

Búsqueda de Rutas Implementando A* en un Entorno Virtual 2D.

Gonzalo Hernández Hernández¹, Víctor Tomás Tomás Mariano¹

¹Universidad Autónoma del Estado de Hidalgo - Escuela Superior Huejutla.
elektrostation_ferrobuss89@hotmail.com, victor_tomasm@hotmail.com.

RESUMEN

En el presente trabajo se muestra el avance de un proyecto de tesis en proceso, el cual consiste en la búsqueda de una ruta óptima en un entorno virtual, caso de estudio laberinto ampliado, mediante la implementación del algoritmo A Star, el cual utiliza dos listas que controlan la secuencia de exploración en el laberinto ampliado y de una medida heurística que lo convierte en un método rápido y óptimo. Las pruebas son realizadas mediante flujos de entrada/salida con archivos de texto cuyos datos son tratados internamente en la aplicación como matrices binarias sobre las que A* opera en los 8 vecinos de cada celda.

Palabras Clave: Planeación de trayectorias, Laberinto Ampliado, A Star.

1 INTRODUCCIÓN.

La planeación de trayectorias *-path planning-* tiene varias aplicaciones, entre estas están, la búsqueda de rutas con robots autónomos (Dracopoulos, 1998), (Ersson & Hu, 2001) y (Jan & Chang K-Y, 2003) auxiliados con sensores que permiten obtener información local del espacio de trabajo y en base a esta información realizar movimientos. Tal es el caso de (Thrun, Gordon, & Linkhachev, 2003) donde el método propuesto es implementado a los movimientos que un brazo robótico debe realizar para alcanzar un objetivo y que puede verse obstaculizado por otros objetos. El aporte principal de dicho método consiste en una mejora que resuelve las inconsistencias locales generadas por A* mediante un coeficiente de decremento variable denominado *límite de sub-optimalidad* cuyo valor mínimo por alcanzar es 1, ya que una vez llegado a este valor, las exploraciones en la búsqueda de trayectoria terminan.

(Stentz, 1998) propone un algoritmo semejante a A*, bajo el nombre de D*, el cual encuentra la ruta óptima del problema representándolo mediante un grafo dirigido, donde los arcos representan los

costos, los cuales pueden ser valores continuos y pueden cambiar durante el proceso de búsqueda.

La búsqueda de trayectorias puede ser tratada también mediante niveles jerárquicos, tal como se hace en (Lanctot, Man Sun, & Verbrugge, 2006). En el nivel más alto, el espacio de búsqueda se divide en zonas implementando una técnica de aceleración que bloquea la última zona que se abandonó mientras la zona actual es analizada. En el nivel más bajo, cada zona es descompuesta en rejillas, en la que se aplica A* para encontrar un recorrido mínimo que lleve desde la entrada de esa zona a la salida de la misma y así entrar a otra hasta alcanzar el punto establecido como objetivo.

1.1 Descripción del Problema.

El problema de búsqueda de rutas, también conocido como *maze problem*, se ha discutido en varios trabajos de investigación (Dracopoulos, 1998), (Jan & Chang K-Y, 2003) y (Lanctot, Man Sun, & Verbrugge, 2006), para el caso de estudio en particular, se tiene un laberinto con múltiples vías o laberinto ampliado (Tomás, 2007), (Tomás, Pozas, & Hernández, 2011), (Tomás & Hernández, 2011), en el que, dependiendo de las dimensiones y complejidad, el número de rutas posibles a encontrar dentro del laberinto se incrementa, ver fig. 1.

Un factor importante de las rutas posibles, depende de la ampliación k que tiene el laberinto, además, una vez abandonado el punto de entrada, el número de combinaciones posibles en los movimientos puede llegar a ser muy grande, lo que provocará que el explorador visite una mayor cantidad de posiciones en el laberinto, ya que por cada celda expandida pueden generarse otras más según la cantidad de vecinos transitables que ésta tenga, tal como se muestra en fig. 1, por lo que es

