

LA INTELIGENCIA ARTIFICIAL Y LA RESOLUCIÓN DE PROBLEMAS

Luis Carlos Torres Soler
Profesor Universidad Autónoma de Colombia
lctorress@gmail.com.

Néstor Manuel Garzón Torres
Profesor Universidad Autónoma de Colombia
nestormg@gmail.com.

Recibido: 03-06-2015, aceptado: 27-06-2015, versión final: 27-06-2015

RESUMEN

Los procesos de búsqueda de solución a un problema se han estudiado con amplitud por la necesidad del ser humano para satisfacer necesidades tanto biológicas como científicas, por tanto, su estudio es algo en que se ha involucrado la inteligencia artificial. Varios problemas complejos, considerados de no aplicación computacional se han solucionado, o por lo menos hallado una solución aproximada, gracias al empleo de las heurísticas, las cuales son importantes entender en toda su dimensión para comprender la gran aplicabilidad que poseen y el beneficio que proveen.

Palabras clave: *heurística, búsqueda A*, búsqueda voraz, haversine.*

ABSTRACT

The processes of searching for a solution to a problem have been studied extensively by the need of the human being needs to meet both biological and scientific, therefore, its study is something that has been involved the artificial intelligence. Various complex problems, considered not computational application have been resolved, or at least found an approximate solution, thanks to the use of heuristics, which are important in understanding all its dimension to the understanding of the great applicability to possess and the benefit they provide.

Keywords: *heuristic, A* search, greedy search, haversine.*

INTRODUCCIÓN

El desarrollo tecnológico crece de manera acelerada facilitando mecanismos y herramientas para beneficio de la sociedad, en su aplicabilidad y para solucionar problemas que se consideran complejos. Una de las disciplinas que diferentes técnicas o métodos ha desarrollado y propuesto para solucionar problemas, es la Inteligencia Artificial (IA); la cual, por su interés en desarrollar máquinas que razonen como lo hace el ser humano, ha llevado a construir diferentes métodos que facilitan en tiempo y en espacio, claro está que empleando el computador, hallar solución a problemas que por métodos operacionales corrientes llevaría bastante tiempo y el requerimiento de un computador de amplias capacidades.

Los métodos de búsqueda informada o basados en heurísticas han mostrado ser eficientes a la hora de encontrar solución a un problema que tratado con la algoritmia o programación tradicional se convierte en una tarea muy dispendiosa y, en ocasiones, hasta en un problema de tipo NP (no tratable computacionalmente).

La búsqueda de una ruta óptima entre diferentes ciudades es un problema que para su solución se han construido diferentes algoritmos y estrategias, pero a la vez métodos que empleando el computador determine una solución lo más aproximada.

ASPECTOS GENERALES

La inteligencia artificial es una disciplina joven, en proceso de consolidación de conceptos y métodos, pero hoy día es una herramienta de ayuda para hallar solución a problemas complejos. Muchos problemas han tenido su respuesta óptima pero la IA no es una receta que pueda darle solución a todo problema. La IA se compone de un conjunto de técnicas que de diferente manera encuentran una solución (no necesariamente la mejor) a un problema que en algunas ocasiones es inabordable por una persona [García, 2012].

Un problema puede tener muchas visiones y enfoques, también ser difícil de abordar; sin embargo, la abstracción del problema permite detallar posibles soluciones. Los problemas que se abordan por medio de la IA requieren de modelamiento simbólico para contextualizar las configuraciones posibles de los estados. En pocas palabras, siempre se tiene cuál es el objetivo. Se abstrae el problema, sus condiciones y posibles estados, se identifica las posibles transformaciones entre estados y se considera el mejor método. Los problemas que se solucionan por técnicas de IA, en sí no tienen una fórmula explícita o algoritmo definido, porque no todo es posible por medio de ello; se debe emplear un método de búsqueda.

El artículo realiza una descripción de algunos métodos de búsqueda de solución a los problemas, en especial, centrada en el mismo proyecto de investigación "Sistema de búsqueda de rutas (óptimas) entre dos puntos". En este tipo de problema, al aplicar las técnicas normales de computación y algoritmia se gasta mucho tiempo computacional, de ahí que una facilidad es empleando técnicas y herramientas de la inteligencia artificial.

ALGORITMOS GENÉTICOS (AG)

Los algoritmos genéticos son un tipo de algoritmo que facilita hallar soluciones a problemas de optimización. Hallar una ruta óptima es uno de ellos. Un caso particular es la solución del famoso problema del Agente Viajero (conocido por sus siglas en inglés TSP -*Travelling Salesman Problem*-). Se trata de determinar el recorrido con menor coste que debe realizar un agente viajero para visitar todos los nodos ubicados sobre un mapa e interconectados entre todos, conociendo la distancia terrestre entre cada nodo, y con la condición de pasar solamente una vez por cada, a excepción del nodo de salida que es el mismo nodo de llegada. El problema trata de establecer cuál es la ruta óptima para recorrer todos en el mínimo de distancia. También, ¿cuál es la mejor ruta?

Encontrar una solución para diez nodos o más es un proceso que puede tardar las 24 horas de computación; sin embargo, los algoritmos genéticos, por trabajar con población de soluciones, su proceso es más rápido, además, que las pruebas indican que haya varias rutas a la vez, hecho que por la programación tradicional no encontrará sino una sola. Cuando el número de nodos aumenta considerablemente y las interconexiones entre cada uno es mayor, este se convierte en un problema de amplia computación.

Para emplear los algoritmos genéticos, se siguen unos pasos, que es lo que determina que se llame algoritmo, y lo de genético se desprende de emplear los procesos que se realizan en la naturaleza para la preservación de las especies.

- Construir un modelo para representar al problema.

- Se determina cuál es el objetivo buscado (¿Qué soluciones son válidas?)
- Se define un criterio que valore las soluciones del problema.
- Se aplican operadores de selección, cruce y mutación para hallar otro grupo de soluciones [Torres y Garzón, 2012].
- Se aplican operadores de selección, cruce y mutación para hallar otro grupo de soluciones [Torres y Garzón, 2012].

