Algoritmos de corte mínimo en grafos

Franss Cruz Davis García Fernando Macuri

***Resumen*—Este artículo desbribe como se aplican los algoritmos de recorrido mínimo dentro del concepto de un grafo y relacionando a la vida real.**

***Index Terms*—Grafo,vértice,rama,algoritmo,corte mínimo.**

I. Introducción

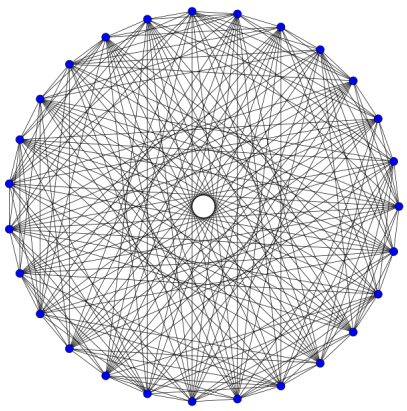
Noción de un grafo

HAy muchas situaciones que puden representarse mediante un esquema formado por dos cosas:

- Un conjunto finitos de puntos.
- Un conjunto de líneas que unen algunos pares de estos puntos.

Por ejemplo los puntos pueden representar participantes de una fiesta de cumpleaños y las uniones corresponden a pares de participantes que se conocen mutuamente. O también estos puntos pueden representarse cruces de calles de una ciudad y las calles sus conexiones. Una red de transporte municipal o una red de tren se representan también por un esquema de este tipo, y a menudo circuitos eléctricos tienen estas carecterísticas similares.En tales casos, los puntos se llaman normalmente vértices(o nodos)y sus uniones ramas.
Con esto llegamos a la noción matemática de grafo, que es uno de los conceptos básicos de la matemática discreta.

Definición de un grafo: Un grafo es un par ordenado (V, E), donde V es algún conjunto y E es un conjunto de subconjuntos de dos puntos de V.Los elementos del conjunto V se llaman vértices del grafo G y los elementos de E se llaman ramas de G.



Grafo completo k_{23}

fransscruz18@gmail.com
yums123@hotmail.com
rfmo2014@hotmail.com

Aplicaciones de los grafos

1. *En Mapas:*
   Una de las aplicaciones más importantes de los grafos en los mapas, es para determinar e intentar disminuir el tiempo y camino en recorridos entre sitios distintos.
   Otra aplicación de los grafos es la del coloreo, es muy usada cuando hablamos de pintar mapas,el objetivo es colorear los vértices de un grafo de tal manera que no existan dos vértices adyacentes del mismo color, ademas de que tenemos que usar el menor número de colores posibles.
   El algoritmo de coloración dice que basta solo de 4 colores para pintar cualquier mapa; este algoritmo es llamado el *"Teorema de los 4 Colores"*.
2. *En Costos:*
   En este tipo de aplicación de los grafos se pueden realizar cálculos de costos; en este caso, la arista del grafo puede representar cosas como: la distancia en kilómetros, el precio de un tiquete de avión o el costo de expandir una red telefónica entre los dos puntos que representan los vértices de los extremos.
   La aplicación consiste en encontrar la forma más rápida o barata de ir de una ciudad a otra o de crear una red telefónica, o el punto más adecuado para instalar un hospital en una ciudad.

Algoritmo: Es una secuencia de instrucciones para resolver un problema en cuestión.
Objetivo: Utilizar los algoritmos de corte mínimo para obtener los menores costos,tiempos,energía en varios ámbitos.

Algoritmos de corte mínimo:

- Algoritmo de Kruskal.
- Algoritmo de Dijkstra.
- Algoritmo De flujo máximo.
- Algoritmo De Flody Y Warshall.

¿Cómo se integra dentro del uso del lenguaje R?

Uno de los paquetes que nos permite realizar análisis de redes en R es igraph. Proporciona rutinas y funciones para crear y manipular grafos con facilidad. Como ejemplo, vamos a crear un gráfico no dirigido ponderado para luego calcular el camino más corto entre dos vértices. Para calcular esta ruta, igraph emplea en este caso el algoritmo de Dijkstra. Para obtener un grafo no dirigido ponderado necesitamos crear un listado de aristas (edgelist) con las pesas de cada arista (o distancias entre ellas). Creamos tres columnas, las dos primeras representan la conexión (las aristas) entre los vértices y la tercera las pesas de cada arista. Al incorporar weight en el título de la columna, igraph asignará como atributo el peso de las aristas a las parejas de vértices.

