

Análisis de algoritmos de búsqueda en espacio de estados.

Grafos y dígrafos.

Los grafos son estructuras discretas que constan de vértices y aristas que conectan estos vértices. Un grafo $G = \{V, E\}$ está formado por un conjunto de vértices, V , y un conjunto de aristas E . Cada arista es un par (u, v) donde $u, v \in V$. Algunas veces las aristas tienen una tercera componente, denominada peso o costo. Las aristas de un grafo pueden ser dirigidas o no dirigidas. Una arista (u, v) es dirigida de u a v si el par ordenado (u, v) es ordenado y u precede a v . Mientras que es no dirigida si es par (u, v) no es ordenado.

Los grafos suelen visualizarse trazando los vértices en forma de óvalos o rectángulos y las aristas como segmentos o curvas que unen pares de vértices. Si todas las aristas de un grafo son no dirigidas, es un grafo no dirigido; en el caso que todas sus aristas sean dirigidas es un grafo dirigido. Si el grafo tiene aristas dirigidas y no dirigidas se le llama grafo mezclado.

- **Adyacencia.** Dos vértices son adyacentes si están conectados por una arista. Los vértices adyacentes de un vértice se les llaman vecinos.
- **Caminos.** Es una secuencia de vértices w_1, w_2, \dots, w_n tal que $(w_i, w_{i+1}) \in E$ para $1 \leq i \leq n$. Donde n es el número de aristas en el camino, es decir, la longitud. Un grafo es conectado si hay al menos un camino desde cualquier vértice a otro vértice cualquiera. Si no existe un camino entre todos los vértices, es un grafo no conectado.

Un laberinto puede ser representado por un grafo, donde los pasillos del laberinto son las aristas y sus intersecciones los vértices del grafo. Esta representación nos brinda la posibilidad de analizar sólo la estructura del grafo; para aplicar la teoría de grafos, es necesario conocer la estructura del grafo e incluso tomar en cuenta ciertas propiedades para poder decidir que algoritmos son los más adecuados para su implementación. Saber estas propiedades nos permite aplicar el recorrido de grados para encontrar la conexión entre dos nodos. Entre los algoritmos que se pueden aplicar están: *primera búsqueda en amplitud (BFS)*, *primera búsqueda en profundidad (DFS)*, *Dijkstra* y *S-Star*.

Algoritmos de búsqueda.

Es posible encontrar la ruta con el menor costo posible entre dos puntos dados. Debido a que en un laberinto se puede modelar también como un arreglo bidimensional de $N \times M$, en la que hay celdas libres y celdas pared; un laberinto es un área de dos dimensiones en forma de rejilla de cualquier tamaño.

- **Algoritmo de búsqueda en amplitud.** Etiqueta todas las celdas, buscando la celda del final en todos sus vecinos adyacentes. Si no llega a la celda "T", la búsqueda continua hacia habitaciones adyacentes encontradas a partir de la habitación inicial; hasta que la celda "T" sea localizada. A continuación se muestran los pasos que sigue.

1. Etiquetar la celda de inicio como 0.
2. $i=0$.
3. Para cada celda etiquetada con i , etiquete todas las celdas adyacentes no etiquetadas con $i+1$. Si no hay celdas adyacentes, parar.
4. Si alguna de las celdas recién etiquetadas es la celda objetivo, terminar; una "ruta" solución ha sido encontrada.
5. $i=i+1$.
6. Ir al paso 3.

- **Algoritmo de búsqueda en profundidad.** Etiqueta todas las habitaciones como no visitadas, usa una pila para recordar donde debe ir cuando no alcanza un extremo muerto. Este algoritmo encuentra un camino si existe, y se puede aplicar a laberintos de conexión simple o múltiple (tiene circuitos internos), pero no garantiza dar la ruta más corta. Para empezar se elige una habitación inicial "S" y final "T" para poder seguir las siguientes reglas:

Regla 1. Si es posible, visitar una habitación (si hay más de una se elige una al azar) adyacente no visitado, marcarlo como visitado, si la celda es la objetivo, terminar.

Regla 2. Si no se puede aplicar la regla 1, entonces, si es posible, regresar a la última bifurcación.

Regla 3. Si no se puede aplicar las reglas anteriores, el proceso ha terminado

- **Algoritmo de Nayfeth.** Un laberinto se puede representar como un arreglo bidimensional con valores "1" y "0" representando paredes y pasillos. Cada celda tiene vecinos adyacentes en cuatro direcciones: *Norte, Sur, Este y Oeste*.

Para empezar se elige una celda inicial E y final S , ambas tienen "0", estas deben ser distintas. El algoritmo bloquea todos los puntos muertos en el laberinto y cada celda libre es accesible solamente en una dirección; las celdas libres se hacen nuevas celdas pared. Este procedimiento se repite hasta que el "espacio de búsqueda" queda sin cambios, es decir que las únicas celdas libres son la solución del laberinto. Este algoritmo no funciona adecuadamente para laberintos de conexión múltiple. Se basa en las siguientes reglas:

Regla 1. Las celdas pared no cambian.

