

Aplicación de la coloración de grafos a la telefonía móvil
Nils Murrugarra Llerena
Universidad Nacional de Trujillo
Escuela académico profesional de informática
2006

Resumen: En el presente paper tratare del problema de coloración de grafos y como se puede aplicar a la telefonía celular. En la parte de la introducción se da un ejemplo que se puede modelar mediante grafos. En la parte de Desarrollo del tema la he dividido en fundamente Teórico, Enunciado del problema tratado y Aplicación en la telefonía móvil. En lo referente al Fundamento Teórico he tenido en cuenta Algoritmos voraces y problemas NP. Luego en lo referente al enunciado del problema tratado he tenido su definición, su modelado mediante un algoritmo voraz y su algoritmo voraz. Y con lo referente a la Aplicación en la telefonía móvil ahí se trata como el problema de coloración de grafos es aplicable a esa área. Y para finalizar presento mis conclusiones y referencias.

1. Introducción

Quien no ha escucha del algoritmo de dijkstra para encontrar la ruta más corta desde un origen a un destino, bueno para comentarles una de las formas de modelarlo es mediante un algoritmo voraz, que conforme se recorra el grafo se vaya seleccionando la arista de menor longitud. Como este problema hay otros que también demuestran ser aplicables en nuestra realidad, muchos de los cuales son extraídos de la informática teórica. En el presente paper tratare el problema de coloración de grafos y hablaré de la aplicación que tiene a la telefonía móvil.

2. Desarrollo del tema

2.1.Fundamento teórico

a. Algoritmos Voraces(greedy)

a.1. Definición

El enfoque que aplican es “miope”, y toman decisiones con la información que tienen disponibles de modo inmediato, sin tener en cuenta los efectos que pueden tener en el futuro.

Estos algoritmos se denominan voraces, ya que seleccionan el mejor bocado que pueden tomar sin preocuparse del futuro, nunca cambian de opinión, una vez que un candidato haya sido seleccionado, queda allí siempre.

a.2. Características

- a. *Lista de Candidatos:* Es el conjunto o lista de elementos que disponemos.
- b. *2 conjuntos:* Se definen 2 conjuntos: uno de los seleccionados y considerados y otro el de seleccionados pero rechazados.
- c. *Función es solución:* Esta función es la encargada si ya se lleo o no a una solución con el conjunto de candidatos.

- d. *Función es Factible*: Esta función es la encargada de determinar si es posible o no encontrar una solución a nuestro problema al añadir algún candidato.
- e. *seleccionar Candidato*: Elige el candidato más adecuado para poder llegar a la solución.
- f. *Función Objetivo*: Función que expresa el valor de la solución que se desea optimizar (minimizar o maximizar).

a.3. Proceso

Los algoritmos voraces avanzan paso a paso, y en cada uno de sus pasos se considera añadir el mejor candidato a la posible solución. Si el conjunto de de candidatos seleccionados ya no fuera factible, rechazamos al candidato seleccionado, por el contrario si el conjunto aumentado sigue siendo factible, entonces añadimos al candidato actual.

b. Problemas NP

b.1. Introducción

En la vida real existen muchos problemas prácticos para los cuáles no se conoce, ningún algoritmo eficiente, pero cuya dificultad intrínseca no ha conseguido demostrar nadie.

Entre estos se encuentran problemas tan conocidos como el agente viajero, el coloreado óptimo de un grafo, los ciclos hamiltonianos, la programación entero, la búsqueda del camino simple más largo de un grafo, etc.

b.2. Definición

Antes de definir que es una clase de tipo NP, definiremos que es una clase del tipo P, en está están todos los algoritmos eficientes y que su tiempo de ejecución es un tiempo polinómico.

Por otro lado los problemas de la clase NP, en primer lugar NP no significa “no polinómico”, mas bien lo que quiere decir es de “tiempo polinómico no determinista”. Además los problemas NP se basan en *problemas de decisión*, para los cuales sus respuestas son si ó no (verdadero ó falso).

Por ejemplo: “Buscar un ciclo hamiltoniano en un grafo G” no es un problema de decisión, pero “¿Es hamiltoniano el grafo G?” si que lo es ya que su respuesta es afirmativa o negativa.

Tomemos en cuenta un ejemplo más general, teniendo en cuenta el problema en decidir si un grafo es hamiltoniano. Aun cuando se cree que este problema es difícil, es verificable eficientemente si el grafo nos muestra un ciclo hamiltoniano y del cual nos es fácil determinar si es correcto o no, por otro lado nada que nos muestre el grafo podrá convencernos de que el grafo es hamiltoniano si no lo es realmente.

b.3. Algoritmos Heurísticos y Aproximados

a. Introducción

El conocimiento de que un problema es difícil de resolver es instructivo. Puede ser útil si evita que desperdiciemos tiempo buscando un algoritmo que probablemente no exista. Sin embargo esto no hace que el problema desaparezca. En algunas ocasiones uno debe de hallar algún tipo de solución para el problema, tanto si es difícil como si no. Estos son los dominios de la heurística y los algoritmos aproximados.

