

Estrategias de búsqueda

Inteligencia Artificial

1^{do} Angely Mendez

Escuela de Informática

Universidad Nacional de Trujillo

Trujillo, Perú

t052701020@unitru.edu.pe

2^{ero} Ciara Mendez

Escuela de Informática

Universidad Nacional de Trujillo

Trujillo, Perú

t022700920@unitru.edu.pe

Resumen—Este documento es una investigación que describe cinco de las estrategias de búsqueda en el área de Inteligencia Artificial, las cuales se dividen en búsquedas no informadas y son: Búsqueda en amplitud(BFS), Búsqueda en profundidad(DFS), Búsqueda de costo uniforme(UCS) e informadas: Búsqueda voraz (greedy) y Búsqueda A*. Las cuales realizan acciones mecanizadas o resuelven problemas, reduciendo a buscar en un espacio de estados. Por lo que en este artículo se define y explica a que se refiere cada búsqueda, además se menciona una investigación que aplica dicha estrategia y cómo es útil usarlos en implementaciones para beneficio de la sociedad.

Palabras claves:—Estrategias de búsqueda, inteligencia artificial.

I. INTRODUCCIÓN

El tema de búsquedas en inteligencia artificial(IA) es prioritario, dado que al realizar acciones mecanizadas o resolver problemas, se reduce a buscar en un espacio de estados. Estas búsquedas utilizan dominios como la deducción, elaboración de planes de actuación, razonamientos de sentido común, prueba automática de teoremas, etc.

En la disciplina de la IA se estudian búsquedas no informadas o ciegas (búsqueda primero en amplitud, primero en profundidad, profundidad iterativa, de costo uniforme, etc.) y búsquedas informadas o inteligentes (búsqueda avara, A*,

IDA*, A* restringida por memoria simplificada, ascenso de cima (hill-climbing), etc.)

En este informe se presenta información al respecto, el cual está organizado de la siguiente manera: en primer lugar, se explican los conceptos teóricos: definiciones de cada una de las estrategias, luego se da énfasis en una investigación que está relacionada con ellas y para finalizar las conclusiones más relevantes.

II. ESTRATEGIAS DE BÚSQUEDA

II-A. Búsqueda primero en amplitud (BFS)

Es un algoritmo para atravesar o buscar estructuras de datos de árboles o gráficos. Explora todos los nodos en la profundidad actual antes de pasar a los nodos en el siguiente nivel de profundidad. [9] Un algoritmo de búsqueda primero en amplitud busca horizontalmente. La siguiente Figura 1 demuestra esto:

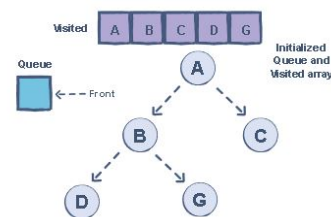


Figura 1. Ejemplo de Grafo dirigido para Búsqueda en Amplitud .

En la Figura 1 tenemos un gráfico dirigido de cinco nodos, siendo G el nodo a buscar. Los nodos se marcarán como visitados utilizando la *visited* matriz, mientras que los nodos adyacentes se agregarán a la *queue*.

[6] menciona las siguientes ventajas:

- Si sabemos que una respuesta estará cerca del nodo raíz, un BFS es más eficiente. Por ejemplo, en el ejemplo del árbol genealógico, podríamos estar buscando un descendiente directo que vivió 50 años después que el antepasado en el nodo raíz, no 300 años después.
- Los BFS son excelentes para calcular la ruta más eficiente entre dos nodos. Por esa razón, son excelentes para aplicaciones de GPS y mapas e incluso para encontrar personas cercanas en las redes sociales.

A continuación, una investigación respecto a la Búsqueda en Amplitud:

1. En su Tesis “*Diseño, implementación y aplicación de una estrategia de búsqueda preferente por amplitud, para uso multidireccional sobre sistemas distribuidos o de procesamiento en paralelo, usando un simulador de escenarios, construido para el trazado de rutas en robótica móvil.*” Maestría en Instrumentación Física. Universidad Tecnológica de Pereira. Pereira-Colombia; [5] plantea la construcción de un simulador de escenarios en dos dimensiones, el cual permitirá que un robot de software (llamado Softbot) y los diferentes elementos que conforman un ambiente, se utilicen como sistema de pruebas en la búsqueda de una ruta apropiada para ir de un punto a otro. Como herramienta para la determinación de la ruta se utilizan técnicas de Inteligencia Artificial (IA), lo que se logra mediante la sucesión de cambios de estado a través de acciones o transformaciones, el uso de estrategias de búsqueda (Ver Figura 2) y heurísticas generales. Finalmente se implementa la capacidad de interactuar no solo uno, sino varios agentes de software, trabajando en forma cooperativa, de tal forma que con la ayuda de ellos y

aplicando la estrategia de búsqueda multidireccional sea posible establecer una ruta.

