

Un acercamiento a la Teoría de grafos y matrices de Robinson.

Erika Patricia Domínguez Ríos, David Romero Vargas.
Facultad de Ciencias Físico Matemáticas, BUAP.
Instituto de Matemáticas Unidad Cuernavaca, UNAM.

Resumen

Las matemáticas tienen diversas aplicaciones, una de las áreas es la arqueología, donde uno de los problemas es ordenar los cacharros cronológicamente para ello se usan matrices binarias, en las cuales se aplica un algoritmo para saber si es o no una matriz de Robinson; si esto ocurre se consigue el orden buscado. Sin embargo para grandes matrices, encontrar combinaciones óptimas es lo que se conoce en ciencias de la computación como un problema NP-complejo.

Hablando de grafos de intervalos, si se da la intersección de intervalos es sencillo encontrar su grafo asociado, el problema es si dado un grafo, este es o no un grafo de intervalos.

Agradecimientos

Se agradece apoyo económico para el desarrollo del presente proyecto al:

Proyecto FORDECYT 265667 "Programa para un avance global e integrado de la Matemática Mexicana"

Teoría de grafos

La teoría de grafos, también llamada teoría de gráficas, es una rama de las matemáticas y las ciencias de la computación que estudia las propiedades de los grafos.

Definición. Un grafo $G = (V, E)$ es una pareja ordenada en la que V es un conjunto no vacío de vértices y E es un conjunto de aristas, V consta de pares no ordenados de vértices, tales como $\{x, y\} \in E$ entonces decimos que x e y son adyacentes; y [en el grafo] se representa mediante una línea no orientada que una dichos vértices.

Definición. Diremos que es un digrafo, si el grafo es dirigido. Se denota D , y entonces el par (x, y) es un par ordenado, y se representa con una flecha que va de x a y , y decimos que $\{x, y\} \in E$.

Definición. Un grafo simple o simplemente grafo es aquel que acepta una sola arista uniendo dos vértices cualesquiera. Esto es equivalente a decir que una arista cualquiera es la única que une dos vértices específicos. Es la definición estándar de un grafo.

Algunos componentes de los grafos son los siguientes:

- **Aristas:** Son las líneas con las que se unen los vértices de un grafo.
- **Aristas adyacentes:** 2 aristas son adyacentes si convergen en el mismo vértice.
- **Aristas paralelas:** Son dos aristas conjuntas si el vértice inicial y final son el mismo.
- **Arista cíclicas:** Es la arista que parte de un vértice para entrar en sí mismo.
- **Camino:** Se denomina camino de un grafo a un conjunto de vértices interconectados por aristas. Dos vértices están conectados si hay un camino entre ellos.

Definición. Se define una matriz de incidencia como una matriz con zeros-ones que se utiliza para representar relaciones binarias.

Podemos describir una relación binaria mediante una matriz de incidencia y un grafo.

- Las columnas de la matriz representan las aristas del grafo.
- Las filas representan a los distintos nodos.
- Por cada nodo unido por una arista, ponemos un uno (1) en el lugar correspondiente, y llenamos el resto de las ubicaciones con ceros (0).

En el ejemplo de la figura, si sumamos las cantidades de 1's que hay en cada columna, veremos que hay solo dos. Pero si sumamos las cantidades de unos 1's que hay por cada fila, comprobaremos que los nodos 1 y 3 poseen un valor de 2. Ese valor indica la cantidad de aristas que inciden sobre el nodo.