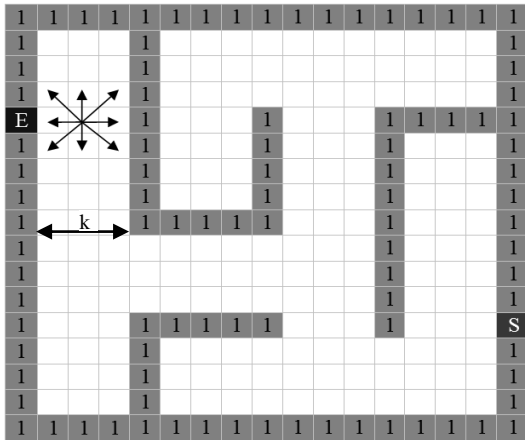


Fig. 1. Laberinto Ampliado con tres pistas ($k=3$).

necesario de algún modo “recordar” que secciones del laberinto ya han sido exploradas, y así, tomar la *mejor decisión de movimiento en el espacio de búsqueda*, debido a que se desea encontrar una ruta solución óptima por la cual transitar desde el punto marcado como entrada a otro punto distinto del primero marcado como salida.

Al intentar minimizar el costo de la trayectoria, el proceso de búsqueda puede incrementar el tiempo de ejecución si el análisis de las celdas toma dirección hacia un obstáculo que le impide alcanzar directamente el objetivo, lo que obligará retroceder para buscar un camino alternativo al que se planeaba conseguir.

1.2 Antecedentes.

Existe un trabajo previo del cual se desprende esta investigación en el que se realiza la propuesta para encontrar una solución en laberintos ampliados (Tomas, 2007). Para ello se hace uso de un laberinto A de una sola pista con dimensión $m \times n$, que conserva la misma estructura del laberinto ampliado B de $p \times q$. La búsqueda de la solución se realiza en el laberinto A, mediante el algoritmo de Lee (Lee, 1961), en el que se lleva a cabo la etiquetación de las celdas cuyo valor representa la distancia existente respecto a la entrada del laberinto, y mediante un proceso de mapeo de coordenadas, se visualiza una solución en el laberinto ampliado B. Sin embargo, la búsqueda de la solución no se realiza en el laberinto ampliado y esta dista mucho de ser la más corta, además de poseer celdas solución con movimientos solo en horizontal y vertical.

2 DESARROLLO DEL PROCESO DE INVESTIGACIÓN.

Una de las bases para iniciar la familiarización del problema, fue tener una investigación y resultados previos. La ventaja de visualizar un laberinto como un grafo, permitió retomar el “comportamiento” de los algoritmos principales: *Depth First Search* (DFS), *Breadth First Search* (BFS), Dijkstra (Johnsonbaugh, 1999) y su posible implementación para encontrar una solución, de los que, el segundo y el último son óptimos pero el espacio en memoria necesario puede volverse crítico, DFS y BFS (Russell & Norvig, 2004), mientras que el primero de ellos no promete una trayectoria con costo mínimo, la ventaja de modelar el problema con grafos es que los costos de desplazamiento pueden variar de un punto a otro y no necesariamente requiere hacer una distinción entre pasillos y muros, ya que los arcos son los únicos medios que enlazan los distintos puntos, sin embargo su representación en memoria resulta ser más compleja que una representación en rejillas.

Otros algoritmos permiten modelar el problema en entornos tipo rejilla, y utilizar una matriz A de $m \times n$, en los que se requiere procesar datos dentro de una matriz binaria, ceros y unos, con la finalidad de identificar áreas libres y obstáculos en el laberinto. Entre los algoritmos analizados están principalmente: *Algoritmo de Lee* (Lee, 1961), *Algoritmo de Hadlock* (Sait & Youssef, 1999), *Algoritmo de Soukup* (Soukup, 1978), *Algoritmo A** (Hart, Nilsson, & Raphael, 1964), siendo este último el elegido para ser implementado ya que la cantidad de nodos que explora es menor que cualquiera de los anteriores, y por garantizar la ruta más corta.

En los algoritmos aplicados a *maze problem* o *maze route*, se decidió por el Algoritmo de A* debido a que también controla mediante dos listas aquellas posiciones que son candidatas a generar la ruta óptima y otra en la que definitivamente se colocan aquellas que no necesitan ser visitadas de nuevo, incorporando una función heurística que a comparación de las anteriores no lo hacen, lo que le permite acercarse más rápido al objetivo, en el que se aceptan movimientos en horizontal, vertical y diagonal.