La figura 1 representa un posible mapa de carreteras para cinco ciudades, al lado su respectiva matriz de adyacencias. Se debe elegir la representación más intuitiva. Para el caso, parece más eficiente utilizar la matriz (tabla) ya que facilita su modelación mediante un arreglo de dos dimensiones.

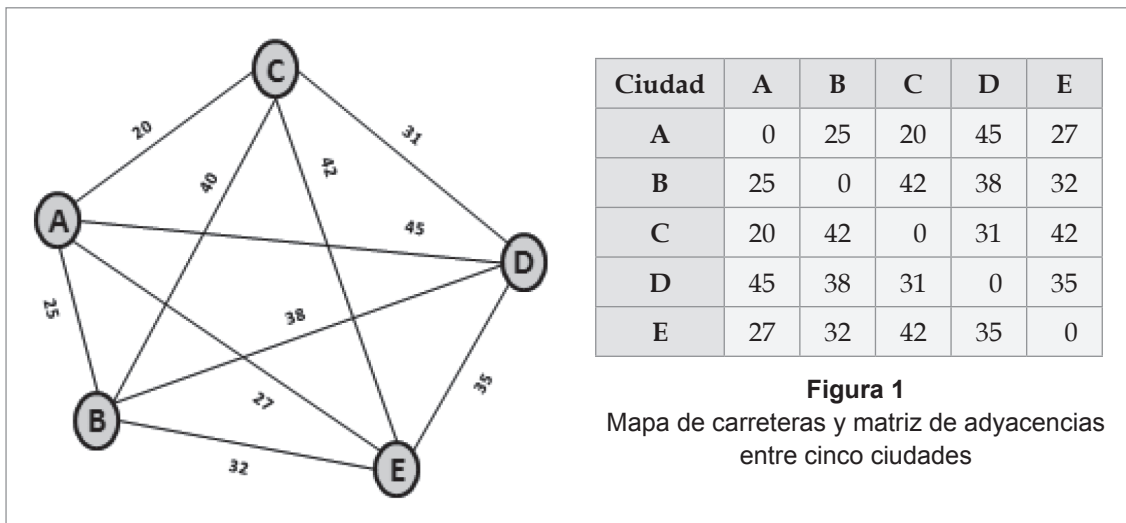


Figura 1
Mapa de carreteras y matriz de adyacencias entre cinco ciudades

Modelada la información del problema, se necesita una forma para representar la solución u objetivo del problema. Una manera simple de representación es una lista de ciudades siempre y cuando cumpla con las siguientes premisas:

- La lista debe comenzar y terminar con la misma ciudad.
- Deben aparecer todas las ciudades en la lista.

- No se debe repetir ciudad alguna, excepto la primera y la última.

¿Cómo encontrar la mejor solución? En teoría, se hace:

- Generar la lista de todas las posibles soluciones, y
- Seleccionar la que tenga la mejor evaluación (menor distancia).

Cuando el número de ciudades es pequeño, el proceso de cómputo es poco. Cuando este número crece, la solución escapa a un manejo manual y se requiere de un computador con amplias capacidades. Para N ciudades, el número de posibles rutas es $N!$ (N factorial).

- Cinco ciudades: $5! = 120$;
- Ocho ciudades: $8! = 40320$;
- Diez ciudades: $10! = 3.628.800$;
- Quince ciudades: $15! = 1.307.674.368.000$.

Esto significa que los métodos de algoritmia, con un computador que emplee una milésima de segundo para evaluar cada ruta, se necesitaría un tiempo de 41.46 años para evaluar todo el conjunto de estados.

Es un tipo de problema que pertenece a la clase NP, cuya complejidad no permite ser resoluble en tiempo moderado por computador. La solución a este tipo de problemas se está logrando por métodos de búsqueda de la inteligencia artificial, por ejemplo, algoritmos genéticos.

MÉTODOS DE BÚSQUEDA NO INFORMADA

La IA aborda problemas de complejidad de alto nivel, en particular buscar un estado concreto o un camino determinado; en situaciones donde se posee algún tipo de información se emplean técnicas de búsqueda informada. Pero si se carece de información alguna se emplea algoritmos de búsqueda no informada.

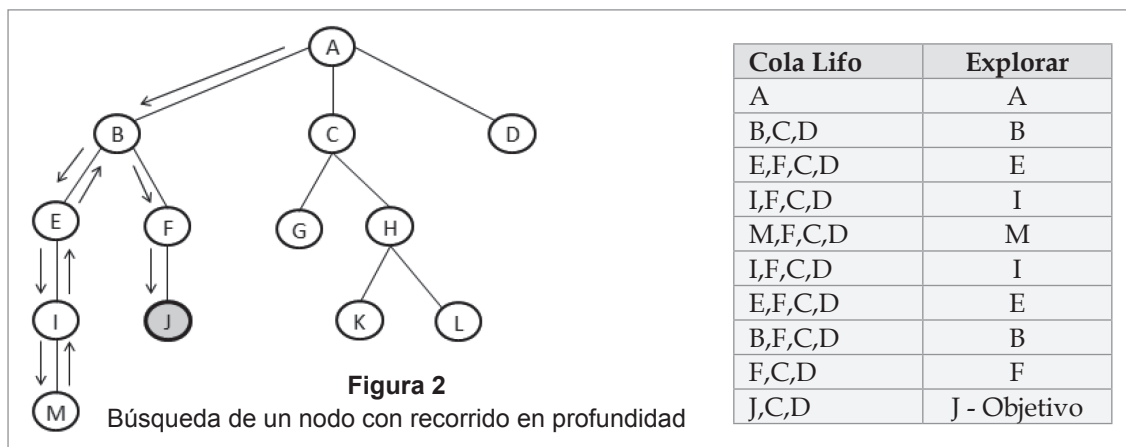
Los métodos de búsqueda tratan de encontrar una solución posible dentro del espacio de estados posibles. Las búsquedas no informadas tienen sus ventajas y desventajas. La idea es buscar una posible solución de una manera rápida y eficiente, para ello se trata de hallar una ruta entre un par de nodos, dos métodos interesantes son: (1) búsqueda en profundidad y, (2) búsqueda en amplitud.