b. Algoritmos heurísticos

Al hablar de algoritmos heurísticos, nos referimos a un procedimiento que puede producir una buena solución para nuestro problema, incluso una solución óptima si somos afortunados, pero que por otro lado puede no producir una solución, o dar lugar a una que no sea precisamente óptima. Una forma en la que son muy usados es con los algoritmos voraces.

c. Algoritmos Aproximados

Un algoritmo aproximado es un procedimiento que siempre proporciona algún tipo de solución para el problema, aun cuando quizá no llegue a encontrar la solución óptima. Para que sea útil, también debe de ser posible calcular una cota buena de la diferencia, buena se refiere a la razón entre la solución óptima y la producida por el algoritmo aproximado.

2.2. Enunciado del problema tratado

a. Enunciado del Problema

Sea $G = \langle N, A \rangle$ un grafo no dirigido. Deseamos colorear los nodos de G de tal manera que no haya dos nodos adyacentes del mismo color. El problema pregunta además cual es el mínimo número de colores que se necesita. Este número mínimo se denomina *número cromático* del Grafo.

b. Modelado del problema de coloración mediante un algoritmo voraz

- a. *Lista de Candidatos*: Todos los nodos del Grafo.
- b. *2 conjuntos*: Uno el cual contiene los nodos coloreados y otro el que contiene los nodos sin colorear.
- c. *Función es solución*: Si el Grafo cumple con el enunciado del coloreado de Grafos.
- d. *Función es Factible*: Si al seleccionar un nodo y pintarlo con el color usado en ese momento se sigue cumpliendo que sus nodos adyacentes tengan distinto color que él.
- e. *seleccionar Candidato*: Elige un nodo del conjunto de nodos sin colorear.
- f. *Función Objetivo*: (Minimizar) Utilizar el mínimo numero de colores para pintar un grafo de acuerdo al problema de coloreado de Grafos.

c. Algoritmo

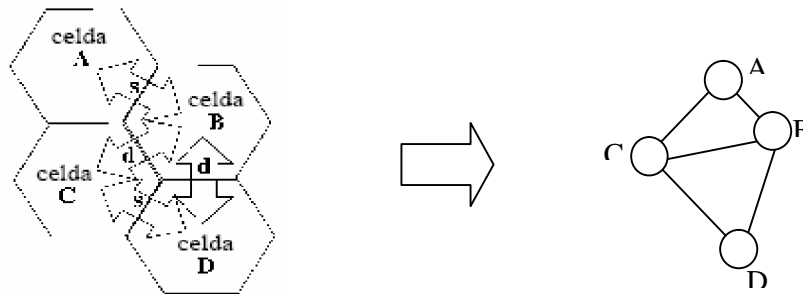
- a. Escoger inicialmente un color y un nodo arbitrario como punto de partida.
- b. Tratar de Asignarle ese color al mayor número posible de nodos, respetando que sus nodos adyacentes tengan distinto color.

- c. Escoger otro vértice aun no coloreado y un color distinto y repetir el proceso hasta haber coloreado todos los vértices.

2.3. Aplicación en la telefonía Móvil

La licencia de explotación del espectro radioeléctrico es uno de los principales costes que debe soportar un operador móvil. En estos casos, la reducción del número de frecuencias en el diseño de una red puede suponer un ahorro considerable. También generaría ahorro poder reutilizar frecuencias en distintas celdas (son las zonas en la cuales se divide la zona de cobertura, estas celdas tienen asignadas un número de frecuencias) de la red.

Una red de telefonía celular se compone de una serie de celdas que pueden interferirse según distintos factores. La relación de interferencia (no pueden tener la misma frecuencia celdas adyacentes) entre las celdas permite modelar la red a través de un grafo, correspondiéndose los vértices con las celdas y designando a las aristas la relación de interferencia entre celdas. De este modo el problema de asignación de frecuencias puede considerarse como el problema de coloreamiento de grafos, ya que cada frecuencia va a ser representada mediante un color y determinaremos de forma óptima la menor cantidad de número de frecuencias. Por ejemplo:



3. Conclusiones

- Ø Los problemas de la informática teórica son aplicables a problema de la realidad.
- Ø Hay que darle más importancia a la informática teórica.
- Ø Necesitamos de algunas herramientas de la algoritmia para poder solucionar estos problemas como a los algoritmos heurísticos y aproximados.

4. Referencias

- [1] G.BRASSARD, P.BRATLEY, Fundamentos de Algoritmia, Pág. 525-528, primera edición, Madrid, 1997.
- [2] JEFFREY J. MCCONNELL, Analysis of Algorithms: An Active Learning Approach, capítulos 8 y 9, Jones and Bartlett Publishers © 2001
- [3] J.M.SAAVEDRA RONDO, Separata de técnicas de construcción de programas, pág. 57-58.
- [4] JAVIER OZÓN, Contribución al coloreado de grafos y la redes pequeño mundo pág. 37-45, 27 de mayo del 2001.