II. Estado del arte

1. Artículos científicos:

a) Dijkstra EW. A Note on Two Problems in Connection with Graphs. Numerische Mathematik.

b) Fuhao Z, Jiping L. An Algorithm of Shortest Path Based on Dijkstra for Huge Data. En: Fourth International Conference on Fuzzy Systems and Knowledge Discovery.

c) Nazari S, Meybodi MR, Salehigh MA, et al. An Advanced Algorithm for Finding Shortest Path in Car Navigation System. En: First International Conference on Intelligent Networks and Intelligent Systems.

d) Shao F, Deng W, Zhang B. Traffic Information Management and Promulgating System Based on GIS. En: Optoelectronics and Image Processing (ICOIP), 2010 International Conference on.

e) Hart PE, Nilsson NJ, Raphael B. A Formal Basis for the Heuristic Determination of Minimum Cost Paths. Systems Science and Cybernetics, IEEE Transactions on.

f) Noto M, Sato H. A method for the shortest path search by extended Dijkstra algorithm. IEEE International Conference on Systems, Man, and Cybernetics.

g) Gutman RJ. Reach-Based Routing: A New Approach to Shortest Path Algorithms Optimized for Road Networks. En: Proceedings of the Sixth Workshop on Algorithm Engineering and Experiments and the First Workshop on Analytic Algorithmics and Combinatorics. New Orleans, LA, USA.

h) Goldberg AV, Harrelson C. Computing the shortest path: A search meets graph theory. In: Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms; Vancouver, British Columbia: Society for Industrial and Applied Mathematics; 2005.

i) Geisberger R, Sanders P, Schultes D, et al. Contraction hierarchies: faster and simpler hierarchical routing in road networks. In: Proceedings of the 7th international conference on Experimental algorithms; Provincetown, MA, USA: Springer-Verlag; 2008.

j) Gonzalez H, Han J, Li X, et al. Adaptive fastest path computation on a road network: a traffic mining approach. In: Proceedings of the 33rd international conference on Very large data bases Vienna, Austria: VLDB Endowment; 2007.

2. Algunas menciones:

a) Este artículo presenta el algoritmo de Dijkstra clásico en detalle e ilustra el método de implementación del algoritmo y las desventajas del algoritmo: los nodos de red requieren memoria de clase cuadrada, por lo que es difícil cuantificar la ruta más corta de los nodos principales. Al mismo tiempo, describe el algoritmo de nodo adyacente que es un algoritmo de optimización basado en el algoritmo de Dijkstra. El algoritmo aprovecha al máximo la relación de conexión de arcos en la información de topología de red, y evita el uso de una matriz de correlación que contiene un valor infinito sustancial, por lo que es un análisis más adecuado de la red para datos masivos. Está demostrado que el algoritmo puede ahorrar mucha memoria y es más adecuado para la red con enormes nodos.

b) Proponemos algoritmos de ruta más cortos que utilizan la búsqueda A * en combinación con una nueva técnica de límite inferior teórico de gráficos basada en puntos de referencia y la desigualdad de triángulos. Nuestros algoritmos calculan caminos óptimos más cortos y trabajan en cualquier gráfico dirigido. Proporcionamos resultados experimentales que demuestran que el algoritmo más eficiente supera los algoritmos previos, en particular la búsqueda A * con límites euclidianos, por un amplio margen en redes de carreteras y en algunas familias de problemas sintéticos.

c) Presentamos una técnica de planificación de ruta basada únicamente en el concepto de contracción del nodo. Los nodos primero se ordenan por 'importancia'. Luego se genera una jerarquía al contratar iterativamente el nodo menos importante. Contratar un nodo significa reemplazar los caminos más cortos que pasan por v por atajos. Obtenemos un algoritmo de consulta jerárquica utilizando búsqueda bidireccional de ruta más corta. La búsqueda directa utiliza solo los bordes que conducen a nodos más importantes y la búsqueda hacia atrás utiliza solo los bordes procedentes de nodos más importantes. Para las rutas más rápidas en redes de carreteras, el gráfico permanece muy disperso a lo largo del proceso de contracción usando heurísticas bastante simples para ordenar los nodos. Tenemos tiempos de consulta cinco veces menores que las mejores técnicas jerárquicas anteriores de aceleración basadas en Dijkstra y una sobrecarga espacial negativa, es decir, la estructura de datos para el cálculo de distancia necesita menos espacio que el gráfico de entrada. Los CH se pueden combinar con muchas otras técnicas de planificación de rutas, lo que permite un mejor rendimiento para el enrutamiento muchos a muchos, enrutamiento de nodo de tránsito, enrutamiento dirigido a objetivos o escenarios móviles y dinámicos.