2.1 Fundamentos de A*

Es un algoritmo de búsqueda “informada” que busca el camino más corto desde un punto inicial a la meta, usando una heurística óptima. La

heurística consiste en elegir una función $h(n)$ que eficiente el costo por alcanzar la meta partiendo desde la ubicación n . Si $h(n)$ es menor o igual a la real -que suele ser el camino recto hacia la meta- se utiliza una función $f(n)$ como evaluadora:

$$f(n) = g(n) + h(n) \quad (1)$$

Donde $g(n)$ es el costo acumulado del movimiento por ir desde el punto inicial hasta la celda n .

2.1.1 Pseudocódigo A*

- 1) Para la *entrada* del laberinto: calcular $f(n)$, $g(n)$ y $h(n)$, e insertar a Lista Abierta (LA).
- 2) Repetir mientras LA contenga elementos
 - i. Extraer en LA la celda con menor costo $f(n)$ (llamémosle *celda analizada*), y se inserta a Lista Cerrada (LC).
 - ii. Para cada *celda vecina* transitable a la *celda analizada*.
 1. Si *celda vecina* no se encuentra en LC ni en LA, hacer que su *celda padre* sea la *celda analizada*, calcular los costos $f(n)$, $g(n)$ y $h(n)$, e insertarla a LA. Si sucede que *celda vecina* es la *salida* del laberinto, el camino se encuentra al retroceder en dirección de los *padres* hasta alcanzar la celda de la que inicialmente se partió, y el proceso de búsqueda termina.
 2. En otro caso, si ya se encuentra en LA, verificar si el trayecto a través de *celda analizada* es menor que el que ya tiene en su costo $g(n)$. Si es así, cambiar su *padre* a *celda actual* y recalculer sus costos $g(n)$ y $f(n)$.
- 3) En caso de que LA se encuentre vacía y no se ha llegado a la *salida* del laberinto, entonces no existe camino posible.

2.2 Funcionamiento del algoritmo A*.

Existen múltiples heurísticas que pueden ser utilizadas (Patel, 2011), en el presente trabajo se hace uso de la heurística Manhattan. Para ello, se define la función heurística para la celda n con coordenadas en (x_n, y_n) , donde (x_{nf}, y_{nf}) corresponde a la ubicación de la meta y W el costo del movimiento, quedando como:

$$h(n) = (|x_{nf} - x_n| + |y_{nf} - y_n|) * W \quad (2)$$

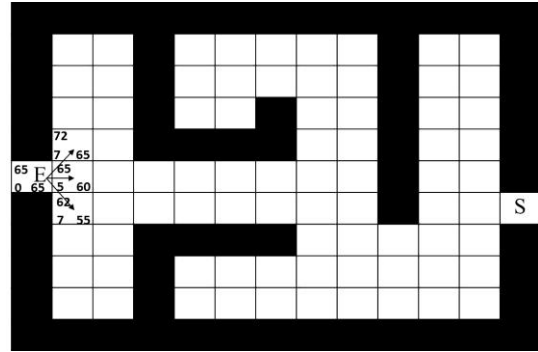


Fig. 2. Cálculo inicial de costos por cada vecino transitable.

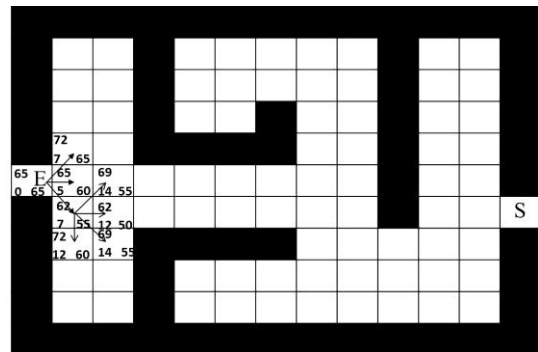


Fig. 3. Exploración de los vecinos transitables después de haber realizado el primer desplazamiento.

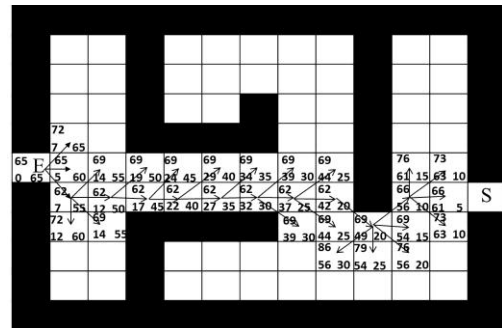


Fig. 4. Celdas exploradas a un paso antes de encontrar la salida de laberinto

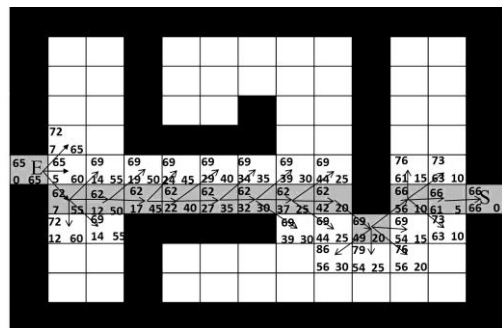


Fig. 5. Trayectoria óptima que enlaza la entrada y la salida del laberinto utilizando A*.