BÚSQUEDA EN PROFUNDIDAD

La búsqueda en profundidad recorre un grafo, nodo por nodo, rama por rama, desde la raíz siguiendo por la izquierda, hasta llegar a la profundidad máxima de la rama (nodo terminal). Retorna un poco y continúa con la siguiente rama cuando no llega al nodo objetivo.

La figura 2 muestra una representación gráfica de la búsqueda en profundidad. Para el proceso se utiliza una pila LIFO (Last In First Out). Esta pila tiene dos operaciones: (1) introducir un dato en la pila y, (2) extraer un dato de la pila. El dato que se extrae de la pila será el último en haber sido introducido. Por ejemplo, el objetivo es hallar el nodo J partiendo del nodo A.

Teniendo el árbol, se hace búsqueda no informada. ¿Cómo es en profundidad? Siendo A el nodo inicial y J el nodo objetivo a alcanzar, se procede como sigue: buscar en todas las ramas que cuelgan, de izquierda a derecha, explorar cada rama hasta llegar a una hoja. Para cada hijo se busca a su vez en profundidad, parando cuando se encuentre el objetivo.



La ruta a seguir es: $A \rightarrow B \rightarrow E \rightarrow I \rightarrow M \rightarrow I \rightarrow E \rightarrow B \rightarrow F \rightarrow J$.

El costo del proceso es pesado. La operación más costosa es llegar a los nodos más profundos (hojas). Si la solución (nodo objetivo) se encuentra en un nivel d del árbol, los nodos de este sólo se habrán generado una vez, los del nivel anterior ($d-1$) dos veces y así sucesivamente. El número de nodos total generados en un árbol con factor de ramificación b y profundidad d será:

$$\text{Número nodos} = b(d) + b^2(d - 1) + b^3(d - 2) + \dots + bd$$

Su grado de complejidad es $O(bd)$. Si no se conoce la profundidad a la que se encuentra el nodo objetivo y se tiene un espacio de búsqueda grande, este algoritmo resulta más eficiente que el de amplitud. No es un algoritmo completo. Si existe solución, se dirige por una rama profunda puede demorarse bastante. No

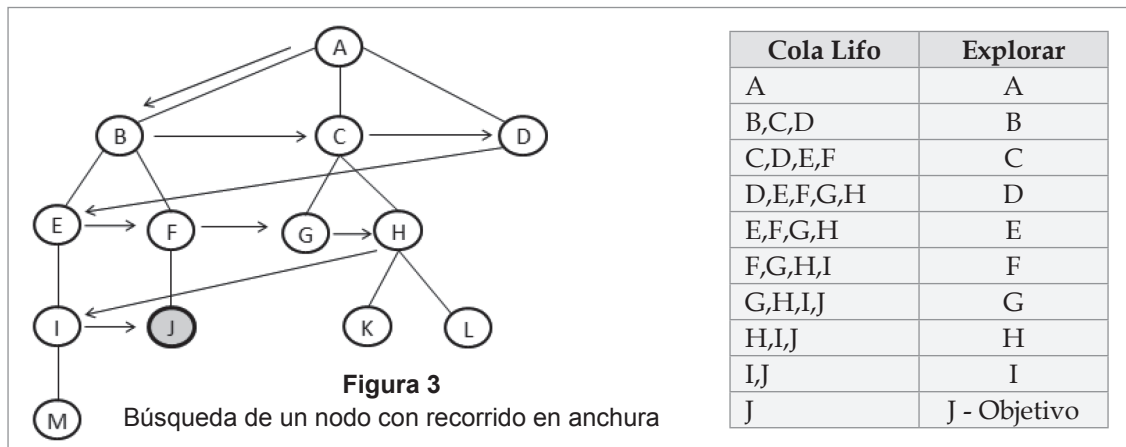
es óptimo. Puede encontrar la solución pero con un recorrido mayor del necesario.

BÚSQUEDA EN ANCHURA

La búsqueda en anchura es una opción para recorrer el espacio de estados al no conocer información alguna. Esta búsqueda recorre el árbol por niveles, partiendo del nodo raíz:

- a. Visitar primero el nodo raíz,
- b. Visitar todos sus hijos (nodos adyacentes),
- c. Para cada hijo (nodo) del paso anterior, se visitan sus hijos (nodos),
- d. Seguir el proceso.

El recorrido se implementa usando una cola FIFO (First In First Out) para almacenar los nodos frontera¹. La figura 3 representa un árbol a recorrer y cuyo objetivo es encontrar el nodo J partiendo de A (raíz).



Los nodos que con este algoritmo se visitan hasta encontrar el objetivo (J) son:

$A \rightarrow B \rightarrow C \rightarrow D \rightarrow E \rightarrow F \rightarrow G \rightarrow H \rightarrow I \rightarrow J$

La complejidad del algoritmo tanto espacial como temporal es exponencial, y se calcula por $O(n^p)$. Siendo n el número medio de sucesores, y p el nivel donde se alcanza la solución.

El algoritmo es sólo aplicable a problemas no demasiado amplios.

Es un algoritmo completo; siempre encontrará una solución si es que la hay, pero en general no es óptimo.

¹ La cola FIFO consiste en una estructura de datos en la que se pueden realizar dos operaciones: (1) almacenar un dato y, (2) sacar un dato, en cuyo caso se extrae el último almacenado.

Como se ha podido ver, la búsqueda en profundidad y la búsqueda en anchura utilizan como criterio de búsqueda solamente la aplicación de operadores definidos al construir el árbol de estados. Las dos se diferencian claramente en la estrategia a utilizar para recorrer el árbol.

Estos métodos hacen una búsqueda ciega y exhaustiva, lo que presenta problemas cuando el espacio de estados es grande. Se mejora expandiendo o seleccionando los estados con mayores expectativas de la solución, lo cual obliga a emplear una función de evaluación heurística $f(n)$ que guíe a dar una prioridad a estos estados.

Con métodos de búsqueda utilizando esta evaluación, se pueden presentar algunas situaciones en las cuales no siempre se garantiza encontrar una solución (de existir ésta). Además, no siempre se garantiza encontrar la mejor solución, aunque la encontrada se acerque bastante bien a la mejor.