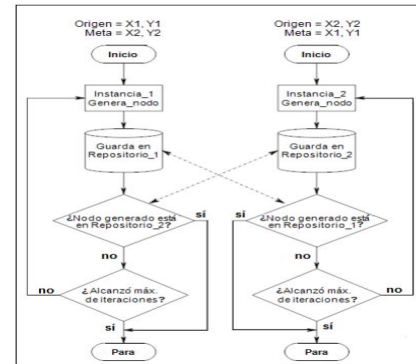


Figura 2. Algoritmo de la búsqueda preferente por amplitud bidireccional.

II-B. Búsqueda en profundidad (DFS)

Es un algoritmo para atravesar o buscar estructuras de datos de árboles o gráficos que utiliza la idea de retroceder. Explora todos los nodos avanzando si es posible o retrocediendo.[10] La siguiente Figura 3 demuestra esto:

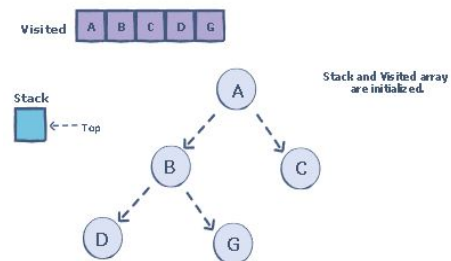


Figura 3. Ejemplo de Grafo dirigido para Búsqueda en Profundidad .

En la Figura 3 Tenemos un gráfico dirigido de cinco nodos, siendo G el nodo a buscar. Los nodos se marcarán como visitados utilizando la *visited* matriz, mientras que los nodos adyacentes se agregarán a *lastack*.

[6] menciona las siguientes ventajas:

- Los DFS son excelentes para crear juegos porque permiten que el algoritmo explore las posibilidades de ganar y perder.
- Si sabemos que la respuesta que buscamos estará en la parte inferior de un árbol, un DFS normalmente será

más rápido que un BFS. Por ejemplo, podríamos estar buscando descendientes vivos de un árbol genealógico largo, y todos los vivos estarán en la parte inferior del árbol.

- DFS suele ocupar menos memoria que BFS.

A continuación, una investigación respecto a la Búsqueda en Profundidad:

1. En la Tesis “*Modulo de creación de laberintos en Unity*” Ingeniería Informática. Universidad Autónoma de Madrid. Madrid-España; [8] se basa en poder proporcionar la misma facilidad para un usuario de crear laberintos aleatorios como la que se proporciona por el entorno de desarrollo mediante interfaz. Para ello se han creado unos scripts en donde se encapsulan las necesidades más imprescindibles que puede tener un usuario a la hora de crear un laberinto. Los algoritmos de generación de laberintos son varios, pensados para distintas necesidades de generación, cada cual tiene sus pros y sus contras. En este caso el Algoritmo DFS (Ver Figura 4), también conocido como búsqueda primero en profundidad, el cual genera un laberinto tradicional perfecto en donde no existen zonas cerradas a las cuales no se pueda acceder y no posee bucles por lo que si se recorriese siempre el lado izquierdo o derecho del laberinto se alcanzarían todas las zonas sin importar la forma del laberinto.

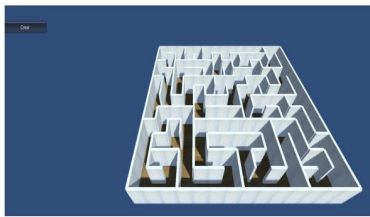


Figura 4. Laberinto con generación de algoritmo búsqueda en profundidad.

II-C. Búsqueda de costo uniforme (UCS)

Es un algoritmo de búsqueda no informado que utiliza el costo acumulativo más bajo para encontrar una ruta desde el origen hasta el destino. Los nodos se expanden, comenzando

desde la raíz, de acuerdo con el costo mínimo acumulativo [4].

A continuación se muestra un ejemplo de UCS en acción. Usaremos el gráfico (Ver Figura 5):

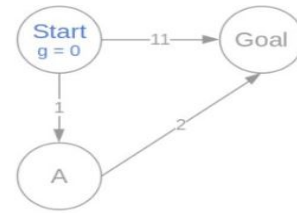


Figura 5. Ejemplo de Grafo de Búsqueda de costo uniforme (UCS).