Regla 2. Las celdas libres, se vuelven pared si tienen tres o más vecinos pared.

Regla 3. Las celdas libres permanecen igual si tiene menos de tres celdas pared.

- **Algoritmo A-Star.** Utiliza una búsqueda heurística para encontrar la ruta óptima entre dos puntos. Maneja tres funciones: F , G y H . Donde:

La función G es el costo del mejor camino desde la celda inicial a la celda n hasta el momento durante la búsqueda.

La función H es el costo del camino más corto desde la celda n a la celda objetivo más cercana n .

$F = G + H$, es decir, F es el costo del camino más corto desde la celda inicial a la celda objetivo.

El costo de la función H , se calcula con una distancia de Manhattan, que consiste en sumar la cantidad de bloques en horizontal y vertical que restan para llegar a la meta y multiplicar por el costo que tiene asignado este tipo de movimientos.

Generación automática de variantes de trayectorias aplicada al diseño óptimo bajo criterios múltiples de redes hidráulicas de abasto.

En la geometría computacional obtener la mejor triangulación de la nube de puntos significa que se consideran todos los segmentos posibles entre puntos. La selección adecuada da como resultado un conjunto de triángulos óptimos, entendiendo por tales, aquellos que tienen los ángulos más abiertos posibles. Uno de los métodos más utilizados para realizar las triangulación es el método **Delaunay**.

Distribución en planta de trayectorias

El problema de optimizar la distribución en planta de trayectorias está orientado a disminuir los costos de inversión y operación, y puede ser abordado desde dos puntos de vista: el primero supone una total libertad para ejecutar el trazado de la trayectoria en el espacio físico previsto; el segundo consiste en la selección de posibles alternativas de trazado sobre zonas públicas y de paso. El conjunto de alternativas suele ser discreto y el problema se limita a seleccionar la alternativa más económica durante la optimización de la distribución en planta de redes urbanas de distribución.

Un grafo \mathbf{G} es una dupla $\mathbf{G} = (\mathbf{V}, \mathbf{A})$ donde \mathbf{V} es un conjunto no vacío de vértices y \mathbf{A} es un conjunto de aristas. Cada arista es un par de (v, w) donde v, w pertenecen a \mathbf{V} . A su vez pueden ser clasificados como *grafo dirigido* y *grafo no dirigido*; en el segundo las aristas no están ordenadas, es decir; $(v, w) = (w, v)$. Un vértice w es **adyacente** a v si y solo si (v, w) pertenece a \mathbf{A} .

El camino de un vértice w_1 hacia w_q es una secuencia $w_1, w_2, \dots, w_q \in V$, tal que todas las aristas $(w_1, w_2), \dots, (w_{q-1}, w_q) \in A$. La longitud de un camino se define por un número determinado de aristas del camino (número de nodos-1); se conoce como camino simple, aquel en que todos los vértices son distintos y ciclo, aquel camino en el cual el primer y último vértice son iguales. Se denomina ciclo simple si es camino es simple.

Un grafo es conexo o conectado si existe un camino entre cualquier par de vértices y es completo si existe una arista entre cualquier par de vértices. Es etiquetado si asociamos a cada arista un peso o valor.

Representación del grafo mediante matrices de adyacencia. El conjunto de aristas es representado mediante una matriz $M[\text{vértices}, \text{vértices}]$ de boléanos, donde $M[v, w] = 1$, si y solo si (v, w) pertenece a A ; tomará

un valor nulo si no existe arista, si está etiquetado la matriz será de elementos de ese tipo.

Representación mediante listas de adyacencia. Para cada nodo de V tendremos una lista de aristas que parten de ese vértice. Estas listas están guardadas en un arreglo de vértices cabecera. En un Grafo etiquetado se añade un nuevo campo a los elementos de la lista. Si el grafo es no dirigido entonces cada arista (v, w) será representada dos veces, en la lista de v y en la de w .

Recorrido sobre grafos

Se desarrolla para visitar los vértices y las aristas de manera sistemática y puede efectuarse de las formas siguientes:

Búsqueda primero en profundida. Es equivalente a un recorrido en pre-orden de un árbol. Se elige un vértice v de partida. Se marca como visitado y se recorren los vértices no visitados adyacentes a v , usando recursivamente la búsqueda primero en profundidad.

Búsqueda primero en amplitud o anchura. Es equivalente a recorrer un árbol por niveles. Dado un vértice v , se visitan primero todos los vértices adyacentes a v , luego todos los que están a distancia 2 (y no visitados), a distancia 3, y así sucesivamente hasta recorrer todos los vértices.