BÚSQUEDAS INFORMADAS

Existen problemas con alto grado de complejidad en los que el paso correcto a tomar en busca de la solución, no puede determinarse a priori. Para ello se emplean heurísticas que permiten encontrar una posible solución, sin que llegue a ser la mejor necesariamente² [Torres y Garzón, 2012, p. 77].

² La palabra heurística viene de la palabra griega *HEURISKEIN* que significa descubrir, que también es origen de *EUREKA*. George Poyla al publicar el libro *How to solve it* (Cómo resolverlo) aplicó el término al estudio de métodos para descubrir e inventar técnicas para la resolución de problemas. En algunos contextos se utiliza el término como opuesto al algorítmico; en otros como una regla práctica que permite generar buenas soluciones sin desarrollar un exhaustivo proceso. En la IA, las heurísticas tienen un papel fundamental pues las estrategias permiten aprender de sus errores. Las heurísticas tratan de resolver problemas difíciles eficientemente. No garantizan encontrar la solución óptima, pero sí buenas aproximaciones con menos esfuerzo. Las heurísticas son adecuadas para problemas excesivamente complejos.

Problemas de este tipo, ofrecen información que puede ser útil en el momento de resolverse; gracias a ella se trata de mejorar la búsqueda para ir más directamente al objetivo, tratando de ir por ramas del árbol que parecen más prometedoras, en lugar de otras que lo son menos.

BÚSQUEDA HEURÍSTICA

La clave de las heurísticas es ahorrar tiempo y/o espacio de almacenamiento evitando recorrer caminos inútiles que no conducen a solución alguna. No consiste en eliminar parte del espacio de estados de búsqueda, sino en hacer uso de información adicional que guíe el recorrido a la solución.

En la mayoría de los espacios de estados, existe información que permite guiar los procesos de búsqueda, aunque no sea perfecta. Con certidumbre ayuda a resolver los problemas a través de un conjunto de estados sin que tenga definido un camino, esto se denomina heurística [Torres y Garzón, 2012, p. 78].

Las técnicas heurísticas mejoran la eficiencia del proceso de búsqueda de la solución. Algunas ventajas de las búsquedas heurísticas son:

- Evitan la explosión combinatoria, lo cual ya justifica su uso.
- No hallan una solución óptima pero una aproximación sirve muy bien.
- La búsqueda informada utiliza conocimiento específico del problema.
- Puede encontrar soluciones de una manera más eficiente.
- Una función heurística, $h(n)$, mide el coste estimado más barato desde el nodo n a un nodo objetivo.
- $h(n)$ se utiliza para guiar el proceso haciendo que en cada momento se seleccione el estado con operaciones más prometedoras.

- A pesar de que una aproximación heurística no puede resultar muy buena, en el peor de los casos, raras veces aparecen los peores casos en la solución y en la práctica.
- El trabajo de comprender por qué una heurística funciona o por qué no funciona, a menudo conduce a una mejor comprensión del problema.

- Códigos DANE de departamentos y códigos DANE de municipios asociados a cada departamento.
- Tipo de vía entre cada nodo, de acuerdo a clasificación Ministerio de transporte.

El conjunto de datos permitió la construcción de una Base de Datos con sus respectivas tablas relacionadas y, cuyo modelo básico se muestra en la figura 4.

CÓMO APLICAR HEURÍSTICAS

El proyecto de investigación, base del presente artículo, requirió recolectar un conjunto de datos de diferentes fuentes (DANE, Instituto Geográfico Agustín Codazzi, Ministerio de transporte,...), los cuales una vez organizados, facilitaron la obtención de los objetivos propuestos. Entre los datos recolectados figuran:

- Coordenadas geográficas de cada nodo (municipio) sobre el mapa vial colombiano. Estas son la Latitud: distancia de un punto de la tierra al Ecuador (latitud 0), expresada en grados(o), minutos(:) y segundos("); longitud: distancia de un punto de la tierra al meridiano de Greenwich (Inglaterra -zona este de Londres) cuya longitud es 0, expresada en grados(o), minutos(:) y segundos(");
- Distancia terrestre (kms.) entre cada nodo adyacente a otro (mapa vial colombiano y mapa vial de cada departamento).

Al observar la tabla de municipios, esta contiene los datos de longitud y latitud de cada uno, los cuales se utilizan dentro del software desarrollado para calcular la distancia en línea recta (DLR) entre dos municipios cualesquiera, utilizando para ello la fórmula del Haversine³. El cálculo de esta distancia, es el principal argumento para poder aplicar métodos de búsqueda de la ruta más corta entre dos nodos (municipios).

A continuación, se desarrolla un ejemplo para pasar la latitud y longitud de Bogotá y Medellín a radianes; los resultados se ven en la tabla 1. Como estas están dadas en grados, minutos y segundos es necesario convertirlas a formato sexagesimal, es decir, convirtiendo segundos en minutos y minutos en grados.

$$\text{Latitud Bogotá: } 04^{\circ} 36' 35'' = 4 + 36 * 1/60 + 35 * 1/3600 = 4.6097$$

$$\text{Longitud Bogotá: } 74^{\circ} 04' 54'' = 74 + 4 * 1/60 + 54 * 1/3600 = 74.0816666$$

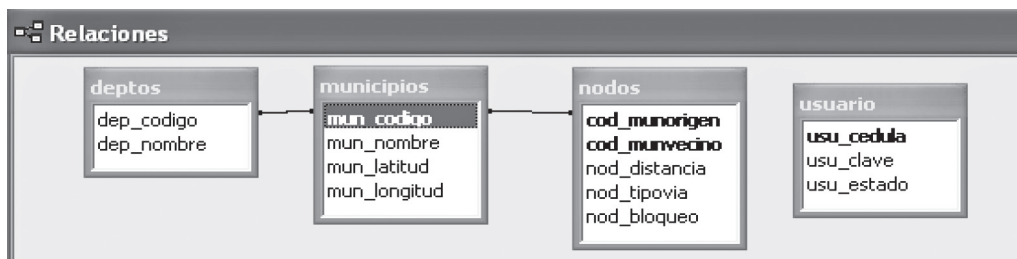


Figura 4

Modelo de datos para aplicar métodos de búsqueda con heurísticas

³ La fórmula de Haversine (semiverseno) es una ecuación para la navegación astronómica. Calcula la distancia del círculo máximo entre dos puntos del globo sabiendo su longitud y latitud.