A modo de ejemplo, se utilizará un laberinto de ampliación $k=2$, con movimientos en horizontal, vertical y diagonal, donde éste último genera un

costo más alto que los dos primeros, con fundamento en el teorema de Pitágoras. En el que para un movimiento en horizontal o vertical se asigna una magnitud de 5 unidades, generando entonces que el costo en diagonal se aproxime a 7 unidades, los cuales son tomados para trazar la ruta más corta en el laberinto, donde la celda E representa el punto de partida, la celda S el punto objetivo o meta, las celdas libres representan caminos posibles y las celdas de color negro representan los obstáculos.

Paso 1: Partiendo desde E, se calculan sus costos para el que $g(E)= 0$, $h(E)= 65$ y $f(E)=65$. Se inserta a LA.

Paso 2: Siendo E el único y por consecuente el de menor costo f en LA, se extrae para calcular los costos f , g y h por cada vecino transitable, como puede verse en fig. 2. Procediendo a insertar los vecinos a LA y la celda analizada (E) dentro de LC.

Paso 3: Continuando con la búsqueda, se extrae aquella celda contenida en LA cuyo costo f sea mínimo, siendo la celda con costo $f=67$ la nueva celda analizada, y se calculan al igual que el paso anterior los costos para cada uno de los vecinos que éste tenga, véase fig. 3. En este paso, es importante notar que dos de sus vecinos transitables (vecino noroeste y vecino norte) se mantienen intactos, esto debido a que los elementos contenidos en LC se ignoran y no vuelven a ser recalculados, así como, los costos de un vecino contenido en LA no pueden ser modificados a menos que la trayectoria que pasa por la celda analizada sea más corta a la que ya se tiene.

El proceso continúa, donde por cada celda extraída en LA genera que más elementos entren a ella, esto siempre que la celda extraída no represente un punto muerto en el laberinto. En la fig. 4, puede verse el cálculo completo que se obtiene a un paso antes de llegar a la salida y notar que los nodos analizados para este laberinto se encuentran a un paso de la solución óptima.

A* establece que, cuando existe más de una celda con costo mínimo $f(n)$ en LA, la elección debe ser de forma arbitraria (Hart, Nilsson, & Raphael, 1964), sin embargo, con la finalidad de minimizar el tiempo de búsqueda, en el presente trabajo se ha establecido un procedimiento adicional para este caso, el cual consiste en extraer aquel que tenga el costo heurístico más bajo.

Una vez que se ha llegado a la salida, la trayectoria buscada se genera en dirección inversa para acceder a la celda padre de cada posición hasta llegar a la ubicación definida como entrada del laberinto, tal como se muestra en fig. 5, del que se observa un recorrido óptimo con costo $f(n)$ en la salida del laberinto.

3 RESULTADOS Y ANÁLISIS.

En las pruebas realizadas se contempló el uso de varios laberintos con dimensiones diferentes, en los que el algoritmo A*, arrojó excelentes resultados.

```

1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1
1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1
1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1
1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1
E 0 0 0 1 0 0 0 1 0 0 0 1 1 1 1 1 1
1 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1
1 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1
1 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1
1 0 0 0 1 1 1 1 1 0 0 0 1 0 0 0 1
1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1
1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1
1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1
1 0 0 0 1 1 1 1 1 0 0 0 1 0 0 0 S
1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1
1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1
1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

```

a)

```

1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1
1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1
1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1
* 0 0 0 1 0 0 0 1 0 0 0 1 1 1 1 1 1
1 * 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1
1 0 * 0 1 0 0 0 1 0 0 0 1 0 0 0 1
1 0 0 * 1 0 0 0 1 0 0 0 1 0 0 0 1
1 0 0 * 1 1 1 1 1 0 0 0 1 0 0 0 1
1 0 0 0 * 0 0 0 0 0 0 0 1 0 0 0 1
1 0 0 0 0 * 0 0 0 0 0 0 1 0 0 0 1
1 0 0 0 0 0 * * * 0 0 0 1 0 0 0 1
1 0 0 0 1 1 1 1 1 * * * 1 * * * *
1 0 0 0 1 0 0 0 0 0 0 0 * 0 0 0 1
1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1
1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

```

b)

Fig. 6. Laberinto ampliado ($k=3$). a) Laberinto inicial, b) Laberinto con Ruta Solución.