Al principio, solo el nodo de inicio está en la frontera y su g valor es 0.

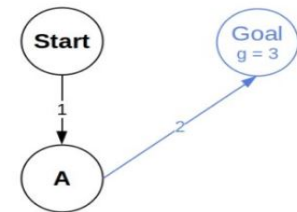


Figura 6. Ejemplo de Grafo de Búsqueda de costo uniforme (UCS).

Después de expandir *Goal*, se observa en la Figura 6 que pasa la prueba del objetivo, por lo que concluimos que la ruta óptima es *Start-A-Goal*.

A continuación, una investigación respecto a la Búsqueda de costo uniforme:

1. En el artículo titulado “*Selección de ubicación de instalaciones de construcción para proyectos de mantenimiento de infraestructura lineal urbana utilizando el método de búsqueda de costo uniforme*”, Mansour, A.G., Eid, M.S. & Elbeltagi, E.E [7] presentan un modelo para asignar ubicaciones de instalaciones de construcción (CFL) a cada segmento de trabajo en proyectos de infraestructura lineal que se caracterizan por cambios frecuentes de ubicación del sitio de construcción a medida que avanza el trabajo. El modelo propuesto utiliza un algoritmo de búsqueda de árbol—Búsqueda

uniforme de costos (UCS)—que garantiza una solución óptima global. El modelo propuesto, con el algoritmo de poda, es capaz de resolver un problema de espacio de solución de 4 billones en menos de 3000 ms utilizando una máquina de 8 GB de RAM y 2 GHz. El modelo propuesto se puede integrar en los procesos de toma de decisiones de planificación y diseño del sitio para proyectos de infraestructura lineal para reducir el costo general del diseño del sitio.

II-D. *Búsqueda voraz (greedy)*

La Universidad Veracruzana en el curso de Inteligencia Artificial [2] establece que la búsqueda voraz greedy expande el nodo más cercano al objetivo, asumiendo que probablemente conduzca más rápidamente a la solución. La función de evaluación $f(n)$ es la función heurística $h(n)$

$$f(n) = h(n)$$

donde $h(n)$ = costo estimado del camino más barato desde el nodo n hasta el objetivo.

El término Voraz (Greedy) ó Avaro es porque en cada paso trata de situarse tan cerca del objetivo como pueda, seleccionando el nodo con menor función de evaluación $f(n)$.

- No necesariamente brinda la solución óptima.
- Al igual que los otros métodos es necesario verificar los “callejones sin salidas” (no expandir estados repetidos).

A continuación, una investigación respecto a la búsqueda voraz (greedy):

1. La investigación cuyo título es “*Estimación robusta de canales de comunicaciones de ultra banda ancha*” [11], explica que los sistemas UWB (ultra-wide band) transmiten señales a baja potencia a través de canales de comunicaciones que operan en entornos cerrados y en distancias cortas, lo que origina múltiples trayectorias de propagación.
- Además, el ruido que afecta a estos canales se puede caracterizar mediante el uso de modelos estadísticos de colas más pesadas que las exhibidas por la distribución

gaussiana. En este trabajo, se propone un enfoque robusto para la estimación de los parámetros de los canales UWB basado en la mediana ponderada.

Específicamente, se desarrolla un **algoritmo de búsqueda voraz**, ver Figura 7, que aprovecha la característica poco densa de la respuesta impulsiva del canal, donde las ganancias y los retardos de los canales relevantes se determinan aplicando la mediana ponderada sobre una versión escalada y desplazada de la señal recibida.

El algoritmo propuesto se evalúa usando extensas simulaciones en las que el rendimiento del algoritmo propuesto supera el desempeño del algoritmo de búsqueda voraz tradicional para diferentes niveles de ruido impulsivo.