Latitud Medellín: $06^{\circ} 15' 07'' = 6 + 15 * 1/60 + 7 * 1/3600 = 6.25194444$

Longitud Medellín: $75^{\circ} 33' 49'' = 75 + 33 * 1/60 + 49 * 1/3600 = 75.5636111$

Luego se deben convertir estas coordenadas en radianes. Se sabe que PI es igual a 180 grados,

entonces aplicando una sencilla regla de tres se convierte cualquier ángulo en formato sexagesimal a radianes:

180 grados \rightarrow PI radianes

1 grado \rightarrow x radianes

De donde $x = \text{PI}/180 = 3.141592/180$.

Tabla 1

Latitud y longitud de Bogotá y Medellín en grados, minutos, segundos; sexagesimal y radianes

Ciudad	Latitud			Longitud		
	o, ''	Sexagesimal	Radianes	o, ''	Sexagesimal	Radianes
Bogotá	4 36 35 N	4.6097N	0.08045444	74 04 54W	74.08166W	-1.29296897
Medellín	6 15 07 N	6.25194N	0.10911701	75 33 49W	75.5636111W	-1.31883379

Aplicando la fórmula del Haversine (Ecuación 1) una vez se tienen todos los datos, se encuentra que la distancia en línea recta entre las ciudades de Bogotá y Medellín es de 245.538 kms.

$$D = 2.r.\arcsin(\sqrt{\sin^2(\phi_2 - \phi_1)/2 + \cos(\phi_2).\cos(\phi_1).\sin^2(\lambda_2 - \lambda_1)/2})$$

Ecuación 1. Fórmula del Haversine

Donde,

r es el radio de la tierra (6372.8 kms. promedio),

Φ_1 es la longitud del punto 1,

Φ_2 es la longitud del punto 2,

Λ_1 es la latitud del punto 1,

Λ_2 es la latitud del punto 2.

Tanto la latitud como la longitud deben estar en radianes. El cálculo de esta distancia, es el principal argumento para poder aplicar métodos de búsqueda de la ruta más corta entre dos nodos (municipios).

PROCESO DE BÚSQUEDA

Con los datos disponibles para calcular la distancia en línea recta entre dos puntos, se aplicaron los métodos de búsqueda informada para encontrar la ruta a seguir entre dos nodos. Primero, se desarrolló el método de *Búsqueda voraz*

primero el mejor y, luego, se aplicó el método de *búsqueda A**.

BÚSQUEDA VORAZ PRIMERO EL MEJOR

La función de evaluación a utilizar en este método es $h(n)$, una heurística que estima el costo desde un nodo n hasta el nodo solución. En general, $h_{DLR}(n)$ = distancia en línea recta desde n al nodo solución.

- La *búsqueda voraz* expande el nodo más cercano al objetivo.
- Evalúa los nodos utilizando solamente la función heurística para minimizar el coste: $f(n) = h(n)$.
- La estructura de abiertos es una cola con prioridad.
- La prioridad la marca la función heurística (coste estimado del camino que falta hasta la solución).

- En cada iteración se escoge el nodo más “cercano” a la solución (el primero de la cola).
- No se garantiza la solución óptima.

Una primera aproximación al uso de este método de búsqueda consiste en encontrar la ruta que partiendo del nodo A tenga como destino el nodo Z. Ver figura 5.

En la figura 5 se puede observar que el arco o conexión existente entre nodos vecinos tiene un valor determinado (distancia) y, además, en la tabla de la parte derecha se puede observar la distancia en línea recta -DLR- desde un nodo en particular hasta el nodo objetivo Z⁴.

El procedimiento a seguir consiste en:

- Ubicar los nodos hijos (adyacentes) al nodo de salida A.
- De entre estos nodos, seleccionar aquel cuya distancia en línea recta desde este hasta Z sea la menor.
- Una vez ubicado este nodo, se convierte en el nodo actual, se ubican sus hijos y se continúa el proceso hasta llegar al objetivo.

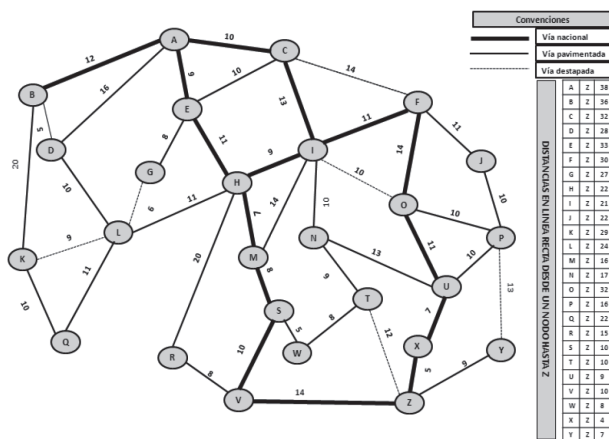


Figura 5

Distancias terrestres y en línea recta entre varios nodos

⁴ Nodo de salida: A, nodo objetivo: Z, distancia en línea recta = 38.

Tabla 2
Recorrido desde nodo A hasta nodo Z

Nodo actual	Nodos hijos	Distancia	DLR	Observ.
A	B	12	36	
	C	10	32	
	D	16	28	Menor
	E	9	33	
D	L	26	24	Menor
L	G	32	27	
	H	37	22	Menor
	K	35	29	
	Q	37	22	Menor
H	E	33	33	
	I	31	21	
	M	29	16	
R	V	42	15	Menor
R	V	50	10	Menor
V	S	60	10	
	Z	64	00	Objetivo

La ruta encontrada es A → D → L → H → R → V → Z (ver figura 6) con un coste de 64 unidades, pero no garantiza que sea la mejor solución. Se puede observar que en los nodos hijos del nodo L, hay dos (H y Q) con la misma función de evaluación, caso en el cual se tomó la H por estar primero que la Q. Al tomar el camino por el nodo Q nos muestra el resultado observado en la tabla 3.