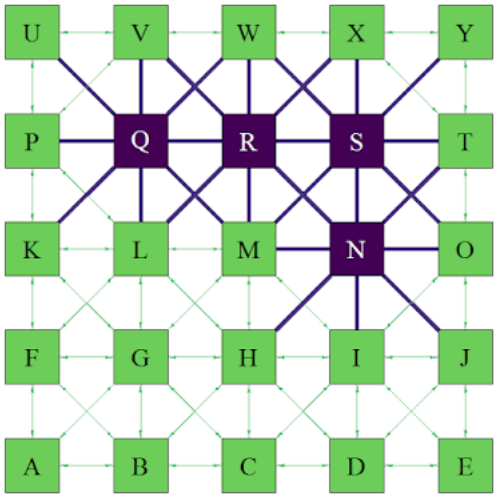
III. Diseño del experimento

1. **Algoritmo de Dijkstra**
   Idea intuitiva: El algoritmo de dijkstra, determina la ruta más corta desde un nodo origen hacia los demás nodos para ello es requerido como entrada un grafo cuyas aristas posean pesos.
   Definición: Sean G y G_r dos grafos, $\mathbb{R} = \{r_1, r_2, ..., r_n\}$ un conjunto de reglas de reescritura de grafos, se puede afirmar G_r que es un grafo reducido a partir de G si al aplicar las reglas de reescritura especificadas en $\mathbb{R}$ al grafo G se obtiene el grafo G.

   A partir de lo anterior, se enuncia un algoritmo de reducción de grafos ,sin pérdida de información relevante para la búsqueda de caminos mínimos. Posteriormente, se presenta una modificación al algoritmo de Dijkstra para realizar búsqueda de caminos mínimos en grafos reducidos con el algoritmo de reducción propuesto.
2. **Reducción de grafos**
   La reducción de grafos se realiza utilizando el Algoritmo 1, en este epígrafe se presenta dicho algoritmo y se expone una breve explicación del mismo.

   En el caso del algoritmo de reducción que se propone, se debe garantizar que no existan vértices reducidos que sean adyacentes con el objetivo de poder obtener un camino optimo de igual costo al obtenido en una búsqueda en el grafo original, para contribuir al logro de este objetivo, se enuncia la siguiente definición:
   Definición: Sea un grafo $G = (V, E)$ y una relación de equivalencia RE sobre V, un vértice $v_i \in V$ es interior si $\forall v_j \in V$, tal que v_i y v_j son adyacentes, se cumple que $RE(v_i, v_j)$, o sea, $([v_i] = [v_j])$. Un vertice $v_i \in V$ es exterior si o $\exists(v_i, v_j) \in E$ o $[\exists(v_i, v_j) \in E$ y $-RE(v_i, v_j)]$.

   Si la entrada del algoritmo es una relación de equivalencia, lo primero que se hace es obtener una partición a partir de dicha relación. Para crear esta partición se puede seguir una estrategia voráz que divida el conjunto de vértices V en subconjuntos disjuntos seun RE. La estrategia consistiría en comparar cada elemento del conjunto V con los restantes elementos para determinar cuáles están relacionados (segúnRE) y a partir de estos crear las clases que forman la partición. Si una clase esta formada por un solo elemento, este será un vértice exterior, Si la clase tiene más de un elemento, entonces todos estos serán vértices interiores. Para construir el conjunto de vértices del grafo reducido, se crea un vértice por cada clase de P.
   Para adicionar una arista al grafo reducido, los dos vértices que la forman deben pertenecer a clases distintas. En cada caso de que exista más de una arista, en el grafo de entrada del algoritmo, entre los vértices que pertenecen a dos clases distintas, se adiciona la de menor costo.
3. **Técnicas a utilizar**
   Con el algoritmo de Dijkstra es posible determinar la distancia más corta (de menos esfuerzo o costo) entre un nodo inicial y cualquier otro nodo en un grafo. La idea del algoritmo es calcular continuamente las distancias más largas cuando se efectúa una actualización. Como ejemplo,en el siguiente grid, se asumeque los nodosen morado son sumamente costosos, por lo que elalgoritmo de Dijkstra deberá hallar la ruta que minimice elcosto total entre cualquier tupla de nodos.



El algoritmo consiste formalmente en los siguientes pasos:

- Inicialización de todos los nodos con distancia ïnfinita", iniciar el nodo de comienzo con 0.
- Marcar las distancias del nodo inicial como permanentes, todas las demás como temporales .
- Establecer el nodo inicial como activo.
- Si la distancia calculada de un nodo es menor a la actual, actualiza la distancia y dija el nodo actual como antecesor, este paso es llamado actualización y es la idea central de Dijkstra.
- Estableciendo el nodo con la distancia temporal mínima como activa, marca sus distancia como permanente.
- Repita los pasos del 4 al 7 hasta que no queden nodos con una distancia permanente cuyos vecinos tengan aún distancias temporales.