Entrada	Señal recibida UWB: r ; diccionario parametrizado: D ; número de iteraciones: L_0 ; tasa de adaptación del parámetro de regularización: β ; energía residual mínima: ε
Inicialización	Contador de iteraciones $l = 1$ Valor inicial del parámetro de regularización $\mu^{(0)} = D^T r_{\infty}$ Estimación en $l = 1$: $\hat{\alpha}^{(1)} = 0_{N_{\text{ms}}}$
	Para la n -ésima entrada de $\hat{\alpha}$, donde $n = 1, 2, 3, \dots, N_{\text{ms}}$. Calcular:
Paso A	$\hat{\alpha}_n^{(j+1)} = \text{MEDIANA} \left[\left d_{j,n} \right \frac{r - \sum_{i=1, i \neq n}^{N_{\text{ms}}} d_i \hat{\alpha}_i^{(j)}}{d_{j,n}} \right]^{1/p}$ $\hat{\alpha}_n^{(j+1)} = \begin{cases} \hat{\alpha}_n^{(j+1)} & \text{Si } \ w_n\ - \ w_n - \hat{\alpha}_n^{(j+1)} d_n\ > \mu^{(j)} \\ 0 & \text{en caso contrario} \end{cases}$
	Actualizar el parámetro de regularización y la estimación de α
Paso B	$\mu^{(j)} = \mu^{(j)} \beta^j$ $\hat{\alpha}^{(j+1)} = [\hat{\alpha}_1^{(j+1)}, \hat{\alpha}_2^{(j+1)}, \dots, \hat{\alpha}_{N_{\text{ms}}}^{(j+1)}]$
	Chequear el criterio de parada
Paso C	Si $\frac{r - D\hat{\alpha}^{(j+1)}}{L_0} > \varepsilon$ y $l \leq L_0$ entonces hacer $l = l + 1$ y volver al paso A
	En otro caso, terminar.
Salida	Señal poco densa reconstruida $\hat{\alpha}$

$\|\cdot\|_{\infty}$ determina el máximo valor absoluto de los elementos de un vector.
 $0_{N_{\text{ms}}}$ Denota un vector nulo de N_{ms} elementos

Figura 7. Algoritmo de búsqueda voraz usando mediana ponderada.

II-E. *Búsqueda A**

A* es un algoritmo de búsqueda inteligente o informada que busca el camino más corto desde un estado inicial al estado meta a través de un espacio de problema, usando una heurística óptima [3].

Como ignora los pasos más “cortos.” en algunos casos rinde una solución subóptima.

Los Fundamentos de la búsqueda A* son:

- Minimiza el costo estimado total de la solución.
- Evalúa los nodos combinando

$$g(n) \text{ y } h(n)$$

$g(n)$: costo de haber alcanzado n

$h(n)$: costo para llegar desde n hasta el objetivo

$f(n) = g(n) + h(n) \rightarrow$ costo más barato estimado de la solución a través de n .

- En cada paso se expande el nodo con el valor más bajo de $f(n)$, ó sea, de $g(n) + h(n)$.
- La búsqueda A* es óptima siempre y cuando la función heurística $h(n)$ sea una heurística admisible, es decir nunca sobreestime el costo de alcanzar el objetivo.
- Son funciones optimistas.

Una investigación sobre la búsqueda A* es:

1. La *“Implementación del Algoritmo A-Star para la Búsqueda de Rutas Cercanas al Punto de Refugio de Evacuación de Tsunami”* [1], evaluó el Purus Kelurahan de Padang, ciudad de Indonesia, que tiene un área de 0,86Km2 que consta de 8RW y 28RT, tiene una población de 8,075 personas con una densidad de 11,875 que tiene el potencial de verse afectado por desastres si ocurre un tsunami en la ciudad de Padang.

Las personas en esta ciudad siempre están acosadas por el miedo y se sienten amenazadas si ocurre un terremoto o un tsunami. Por esta razón la investigación menciona que es necesario que haya un proceso educativo para que la comunidad tenga una cultura de concientización sobre desastres en forma de un sistema que sea capaz de informar a la comunidad cuál es la zona segura más cercana a la que pueden llegar y la ruta que tienen que recorrer, obsérvese Figura 8.

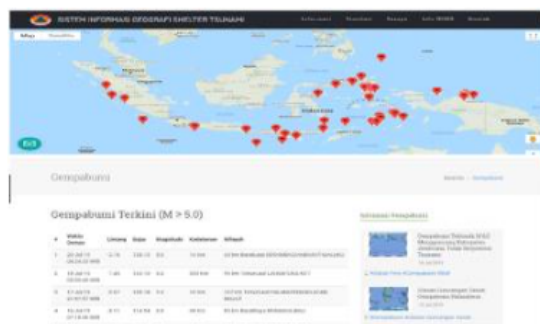


Figura 8. Página del Terremoto.

Por lo tanto, se creó un sistema de información geográfica (SIG) utilizando el método del Algoritmo A-Star.

