Busquedad en profundidad y anchura

Andres Felipe Galvis Cubillos - 1088035102

Mateo Fuentes Aguirre - 1088335441

[[1]](#footnote-1)

*Abstract*— **This text gives the most detailed explanation possible about the use of search algorithms, from how they work to their main algorithms, and because thanks to these, a new panorama has been opened to artificial intelligence, the document places great emphasis to the width or width search algorithms and depth search.**

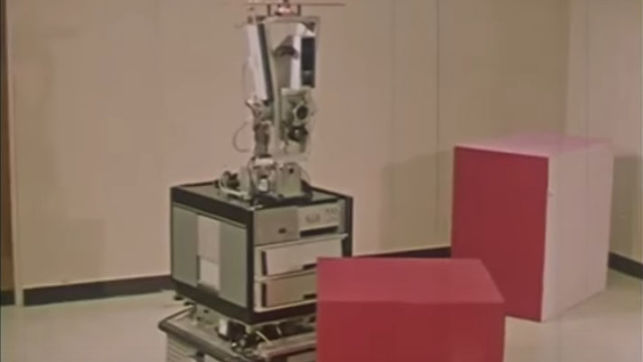
*Index Terms*— search, algorithms, artificial intelligence, autonomy, depth

***Resumen*—** **por medio de este texto se da la explicación mas detallada posible acerca del uso de los algoritmos de búsqueda desde como funcionan hasta sus principales y como gracias a estos se ha abierto un nuevo panorama a la inteligencia artificial, en el documento se hace un gran énfasis al los algoritmos de busquedad por anchura o amplitud y busquedad de profundidad.**

***Palabras claves—* búsqueda, algoritmos, inteligencia artificial,autonomia,profundidad**

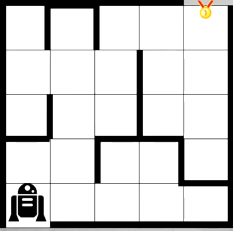
# INTRODUCTION

La búsqueda de información para su propia utilización por medio de la inteligencia artificial ha dado solución a la resolución de problemas de pathfinding y planificación como pueden ser los problemas de navegación (Google maps)

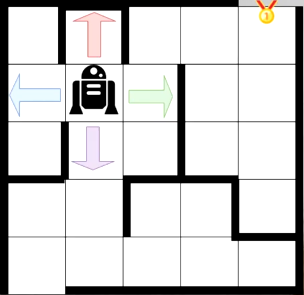


Por medio de inteligencia artificial tuvieron sus inicios con el robot shakey fue el primer robot móvil de propósito general capaz de razonar sobre sus propias acciones. Mientras que otros robots tendrían que ser instruidos en cada paso individual de completar una tarea más grande, Shakey podría analizar los comandos y dividirlos en trozos básicos por sí mismo. esto se logró gracias a la aplicación del algoritmo A\* propuesto por el científico estadounidense Nils Nilsson

# ¿DE QUE MANERA EL ROBOT LOGRARA LA META?



En este momento es que se utilizaran los algoritmos de búsqueda, Cuando el robot tiene cierta información previa del medio en el cual se está moviendo podríamos estar hablando que este problema puede ser abordado desde un manejo de búsqueda informada pero si por el contrario este carece de conocimiento previo lo debería realizar por medio de la búsqueda no informada tenemos el caso de que nuestro robot lo único que sabe es los lados hacia donde se puede mover y parar cuando este llegue a la meta para esto el robot debe tener una clase de representaciones o estados que le indiquen como se encuentra en ese momento, esta clase de estados puede estar representado por medio de un grafo haciendo así que cada nodo del grafo se represente un espacio en blanco de un laberinto por ejemplo si nos basamos en el estado inicial del robot que se representa en la imagen anterior podemos visualizar que este solo se puede mover hacia la derecha o la izquierda, pero en la siguiente imagen podemos ver representados que el robot ahora puede moverse hacia varios lados



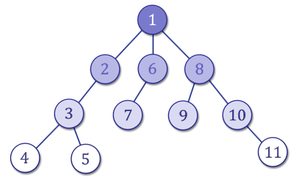
# ¿QUE PODRÍAMOS APLICAR PARA DARLE LAS INSTRUCCIONES DE ESTADO AL ROBOT?

Gracias a múltiples algoritmos que tenemos a nuestra disposición podemos darle al robot su compleja autonomía en este documento nos vamos a basar tanto en la busquedad de profundidad como también en la busquedad por anchura

# BUSQUEDAD DE PROFUNDIDAD

Este es un algoritmo que permite recorrer todos los nodos de un grafo o de un árbol de manera ordenada, pero no uniforme. Su funcionamiento consiste en ir expandiendo todos y cada uno de los nodos que va localizando de forma recurrente en un camino concreto. Cuando ya no queden más nodos que visitar en dicho camino, regresa (Backtracking) de modo que repite el mismo proceso con cada uno de los nodos ya procesados

este genera el recorrido del árbol en PRE-ORDEN (desde izquierda a derecha). Acá un pequeño ejemplo para ayudar a entender de mejor manera



Esto quiere decir que el recorrido en PRE-ORDEN será : 1, 2, 3, 4,  5, 6, 7, 8, 9, 10 ,11.