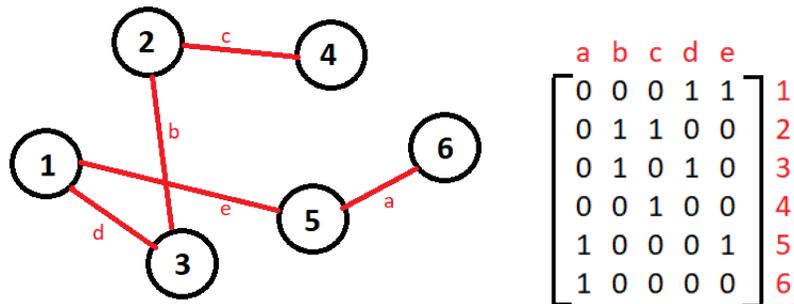


Figura 1: Grafo y su matriz de incidencia

Las matrices de incidencia con los la propiedad de 1's consecutivos (es decir, matrices zero-one que permiten una reorganización de filas que agrupa los 1's en cada columna separada), y de ordenar las filas de una matriz para así juntar los 1's. Esto tiene mucho en común con el problema arqueológico de 'secuencia de datos' primero formulado en 1899 por Flinders Petrie.

Lo que distingue a los dos es que en la situación arqueológica uno se preocupa por matrices zeros-ones que permiten una reorganización de las filas que solo agrupa aproximadamente los 1's en cada columna por separado. (Normalmente las filas representan decir tumbas, y las columnas representan objetos o aspectos de objetos que pueden o no estar presentes en una tumba dada.

La reorganización de filas determina la cronología ordinal de las tumbas, y esto a su vez asigna un rango de 'fechas de secuencia' para cada objeto o característica). A pesar de esta importante diferencia, se mantiene aquí que los problemas matemáticos pueden ser una fuente conveniente de algoritmos.

Definición. Una matriz es de Petrie si es una matriz binaria y cumple con la propiedad de 1's consecutivos.

Un ejemplo de una matriz de Petrie es:

$$\begin{bmatrix} 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

Ya que si nos fijamos en las columnas de la matriz los 1's ocurren consecutivos. Dada una matriz de incidencia A esta puede ser re-organizada por filas para obtener todos los 1's en cada columna separada, sin embargo para grandes matrices encontrar combinaciones óptimas es lo que se conoce en la ciencia de la computación como un problema NP-duro o NP-complejo. lo que significa que el problema de optimización, aunque finito, no puede ser resuelto en una cantidad de tiempo practico aún utilizando computadores rápidos.

Teoria de la complejidad

Para conocer un poco mas sobre problemas NP-duros, veremos la teoria de la complejidad.

La teoría de la complejidad computacional es una rama de la teoría de la computación que se centra en la clasificación de los problemas computacionales de acuerdo con su dificultad, y en la relación entre dichas clases de complejidad.

Un problema se cataloga como "inherentemente difícil" si su solución requiere de una cantidad significativa de recursos computacionales, sin importar el algoritmo utilizado. La teoría de la complejidad computacional formaliza dicha aseveración, introduciendo modelos de computación matemáticos para el estudio de estos problemas y la cuantificación de la cantidad de recursos necesarios para resolverlos, como tiempo y memoria.

Una de las metas de la teoría de la complejidad computacional es determinar los límites prácticos de qué es lo que se puede hacer en una computadora y qué no. Otros campos relacionados con la teoría de la complejidad computacional son el análisis de algoritmos y la teoría de la computabilidad. Una diferencia significativa entre el análisis de algoritmos y la teoría de la complejidad computacional, es que el primero se dedica a determinar la cantidad de recursos requeridos por un algoritmo en particular para resolver un problema, mientras que la segunda, analiza todos los posibles algoritmos que pudieran ser usados para resolver el mismo problema.

La teoría de la complejidad computacional trata de clasificar los problemas que pueden, o no pueden ser resueltos con una cantidad determinada de recursos. A su vez, la imposición de restricciones sobre estos recursos, es lo que la distingue de la teoría de la computabilidad, la cual se preocupa por qué tipo de problemas pueden ser resueltos de manera algorítmica.

Muchas veces podemos evitar utilizar la fuerza bruta en los problemas para obtener soluciones en tiempo polinómico. Sin embargo, para algunos problemas esto no ha podido

lograrse, es decir, no se conocen algoritmos que los resuelvan en tiempo polinómico. Quizás estos problemas tengan algoritmos en tiempo polinomial que se basan en principios por ahora desconocidos, o quizás estos problemas no pueden ser resueltos en tiempo polinómico, debido a que son “inherentemente difíciles”.