El **algoritmo A-Star** utiliza la estimación de distancia más cercana para alcanzar la meta (objetivo) y tiene un valor heurístico que se utiliza como base para la consideración, ver Figura 9. En este sistema hay un camino alternativo y muestra la cantidad de capacidad y la distancia desde el refugio a abordar.

Los investigadores utilizaron un método descriptivo mediante la recopilación de datos e información sobre rutas de evacuación de tsunamis y refugios en el área de Purus de la ciudad de Padang que ayuda a minimizar las víctimas en casos de desastre.

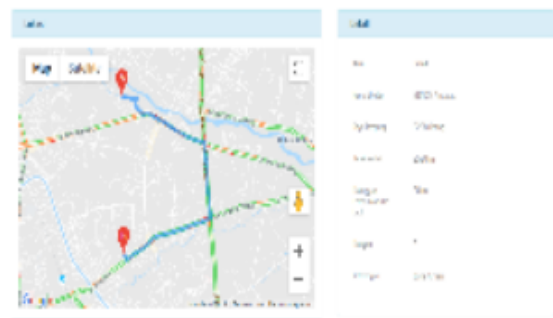


Figura 9. Página de la ruta del refugio más cercano.

III. CONCLUSIÓN

Este informe presentó información relevante acerca de las cinco estrategias de búsqueda no informadas y son: Búsqueda en amplitud(BFS), Búsqueda en profundidad(DFS), Búsqueda de costo uniforme(UCS) e informadas: Búsqueda voraz (greedy) y Búsqueda A*, se ha explicado los conceptos teóricos relacionados con ellas. Además de permitir conocer algunas investigaciones que se vienen realizando año a año, entendiendo las diversas aplicaciones de ellas en la vida diaria. Concluimos respecto a ello que la mayoría analiza la problemática, establece el proceso y propone resolver problemas utilizando el método correcto en los que para llevarlos a cabo de forma efectiva y eficientemente, utilizan bases teóricas sobre el conocimiento heurístico.

REFERENCIAS

- [1] R. Astri et al. Implementación del algoritmo a-star para la búsqueda de rutas cercanas al punto de refugio de evacuación de tsunami. *Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi)*, 4(2):254–259, 2020.
- [2] I. y. G. Benítez. Inteligencia artificial ii - resolución de problemas mediante búsquedas 3. búsquedas informadas. URL <https://www.uv.mx/personal/edbenitez/files/2010/09/CursoIA10-II-3.pdf>. (Accedido el 06/23/2022).
- [3] EcuRed. Algoritmo de búsqueda heurística a*. URL <https://www.ecured.cu>. (Accedido el 06/23/2022).
- [4] E. I. C. f. S. D. F. Hasan. ¿qué es la búsqueda de costo uniforme? <https://www.educative.io/answers/what-is-uniform-cost-search>, Apr. 26 2021. (Accesed on 06/23/2022).
- [5] A. González Ospina and H. B. Cano Garzón. Diseño, implementación y aplicación de una estrategia de búsqueda preferente por amplitud, para uso multidireccional sobre sistemas distribuidos o de procesamiento en paralelo, usando un simulador de escenarios, construido para el trazado de rutas en robótica móvil. 2011. URL <https://hdl.handle.net/11059/2609>.
- [6] Learnhowtoprogram.com. Algoritmos de búsqueda de profundidad y amplitud. URL <https://www.learnhowtoprogram.com/computer-science/graph-theory/depth-and-breadth-search-algorithms>. (Accedido el 06/23/2022).
- [7] A. G. Mansour, M. S. Eid, and E. E. Elbeltagi. Construction facilities location selection for urban linear infrastructure maintenance projects using uniform cost search method. *Soft Computing*, 26(3):1403–1415, 2022.
- [8] R. Montero Sacristán et al. Modulo de creación de laberintos en unity. B.S. thesis, 2016. URL <http://hdl.handle.net/10486/677166>.
- [9] E. A. Team. ¿qué es la búsqueda primero en amplitud? <https://www.educative.io/answers/what-is-breadth-first-search>, Abril, 04 2019. (Accedido el 06/23/2022).
- [10] E. A. Team. What is depth first search?, Abril, 04 2019. URL <https://www.educative.io/answers/what-is-depth-first-search>. (Accessed on 06/23/2022).
- [11] N. A. Yáñez-Sánchez, J. M. Ramírez, J. L. Paredes, Á. D. Pinto, J. M. Torres, and M. L. Pérez. Estimación robusta de canales de comunicaciones de ultra banda ancha. *Revista Ingenierías Universidad de Medellín*, 18 (34):181–197, 2019.