En la fig. 6 a), se muestra un laberinto con tres vías, como entrada al sistema, donde los 1's representan los muros del laberinto y los 0's las celdas por las que es posible trasladarse. Después de aplicar el algoritmo A*, se visualiza la ruta indicando con asteriscos (“*”) las celdas solución del laberinto, ver fig. 6 b).

Existen laberintos en los que no es posible encontrar una ruta solución, debido a que los puntos elegidos como entrada y punto objetivo son inalcanzables, en la fig. 7, se observa que no existe manera para llegar a la salida, debido a que esta celda está completamente aislada.

```

1 1 1 1 1 1 1 1 1 1 S
1 0 0 0 1 0 0 0 1 1
1 0 1 1 0 0 1 1 0 1
1 0 1 0 1 1 0 1 0 1
1 0 1 0 0 0 0 0 0 1
1 0 1 0 0 0 0 0 0 1
1 0 0 0 1 1 1 0 1 1
E 1 1 1 1 1 1 1 1 1 1

```

Fig. 7. Laberinto sin solución.

```

1 1 1 1 1 1 1 1 1 1 * 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 0 0 0 0 1 0 0 1 0 * * * * * * * * * * * * * * * 0 0 1
1 0 0 0 0 1 0 0 1 0 0 0 0 1 1 1 1 1 0 0 1 1 1 * 0 1
1 0 0 1 1 1 0 0 1 0 1 0 0 1 0 0 0 0 0 0 0 0 1 * 0 1
1 0 0 1 0 0 0 1 1 1 1 1 1 1 1 1 1 1 0 0 1 0 1 * 0 1
1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 1 1 1 1 1 * 0 1
1 0 0 1 1 1 0 0 1 1 1 1 0 1 0 0 0 0 0 1 0 1 * 0 1
1 0 0 0 0 1 0 0 0 0 0 1 0 1 0 0 0 0 0 1 0 1 * 0 1
1 1 1 0 0 1 1 1 1 1 1 1 1 1 0 0 1 1 1 1 0 1 * 0 1
1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 * 0 1
1 1 1 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 1 1 1 * 0 1
1 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 0 0 0 0 1 * 0 1
1 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 1 * 0 1
1 0 0 1 0 1 1 1 1 0 0 1 0 0 1 0 0 0 0 1 1 1 * 0 1
1 0 0 1 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 1 0 0 * 0 1
1 0 0 1 0 0 0 0 1 1 1 0 0 0 1 0 0 1 1 1 1 1 * 0 1
1 1 1 1 1 1 0 0 1 0 1 0 1 1 1 0 0 0 0 1 0 * 0 0 1
1 0 0 0 0 1 0 0 1 0 1 0 0 0 1 0 0 0 0 1 * 0 0 0 1
1 0 0 0 0 1 0 0 1 0 0 0 0 0 1 0 0 1 1 1 * 0 1 1 1
1 0 0 0 0 1 0 0 1 0 0 0 1 1 1 0 0 0 0 * 0 0 0 0 1
1 0 0 1 1 1 1 1 1 0 0 0 0 0 1 0 0 * * 0 0 0 0 0 1
1 0 0 1 0 1 0 0 1 0 0 0 1 1 1 0 * 1 1 1 0 0 1 1 1
1 0 0 1 0 1 0 0 1 0 0 0 1 0 0 * 0 1 0 0 0 0 0 0 1
1 0 0 0 0 0 0 0 0 0 0 0 1 0 * 0 0 1 0 0 0 0 0 0 1
1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 1 1 0 0 1 0 1
1 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 1 0 0 1 0 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

```

Fig. 8. Laberinto con punto de salida en una sección interna del laberinto.

En la fig. 8, se muestra una ruta solución cuyo punto objetivo se ubica en el interior del laberinto, casos como éste son comunes en concursos de robótica.