En el caso de nuestro ejemplo del robot este tendría que comenzar por el lado derecho recorriendo por completo ese lado del laberinto al captar la información de que ese no es el camino tomara la dirección de arriba teniendo así la opción más eficaz y el camino más corto de todos para poder llegar a la meta

Uno de los principales problemas que se puede denotar con nuestro ejemplo es que probablemente este método no pueda ser muy optimo ya que como pudimos ver este si nos da el camino más corto pero primero este tuvo que tomar un camino diferente para poder descartarlo en casos donde hallan mayores opciones en el estado inicial y caminos más largos harían que este proceso se haga un poco tedioso y muy poco posible que tenga un tiempo estimado menor al esperado para estos casos hay mejores algoritmos como uno de ellos que es el de búsqueda por anchura

Pseudocodigo:

funcion buscar\_en\_hijos(Nodo:n)

variable encontrado=boolean

inicio

    si solucion(n->hijo)

    retornar n->hijo

    sino

    n1=n->hijo

    encontrado=falso

        mientras no (encontrado)

        n1=n1->hermano

        sisolucion(n1)

        retornar n1

        sino

            n1=null

            romper ciclo

            buscar\_en\_hijos(n->hijo)

            n2->n->hijo

                mientras(n2->hermano!=null)

                n2=n2->hermano

                buscar\_en\_hijos(n2)

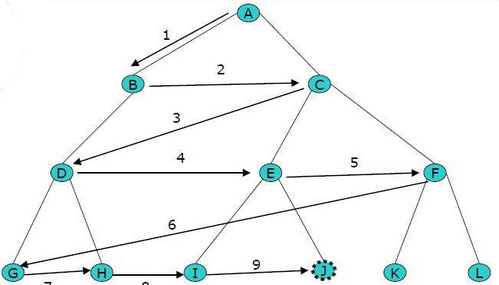
        fin si

    fin mientras

fin funcion

# BUSQUEDAD DE ANCHURA

Búsqueda en anchura es un algoritmo para recorrer o buscar elementos de un grafo(usado frecuentemente en arboles). Se comienza por la raíz y se explora todos los hijos de este nodo. A continuación se explora cada unos de los hijos de los hermanos y así sucesivamente hasta encontrar la solución.

[](https://sites.google.com/a/utp.edu.co/inteligencia-artificial/algoritmo-busqueda-en-anchura/2.png?attredirects=0)

Pseudocogido Algoritmo:

Establecer nodo origen

Evaluar primer hijo

si cumple, establecer como origen y salir

si valido, repetir búsqueda a partir del nuevo estado

sino valido, repetir búsqueda para todos los hermanos

si encuentra, establecer como origen y salir

si no encuentra, marcar al padre como no valido

establecer origen como abuelo y seguir buscando.

La búsqueda primero en anchura se puede implementar utilizando una estructura de tipo cola primero en entrar primero en salir, asegurándose que los nodos primeros visitados serán los primeros expandidos.

La principal desventaja de la búsqueda en anchura es los requisitos de memoria para almacenar todos los nodos que no han sido expandidos durante la búsqueda.

Continuaremos con el ejemplo del problema del agente de viajes con la misma formulación vista en la búsqueda en profundidad.

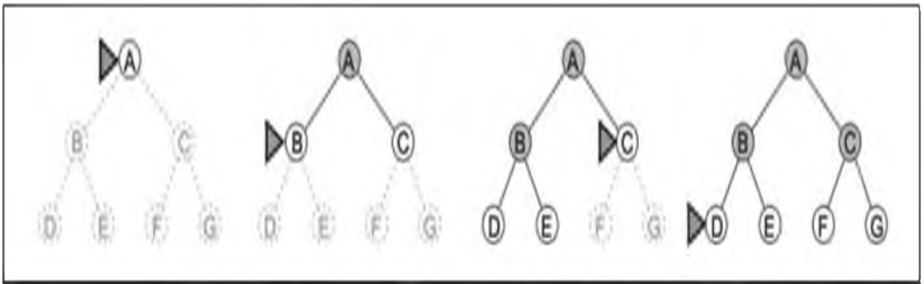
Si tenemos una estructura de datos de tipo cola para implementar esta estrategia de búsqueda:

La búsqueda primero en amplitud o en anchura es una estrategia sencilla en la que se expande primero el nodo raíz, a continuación se expanden todos los sucesores del nodo raíz, después sus sucesores, etc.

En general, se expanden todos los nodos a una profundidad en el árbol de búsqueda antes de expandir cualquier nodo del próximo nivel.

La búsqueda primero en anchura se puede implementar utilizando una estructura de tipo cola primero en entrar primero en salir, asegurándose que los nodos primeros visitados serán los primeros expandidos.

La principal desventaja de la búsqueda en anchura es los requisitos de memoria para almacenar todos los nodos que no han sido expandidos durante la búsqueda.



# CONCLUSIONES

* Logramos entender la busqueda de anchura y profundidad con el ejemplo del robot.
* Persibir diferentes factores los cuales implican en la busqueda.
* Solucionar diferentes problemas implementando este modelo optimo para los problemas futuros.
* El mundo real es más complejo de lo que se formula en los problemas para solucionar por computadora, sin embargo asumimos que los seres humanos para encontrar soluciones tampoco requieren de mucha información, o al menos no requiere conocer todo el universo para encontrar soluciones buenas. Por ejemplo, no requerimos de mucha información, ni de mucho tiempo para seleccionar una botella de refresco que compramos en el supermercado. Esto justifica en parte lo que hacemos cuando reducimos nuestro problema. Aún cuando por su simplicidad sean problemas de juguete.

VII REFERENCIAS

<https://sites.google.com/a/utp.edu.co/inteligencia-artificial/algoritmo-busqueda-en-anchura>

<http://repositorio.utp.edu.co/dspace/bitstream/handle/11059/2609/629892G643.pdf?sequence=1&isAllowed=y>

<http://ia2gerardoloor.blogspot.com/2014/10/estrategias-de-busqueda-no-informada.html>

1. [↑](#footnote-ref-1)