Como se observa, al seguir la ruta Q → K la función de evaluación aumentó y por consiguiente se hace necesario bloquear el nodo Q y regresar la búsqueda al nodo L. Si se tiene en cuenta el coste de la ruta entre Q → L (vuelta atrás) que es de 11 unidades, el coste total de la ruta se aumenta de 64 a 75 unidades.

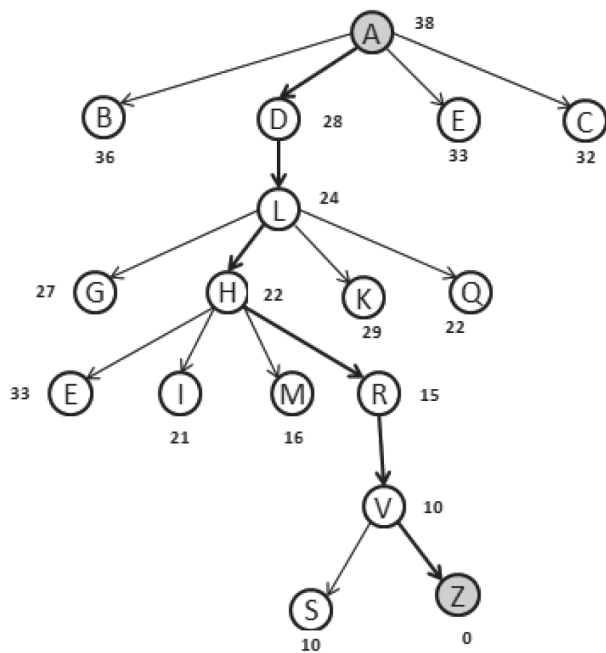


Figura 6

Árbol de búsqueda de ruta basada en la menor distancia en línea recta

Tabla 3

Otro recorrido desde nodo A hasta nodo Z

Nodo actual	Nodos hijos	Distancia	DLR	Observ.
A	B	12	36	
	C	10	32	
	D	16	28	Menor
	E	9	33	
D	L	26	24	Menor
L	G	32	27	
	H	37	22	Menor
	K	35	29	
	Q	37	22	Menor
Q	K	47	29	Menor-aumenta
L	G	32	27	
	H	37	22	Menor
	K	35	29	
H	E	33	33	
	I	31	21	
	M	29	16	
	R	42	15	Menor
R	V	50	10	Menor
V	Z	64	00	Objetivo

Dada una situación real sobre un mapa vial, las rutas se pueden clasificar entre:

- Ruta de vía nacional
- Ruta de vía pavimentada, y
- Ruta de vía destapada o semipavimentada.

En el árbol de la figura 5, se observa que existen los tres tipos de vías o conexiones, de acuerdo al grosor del arco entre dos nodos. El siguiente ejemplo muestra una búsqueda voraz que trata de localizar la mejor ruta a seguir para llegar al nodo destino. Igualmente saliendo del nodo A se debe llegar al nodo Z ubicando la mejor ruta posible, como se puede ver en la tabla 4 y en el árbol de búsqueda de la figura 7. El procedimiento a seguir consiste en:

- Ubicar los nodos hijos de A que sean tipo ruta nacional. En caso de no tener hijos con este tipo de conexión, buscar los hijos con conexión tipo vía pavimentada y en caso de no tenerlos, ubicar los hijos con tipo de conexión destapada.
- Una vez ubicados sus hijos con alguna de las tres conexiones, se selecciona aquel cuya distancia en línea recta hasta el nodo Z sea la menor. En este estado, este hijo se convierte en el nodo actual y se procede a buscar sus hijos de acuerdo con lo expuesto anteriormente.
- Continuar con la búsqueda hasta llegar al nodo objetivo.

Tabla 4

Recorrido desde nodo A hasta nodo Z buscando la mejor ruta

Nodo actual	Nodos hijos	Distancia	DLR	Observ.
A	B	12	36	
	C	10	32	Menor
	E	9	33	
C	I	23	21	Menor
I	H	32	22	Menor-aumenta
	F	34	30	
I	M	37	16	Menor
	N	33	17	

Nodo actual	Nodos hijos	Distancia	DLR	Observ.
M	H	44	22	
	S	45	10	Menor
S	V	55	10	Menor
V	Z	69	00	Objetivo

La ruta encontrada es $A \rightarrow C \rightarrow I \rightarrow H \rightarrow M \rightarrow S \rightarrow V \rightarrow Z$ con un coste de 69 unidades, mayor al coste anterior, pero garantiza que es la mejor ruta en cuanto al tipo de vía a seguir. Esto quiere decir, que la ruta más corta entre dos puntos no es siempre la mejor ruta a seguir, y que la mejor ruta entre dos puntos no siempre es la más corta.

La figura 7 es el árbol de búsqueda para este ejemplo, en el cual se observa que al seleccionar los hijos del nodo I con tipo ruta nacional, los dos seleccionados (H y F) tienen una función de evaluación que es mayor a la que tiene el nodo padre (I). De acuerdo con el método a seguir, se buscan los hijos del nodo I con tipo ruta pavimentada y se continúa con el proceso.

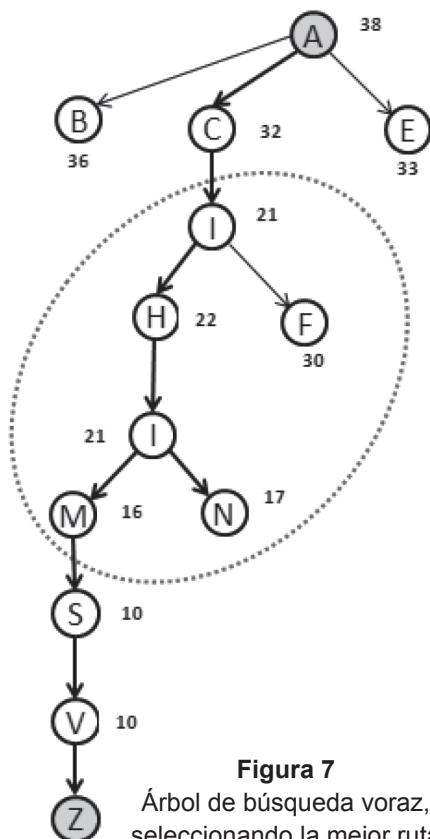


Figura 7
Árbol de búsqueda voraz, seleccionando la mejor ruta

BÚSQUEDA A ESTRELLA (A*)

La función de evaluación (heurística) utilizada se compone de dos partes.