La clase de complejidad NP consta de los problemas “verificables” en tiempo polinómico. Por verificable se entiende a un problema tal que dado un certificado de solución (candidato a solución), se puede verificar que dicho certificado es correcto en un tiempo polinómico en el tamaño de la entrada. A los problemas en la clase NP usualmente se les llama problemas NP.

El término NP proviene de no determinista en tiempo polinómico y se deriva de una caracterización alternativa de esta clase, donde se utilizan Máquinas de Turing no deterministas. Informalmente, se puede definir la clase NP en términos de un algoritmo no determinista.

El algoritmo mencionado está compuesto por 2 etapas separadas. Dada una instancia del problema I , la primera etapa simplemente “adivina” un candidato a solución S . Entonces, la etapa de verificación recibe como entrada a I y a S , y procede a realizar el cómputo de una manera determinista, finalmente deteniéndose con la respuesta “sí”, o con la respuesta “no”, o sigue computando sin detenerse.

Al igual que la clase P, la clase NP es insensible a la elección del modelo de cómputo no determinista, debido a que dichos modelos son equivalentes polinómicamente.

La relación entre las clases P y NP es fundamental para la teoría de la NP-completitud. Intuitivamente, creemos que P es un subconjunto de NP. Y, efectivamente, cada problema de decisión resuelto por un algoritmo de tiempo polinomial determinista, también puede ser resuelto por un algoritmo de tiempo polinomial no determinista. Simplemente se necesita observar que cualquier algoritmo determinista puede ser utilizado en la etapa de verificación de un algoritmo no determinista. Si B es un problema de P, y A es un algoritmo de tiempo polinomial para B, entonces se puede construir un algoritmo de tiempo polinomial no determinista para B, simplemente utilizando A en la etapa de verificación e ignorando la etapa de adivinación. Por tanto, si B pertenece a P, entonces B también pertenece a NP.

La pregunta $P=NP$ es una de las más importantes en el campo de las ciencias de la computación, debido a las grandes repercusiones que habría, en caso de encontrarse una solución. Si $P=NP$, cualquier problema polinómicamente verificable sería polinómicamente decidible. La mayoría de los investigadores cree que estas clases no son iguales, porque se ha realizado bastantes esfuerzos, sin éxito, para encontrar algoritmos de tiempo polinomial para varios problemas en NP. Los investigadores también han tratado de probar

que las clases son distintas, pero eso conllevaría a mostrar que no existe un algoritmo “eficiente” para reemplazar a la búsqueda por fuerza bruta.

Un problema del tipo NP-complejo es el problema de viajero responde a la siguiente pregunta: dada una lista de ciudades y las distancias entre cada par de ellas, ¿cuál es la ruta más corta posible que visita cada ciudad exactamente una vez y al finalizar regresa a la ciudad origen? Este es un problema NP-Hard dentro en la optimización combinatoria, muy importante en la investigación de operaciones y en la ciencia de la computación.

Este problema puede ser modelado como un grafo no dirigido, de manera que las ciudades sean los vértices del grafo, los caminos son las aristas y las distancias de los caminos son los pesos de las aristas. Esto es un problema de minimización que comienza y termina en un vértice específico y se visita el resto de los vértices exactamente una vez. Con frecuencia, el modelo es un grafo completo (cada par de vértices es conectado por una arista). Si no existe camino entre un par de ciudades, se añade arbitrariamente una arista larga para completar el grafo sin afectar el recorrido óptimo.