En la ruta solución obtenida, se observa que se realizan movimientos en *diagonal*, esta es una mejora respecto al trabajo previo del cual se desprende esta investigación, debido a que se tiene la ventaja de encontrar una ruta y se garantiza que es la más corta.

4 CONCLUSIONES.

En este documento se implementó A* sobre laberintos ampliados, el cual arrojó excelentes resultados al generar en todos los casos que la trayectoria obtenida fue la menos costosa, para el que la heurística utilizada fue admisible.

Cabe resaltar que la manera en que se represente el laberinto facilitará el procedimiento para obtener la solución.

4.1 Trabajo Futuro

Otra parte fundamental del caso de estudio, se desea comparar la ruta encontrada con A* y la ruta que realiza el usuario -explorador- al recorrer el laberinto, con la finalidad de conocer el grado de error cometido, una propuesta de solución que se está analizando es mediante la correlación de imágenes, debido a que se puede obtener dos imágenes de insumo al final de una prueba, una imagen en donde se indique la solución y otra imagen donde se visualice el recorrido realizado. El grado de error se puede calcular con el uso de una medida de distancia o disimilaridad entre vectores o matrices y ésta es la segunda parte en la que actualmente se está trabajando.

REFERENCIAS

Dracopoulos, D. (1998). Robot Path Planning for Maze Navigation. Brunel University, Department of Computer Science, IEEE.

Ersson, T., & Hu, X. (2001). Path Planning and Navigation of Mobile Robots in Unknown Environments. Royal Institute of Technology, Stockholm Sweden.

Hart, P. E., Nilsson, N. J., & Raphael, B. (1964). A Formal Basis for the Heuristic Determination of Minimum Cost Paths. *IEEE Transactions on Systems Science and Cybernetics*, SSC-4 (2).

Jan, G. E., & Chang K-Y, P. I. (2003). A New Maze Routing Approach for Path Planning of

- Mobile Robot. *IEEE International Conference on Advanced Intelligent Mechatronics*.
- Johnsonbaugh, R. (1999). Teoría de Gráficas: Un algoritmo para la ruta más corta. En *Matemáticas Discretas*. México: Prentice Hall.
- Lanctot, M., Man Sun, N. N., & Verbrugge, C. (2006). Path Finding for Large Scale Multiplayer Computer Games. School of Computer Science, McGill University, Montréal, Canada.
- Lee, C. Y. (1961). An Algorithm for Path Connections and its Applications. *IRE Transactions on Electronic Computers*, EC-10.
- Patel, A. (2011). *Heuristics: Stanford Theory Group*. Recuperado el 17 de Noviembre de 2011, de <http://theory.stanford.edu/~amitp/GameProgramming/Heuristics.html>
- Rusell, S. J., & Norvig, P. (2004). Inteligencia Artificial. Un Enfoque Moderno. Madrid: Pearson Education, S.A.
- Sait, S. M., & Youssef, H. (1999). En *VLSI Physical Design Automation: Theory and Practice* (págs. 233-276). Singapore: World Scientific Publishing Co.
- Soukop, J. (1978). Fast Maze Router. *DAC '78 Proceedings of the 15th Design Automation Conference*. Ottawa, Ontario.
- Stentz, A. (1998). Optimal and Efficient Path Planning for Partially-Known Enviroments. Pittsburgh, PA.
- Thrun, S., Gordon, G., & Linkhachev, M. (2003). ARA*: Anytime with Provable Bounds on Sub-Optimality. School of Computer Science, Carnegie Mellon University, Pittsburgh, PA.
- Tomás, V. T. (2007). Generación y Ampliación de Laberintos. Pachuca de Soto, CITIS, Hidalgo, México.
- Tomás, V. T., & Hernández, J. (2011). Algoritmo para la Generación de Laberintos de Conexión Múltiple en 2D. *Vigesimosegunda Reunión Internacional de Otoño, de Comunicaciones, Computación, Electrónica, Automatización, Robótica y Exposición Industrial. La Conjunción de las Tecnologías Digitales en las Redes Inteligentes. IEEE Sección México*. Acapulco, Guerrero, México.
- Tomás, V. T., Pozas, M., & Hernández, J. (2011). Propuesta para la generación de laberintos ampliados en 2D. *Simposio Iberoamericano Multidisciplinario de Ciencias e Ingeniería*. Zempoala, Hidalgo.