Procedimiento general de preparación de decisiones

A continuación se listan los pasos generales del procedimiento:

- **Determinación del trazado de la red cerrada de mayor cantidad de ciclos.** Es decir que a partir de los vértices definidos se construye la red mallada de mayor cantidad de ciclos, garantizando la conexión de todos los vértices a partir de la mayor cantidad posible de aristas de forma tal que estos no se crucen formando una malla triangular. Para esto se aplica el método Delaunay.
- **Modificación de la red desarrollada con ayuda del sistema a criterio del usuario.** El sistema adecuará los tramos obtenidos para evitar intersecciones con elementos preestablecidos.
- **Determinación de la red mínima priorizada.** Un criterio importante para obtener menos pérdidas es minimizar la longitud de las aristas.

- **Problema del árbol generador mínimo (AGM).** Para minimizar la longitud de la red nunca será conveniente tener un ciclo ya que se puede prescindir de una de las aristas que forman parte del ciclo sin destruir la conectividad de la red y de esta manera disminuir la longitud de la misma. Para encontrar el árbol generador mínimo de un grafo es la utilización de algoritmos golosos si se tienen en cuenta las siguientes consideraciones:

- Un árbol con n vértices tiene exactamente $n-1$ aristas.
- Siempre existe un único camino entre dos vértices de un árbol.
- Agregar una arista cualquiera a un árbol crea un ciclo, eliminando cualquier arista de este ciclo se recupera el árbol.

Estos algoritmos permiten resolver problemas de optimización en forma extremadamente sencilla, en algunos problemas generan una solución óptima pero en otros lamentablemente tal cosa no es posible. Para que sea óptimo deben darse dos condiciones: el problema debe poder dividirse en sub-problemas, y debe ser posible demostrar que existe al menos una solución óptima que comienza con la decisión golosa.

- **Implementación del Algoritmo Kruskal para determinar el árbol de expansión de costo mínimo dado un grafo ponderado no dirigido.** Dado un grafo ponderado $G=(V, A)$, el algoritmo parte de un grafo $G'=(V, \emptyset)$. Cada vértice es una componente conexa en sí misma. Necesita que las aristas estén ordenadas según el coste y operaciones para saber si dos vértices están en la misma componente conexa y para unir componentes. En cada paso de ejecución se elige la arista de menor costo de A dónde:

- Si une dos vértices que pertenecen a distintas componentes conexas entonces se añade al árbol de expansión G' .
- En otro caso no se coge, ya que formaría un ciclo en G' .

El algoritmo concluye cuando G' sea conexo: cuando se obtiene $n-1$ aristas. A continuación se ve la estructura del algoritmo:

- \Rightarrow Sea T de tipo Conjunto de aristas, el lugar donde se guardarán las aristas del árbol de expansión. Asignar T a \emptyset .
- \Rightarrow Mientras T contenga menos de $n-1$ aristas hacer:
 - Elegir la arista (v, w) de A con menor costo.
 - Borrar (v, w) de A .
 - Si v, w están en distintos componentes conexos entonces añadir (v, w) a T . En otro caso descartar (v, w) .

- **Generación de variantes de trayectoria de redes malladas.** A partir de la red de mayor cantidad de ciclos se analiza la posibilidad de eliminar aquellas aristas que no pertenezcan a la red priorizada de manera tal que genere el menor circuito de los posibles. Para ellos se aplican algoritmos para el recorrido de grafos. De esta manera se calcula la distancia mínima desde un vértice dado hasta cualquier otro; aplicando el principio de optimalidad que plantea que si un camino es mínimo entonces todos sus subcaminos también lo son. La cantidad de tramos incidentes en dicho nodo sea mayor o igual a 2, esto garantiza que se mantenga la condición de red cerrada.
- **Generación de soluciones de diseño que resultan próximas al criterio de eficiencia del decidor.** Se obtiene la cadena con tantos dígitos como tramos halla, donde cada dígito codifica el diámetro a utilizar en el tramos correspondiente. Todas las variantes de redes generadas en las diferentes soluciones generadas en el proceso de búsqueda por el algoritmo de búsqueda por exploración aleatoria del extremo de una función de código variable.
- **Selección de la solución que satisface de la mejor manera el criterio completo de preferencias del decidor.** A partir de bondades gráficas que brinda el sistema se pueden visualizar las solución que satisfacen de la mejor manera el criterio completo de preferencias del decidor esto permiten lograr una adecuada integración entre indicadores de eficiencia formalizables y factores del tipo subjetivo.
- **Elaboración de planos, informes y datos técnicos.** Se realiza el proceso de laboración de la documentación necesaria. Esto incluye la elaboración de planos, informe de materiales y tablas de datos técnicos según los requerimientos de las normas empresariales.

La implementación de algoritmos en problemas de grafos y la ilustración gráfica de las variantes de trayectoria facilitarán al diseñador la toma de decisiones constructivas con gastos mínimos de tiempo y con la precisión requerida