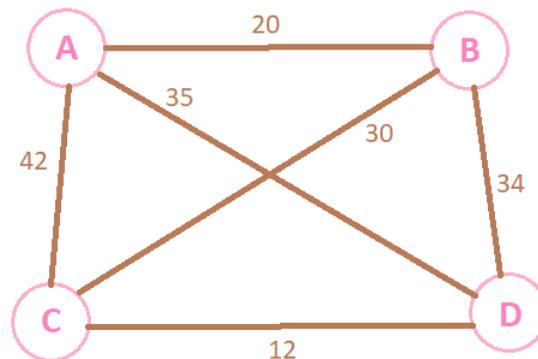


Figura 2: Grafos de el problema del viajero

Matrices de Robinson.

Una matriz simétrica completa cumple la propiedad de Robinson si sus entradas varían monótonamente de la diagonal a lo largo de las filas y las columnas; precisamente, si sus entradas disminuyen (respectivamente aumentan).

Notamos lo siguiente: si todas las entradas son iguales, la matriz es claramente Robinsoniana y si todas las entradas son diferentes, entonces es suficiente ordenar una sola línea y verificar que la permutación correspondiente se ajuste a la Propiedad de Robinson; en ambos casos, la complejidad sigue siendo $O(n^2)$.

Para relacionar una matriz de Petrie y una de Robinson se tiene lo siguiente.

Teorema. A es una matriz de Petrie si y solo si $R = AA^t$ es una matriz de Robinson.

Ejemplo. Sea $A =$

$$\begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

y haciendo el producto de AA^t obtenemos $R =$

$$\begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 2 & 1 & 0 & 0 & 0 \\ 0 & 1 & 2 & 1 & 0 & 0 \\ 0 & 0 & 1 & 2 & 1 & 0 \\ 0 & 0 & 0 & 1 & 2 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}$$

La importancia de el teorema es debido a que a algunos modelos los satisfacen únicamente las matrices de Robinson.

Grafos de Intervalos.

Los Grafos de intervalos son los grafos que se forman a partir de la intersección de intervalos en una línea, estos surgen naturalmente en el proceso de modelar situaciones de la vida real, especialmente aquellas involucrando dependencias de tiempo u otras restricciones que son de naturaleza lineal.

Cincuenta años se usaron grafos de intervalos para modelar la estructura genética, y desde entonces docenas de artículos describieron aplicaciones de estos en áreas tan diversas

como la biología, psicología, sociología, gestión, genética, ingeniería, programación, y transporte.

Un ejemplo es el siguiente, dados los intervalos, existe una arista entre cada intervalo que se intersecte.

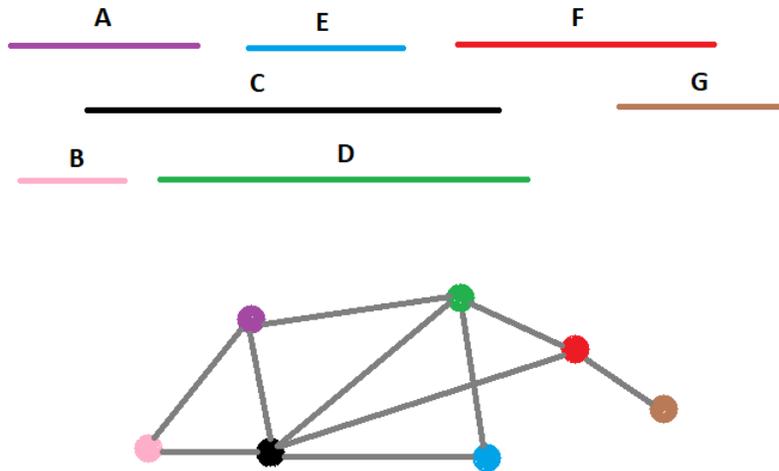


Figura 3: Grafos de intervalos

Notamos que dado la intersección de intervalos es relativamente sencillo encontrar el grafo asociado, el problema consta cuando ocurre el recíproco, es decir, que dado un grafo, como saber si es de intervalos o no.