$f(n) = g(n) + h(n)$ denominada potencia heurística, donde $g(n)$ es el coste para llegar al nodo n desde el nodo inicial. En otras palabras, es la distancia terrestre acumulada desde el nodo inicial hasta un nodo n actual.

$h(n)$ es el coste estimado para llegar al nodo objetivo desde el nodo n . Este coste estimado, no es otro valor que la distancia en línea recta (DLR) desde el nodo actual hasta el nodo objetivo, la cual se calcula usando la fórmula del haversine.

$f(n)$: coste total estimado del camino para llegar al nodo objetivo a través del nodo n . Es decir, la suma de las dos distancias: una $g(n)$ -real y, la otra, $h(n)$ -heurística.

Para conocer el valor de $g(n)$ se necesita conocer la distancia en kms entre nodos adyacentes; esta distancia se encuentra almacenada en una tabla de la Base de Datos creada para el desarrollo del software.

El coste estimado $h(n)$ se calcula conociendo las coordenadas geográficas -latitud y longitud- del nodo n (actual) y las del nodo objetivo, y aplicando la fórmula del Haversine. Este coste, similar al método de búsqueda voraz es $h_{DLR}(n) = \text{distancia en línea recta desde } n \text{ al nodo solución}$.

Lo anterior significa que la función de evaluación $f(n)$ es la suma de un coste real y conocido (*distancia en kms.*) más un valor estimado (*distancia en línea recta*).

Ya que en la Base de Datos se dispone de las coordenadas de cada una de las ciudades y, para cada trayecto entre dos ciudades adyacentes, del número de kilómetros que las separa, se

puede obtener la ruta para ir entre dos nodos (ciudades) cualesquiera dentro del mapa vial nacional utilizando este método de búsqueda.

Para el proceso de buscar una ruta entre dos nodos, estando ubicados en uno cualquiera, se debe decidir cuál es el siguiente paso a dar, es decir, cuál es el siguiente nodo (ciudad) a la que se debe pasar. Se pueden utilizar varios criterios:

- Elegir uno cualquiera, sin tener en cuenta el conocimiento adicional del que se dispone (coordenadas, costes). Esto es lo que hacen las búsquedas no informadas en anchura o profundidad. No se garantiza obtener un camino solución óptimo ni tampoco que la búsqueda esté dirigida por alguna otra consideración.
- Tener en cuenta la estimación (heurística) de la distancia en línea recta al nodo objetivo. Esto es lo que hace la búsqueda voraz primero el mejor. En estas aproximaciones no se considera la distancia previamente recorrida (es decir, la infor-

mación acerca del coste de las distancias entre nodos). La búsqueda estará dirigida por cierto conocimiento específico, aunque no totalmente seguro, pero no se garantiza el encontrar una ruta óptima.

- Ir calculando el coste de las rutas por las que se va pasando, eligiendo en cada paso el menos costoso. Pueden asegurar encontrar una solución óptima, pero, al no tener en cuenta el conocimiento heurístico (DLR), pueden resultar tan ineficaces como la búsqueda en anchura.

Para este método de búsqueda se integran los criterios (b) y (c) anteriores, para obtener una estrategia que encuentre una solución óptima, y de un modo más eficaz que la búsqueda voraz. En general, este es el mecanismo a utilizar con la búsqueda A*.

Como ejemplo, se utiliza el grafo de la figura 5 para realizar la búsqueda entre los nodos A (inicio) y Z (destino), cuyo resultado se muestra en la tabla 5.

Tabla 5
Búsqueda A* entre A y Z

Nodo actual	Nodos hijos	Distancia	DLR	$g(n)$	$h(n)$	$f(n)$	Observ.
A	B	12	36	12	36	48	
	D	16	28	16	28	44	
	E	9	33	9	33	42	Menor
	C	10	32	10	32	42	
E	G	8	27	17	27	44	
	H	11	22	20	22	42	Menor
	C	10	32	19	32	51	
H	L	11	24	31	24	55	
	R	20	15	40	15	55	
	M	7	16	27	16	43	Menor
	I	9	21	29	21	50	
M	S	8	10	35	10	45	Menor
	I	14	21	41	21	62	
S	V	10	10	45	10	55	
	W	5	8	40	8	48	Menor
W	T	8	10	48	10	58	Menor
T	Z	12	0	60	0	60	Objetivo

La ruta encontrada es $A \rightarrow E \rightarrow H \rightarrow M \rightarrow S \rightarrow W \rightarrow T \rightarrow Z$, con un coste de 60 unidades, menor a la encontrada con el método de búsqueda voraz, la cual fue de 64 unidades. La figura 8 muestra el árbol de búsqueda para este ejemplo.

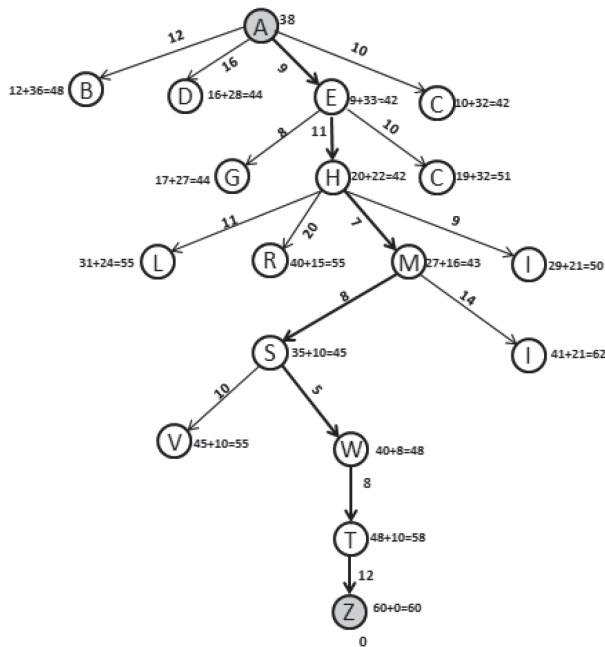


Figura 8
Árbol de búsqueda A* entre A y Z

Un segundo ejemplo utilizando el mismo grafo presentado en la figura 5 y los mismos nodos

