opciones para la primera casilla.

A medida que se van generando estados, estos se almacenan en un vector *camino*, con el fin de disponer de ellos en caso de tener que devolverse en el árbol.

## 3. Estrategia de detección de errores y solución

Un error se presenta cuando la cantidad de casillas vacías es diferente a la cantidad de casillas con posibilidades. Este análisis se hace en cada iteración para predecir cuando el tablero está llegando a un estado erróneo y así no tener que generar el árbol hasta sus hojas.

Cuando el error es detectado, el agente empieza a devolverse a través del vector *camino*, probando así, cada estado anterior con diferentes posibilidades, o cambiando de casilla (utilizando sus hermanos).

## 4. Definición de dificultad del sudoku

El agente implementado para la búsqueda de la solución para sudokus, es capaz de resolver los que tengan dificultad normal o inferior. El agente no resuelve cualquier sudoku, puesto que se busca que este sea rápido. Sin embargo, en un documento posterior, se evidencia una solución cuya implementación es compleja, pero garantiza la solución de sudokus de alta dificultad en tiempo mínimo. Por otro lado, si la estrategia no fuese compleja, la rapidez se lograría utilizando maniobras de combinatoria como: calcular las casillas con menos probabilidades, asignar estos valores a dichas casillas y probar hasta que una combinación de estos valores probables sea correcta (no genere error en el tablero). Pero el objetivo de este agente es no usar este tipo de estrategias.

### 4.1 Dificultades:

**4.1.1 Dificultad-Baja:** Desde el inicio se presentan casillas con una sola probabilidad y durante el recorrido del árbol, se generan más con una sola opción, por lo tanto, el camino inicial siempre será el correcto.

**4.1.2 Dificultad-Normal:** Durante el recorrido del árbol, se presenta 2 como probabilidad menor en las casillas

del tablero, por lo tanto, este nodo tendrá más de un hijo directo y tendrá que ser considerado un cambio entre nodos hermanos en caso de haber errores posteriores en el tablero.

**4.1.3 Dificultad-difícil:** Durante el recorrido del árbol, se presenta 3 como probabilidad menor en las casillas del tablero, por lo tanto, este nodo tendrá más de un hijo directo y tendrá que ser considerado un cambio entre nodos hermanos en caso de haber errores posteriores en el tablero y, incluso después de haber considerado el siguiente hermano (el segundo), este camino podría ser incorrecto y tendría que devolverse hasta el nodo donde se presentaron las 3 probabilidades.

## 5. Árbol generado por el agente

El árbol generado por este agente es el resultado de una búsqueda de soluciones por profundidad, pero para ahorrar memoria cuando se trabaja con este, sólo se almacenan los nodos del camino a utilizar y los nodos hermanos.

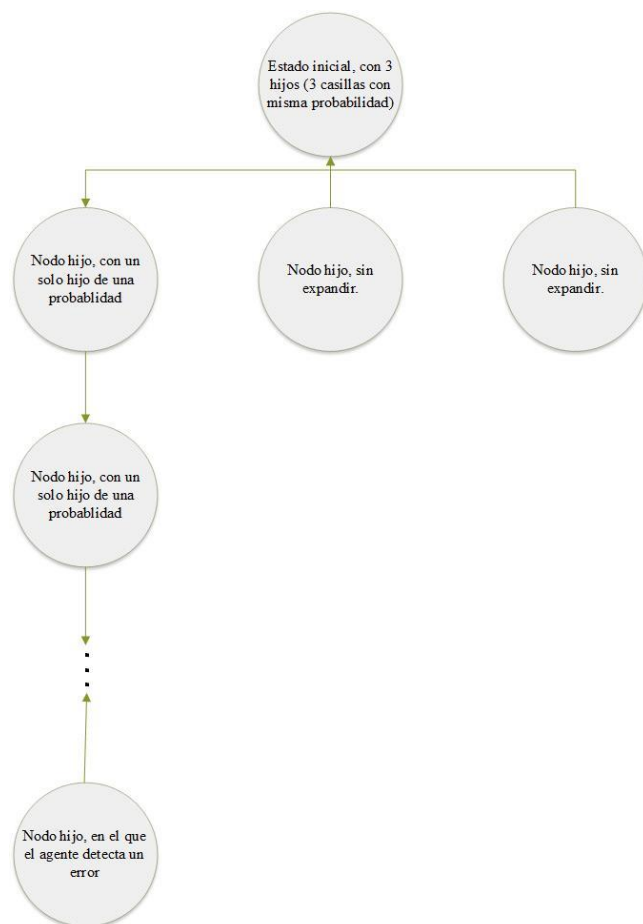


Imagen 1. Árbol generado

Cómo se puede apreciar en la imagen anterior, el primer hijo se expandió hasta que encontró un error en la solución del tablero. Por lo tanto, se devolverá hasta el estado inicial puesto que tiene 3 hijos (para el caso de este árbol en específico), borrará el camino recorrido e iniciará uno nuevo a través de su hermano (el segundo hijo); evidenciando así, cómo el agente ahora memoria durante su búsqueda.

## 6. Conclusiones

Se pueden usar múltiples estrategias para abordar este tipo de problemas, obteniendo un mismo resultado con diferentes tiempos de ejecución y diferentes complejidades (del algoritmo y de diseño).

El sudoku es un juego en el cuál varía la complejidad, por lo tanto, pueden resultar diferentes estrategias para su solución. Dejando a un lado la combinatoria, el uso de diferentes estrategias para un mismo sudoku, puede resultar interesante en la medida que reduce o aumenta la complejidad de la solución.

El uso de árboles resulta pesado en la computación, pero con una buena estrategia que sólo almacene los nodos a utilizar (sin olvidar que existen hermanos y otras posibilidades), se puede lograr una gestión eficaz de la memoria para la implementación del proyecto.

## 7. Bibliografía

*Agentes inteligentes*, Introducción a la inteligencia artificial.

*Búsqueda por profundidad*, Introducción a la inteligencia artificial.

*Definición de sudoku, consejos y estrategias*, <http://www.sudoku.com>

*Ejemplos de sudokus, selección del ejemplo*, <http://www.sudoku-online.org>