La determinación de si un grafo dado $G = (V, E)$ es un grafo de intervalo se puede hacer en tiempo $O(|V| + |E|)$ mediante la búsqueda de un ordenamiento.

El algoritmo de reconocimiento de tiempo lineal original del Booth y Lueker (1976) se basa en su compleja estructura de datos PQ-tree, pero Habib et al. (2000) mostró cómo resolver el problema más simplemente usando búsqueda en profundidad lexicográfica.

Busqueda en profundidad lexicográfica.

Se recorre el árbol por ramas, un ejemplo de ello es la siguiente imagen:

Otro tipo de búsqueda en los árboles es la **Busqueda en amplitud**, la cual recorre el árbol por niveles:

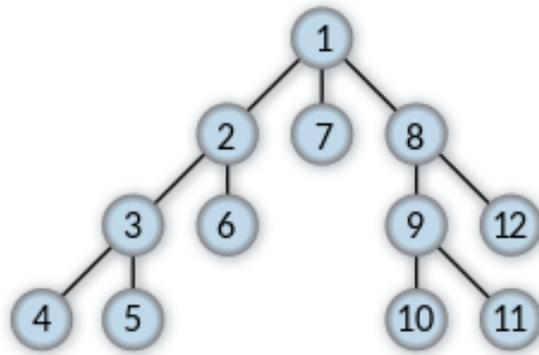


Figura 4: Búsqueda de profundidad

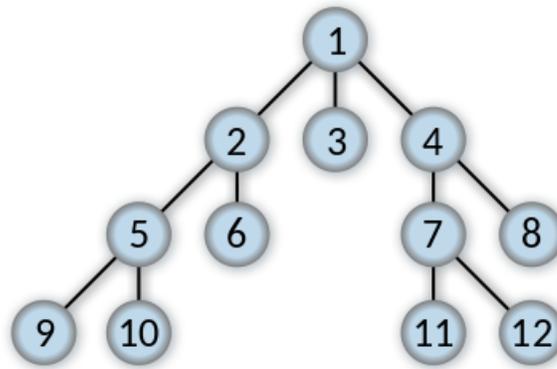


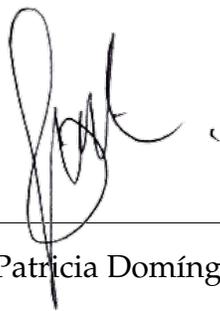
Figura 5: Búsqueda de amplitud

Una de las aplicaciones de los grafos de intervalos se realizó a modelos matemáticos de población biológica, en concreto red alimentarias.

Otras aplicaciones incluyen la genética, bioinformática, y ciencias de la computación. Encontrar un conjunto de intervalos que representan un grafo de intervalos se puede utilizar también como una forma de montaje de subsecuencias contiguas en mapeo de ADN. Los grafos de intervalos son usados para representar problemas de asignación de recursos en investigación de operaciones y teoría de planificación, Cada intervalo representa una solicitud de un recurso por un período específico de tiempo, el problema del conjunto independiente de peso máximo para los grafos representa el problema de encontrar el mejor subconjunto de solicitudes que pueden ser satisfechas sin conflictos. Los grafos de intervalos también juegan un papel importante en el razonamiento temporal.

Referencias

- [1] Derek, G. Corneil & Stephan, O. (2009). The lbfs structure and recognition of interval graphs. Society for Industrial and Applied Mathematics.
- [2] David, G. Kendall (1969). Incidence matrices, interval graphs and seriation in archeology. Pacific Journal of Mathematics.
- [3] D. Fortin (2017). Robinsonian Matrices: Recognition Challenges. Inria, Paris, France: Springer EE. UU.
- [4] Pascal. Prea (2014). An Optimal Algorithm To Recognize Robinson Dissimilarities. Inria, Paris, France: Springer EE. UU.
- [5] Innar. Liiv (2010). Seriation and Matrix Reordering Methods: An Historical Overview. Estonia. Wiley InterScience.



Erika Patricia Domínguez Ríos.