de inicio (A) y objetivo (Z), pero seleccionando la mejor ruta, se muestra en la tabla 6. La ruta encontrada es $A \rightarrow E \rightarrow H \rightarrow M \rightarrow S \rightarrow V \rightarrow Z$, con un coste de 59 unidades, menor a la encontrada con el ejemplo anterior (60). La figura 9 muestra el árbol de búsqueda para este ejemplo.

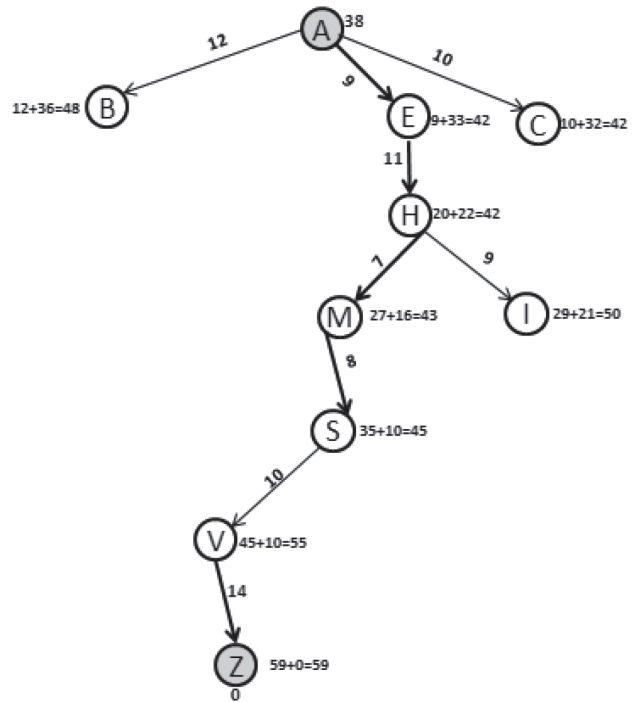


Figura 9
Árbol de búsqueda A*, mejor ruta

Tabla 6
Búsqueda A* entre A y Z, seleccionando la mejor ruta.

Nodo actual	Nodos hijos	Distancia	DLR	$g(n)$	$h(n)$	$f(n)$	Observ.
A	B	12	36	12	36	48	
	E	9	33	9	33	42	Menor
	C	10	32	10	32	42	
E	H	11	22	20	22	42	Menor
H	M	7	16	27	16	43	Menor
	I	9	21	29	21	50	
M	S	8	10	35	10	45	Menor
S	V	10	10	45	10	55	
V	Z	14	0	59	0	59	Objetivo

A partir de los ejemplos, para entender el funcionamiento del método de búsqueda voraz y el método de búsqueda A*, se utilizó el software desarrollado tanto para la web, como para aplicación local⁵.

CONCLUSIONES

Las técnicas de búsqueda que facilitan los enfoques de inteligencia artificial permiten tanto en tiempo y espacio hallar solución a diferentes problemas, que si se emplean las técnicas operacionales de ruta crítica u óptima, el tiempo de computación es amplio y puede, en general, no

llegar a la mejor solución; que en el peor de los casos, se queda en un ciclo que no permite que el proceso termine, además, lo fundamental, es que en estos procesos la información debe estar completa, mientras que para las técnicas heurísticas, algo podría faltar y no inhibe que el proceso halle una solución.

La búsqueda informada, garantiza encontrar una solución al problema, mediante el uso de alguna heurística; si bien es cierto, la solución encontrada no siempre es la mejor, si se puede estar seguro que en la mayoría de las veces, el resultado obtenido se acerca a la mejor solución.

BIBLIOGRAFÍA

- García, A. (2012). *Inteligencia Artificial: fundamentos, práctica y aplicaciones*. México D.F.: Alfaomega.
- Mapas de Ruta. Instituto Geográfico Agustín Codazzi - IGAC - 2013.
- Michalewics, Zbigniew (1996). *Genetic algorithms + data structures = evolution programs*. 3a-ed. Berlin, Springer.
- Ponce, P. (2011). *Inteligencia artificial con aplicaciones a la ingeniería*. México D.F: Alfaomega.
- Stuart, R. y Norvig P. (2008). *Inteligencia artificial - un enfoque moderno*. New York: Prentice Hall.
- Torres, L. C. (2007). *Inteligencia artificial, conceptos básicos*. Bogotá D.C.: Universidad Nacional.
- Torres, L.C. y Garzón N.M. (2012). *Inteligencia artificial - una aproximación*. Berlín: Editorial Académica Española.

Sitios web usados para generar la información del ejemplo realizado en el documento

<http://co.lasdistancias.com/colombia> Distancias en línea recta entre municipios.

<http://carta-natal.es/ciudades/colombia> Coordenadas geográficas municipios Colombia
DANE, Mapas viales de Colombia.

http://www.invias.gov.co/images/mapas/imagenes_red_vial/
DANE, División político - administrativa de Colombia.

www.dane.gov.co/index.php?option=com_content&view=article&id=378&Itemid=93.

⁵ La aplicación web se encuentra disponible en www.RutaFacil.co, en el cual existen las opciones para buscar una ruta usando el método voraz y el método A*, la mejor ruta, la ruta entre capitales, entre otras. El aplicativo local se encuentra disponible en www.RutaFacil.co/descargas/IA,

el cual se puede descargar e instalar en sistemas operativos Windows XP y Windows 7. Dentro de los archivos que componen el aplicativo, se encuentra un archivo llamado Readme.txt en donde se encuentran las instrucciones a tener en cuenta para su instalación y normal funcionamiento.