Análisis de Algoritmos (MST, Cliques en grafos) (Mayo 2017)

Vasquez R. Javier

Abstract*—This article talks about the description of the analysis and the detail of the algorithms which they develop a particular solution for each one, for each problem raised solutions are analyzed and for many cases it is necessary to resort to algorithms more efficient than the algorithms of Brute force, for the problems of resource optimization, maximization or minimization, we take into account linear programming algorithms (LP) that have as characteristics some variables that are converted into equations and inequalities at the moment when the constraints are added to the problem and this solution is no longer that a system which must be solved with operations. For the case in which we want to find an optimization of routes and resources such as the problem of network theory, they take into account the algorithms of minimum spanning trees (MST), which aim to design a tree from a directed graph connected passing through all its edges and that as its name indicates, it looks for the minimum extension of its branches to reach all its leaves with the lowest cost of displacement, two are raised, the algorithm of Kruskal and the algorithm of Primm, that give a different solution to the same problem. For the case of cliques in graphs refers to the sub-graphs that are contained in a parent graph and that all edges are interconnected between them, ie, are complete graphs, for this solution we propose the Hammilton algorithm for cycles

***Index Terms*—Linear Programming, Minimum Spanning Trees, Cliques, Graph, Kruskal, Primm**

I. Introduction

ESTE DOCUMENTO SE REFIERA A LAS DESCRIPCION, ANALISIS Y CONTEMPLACION DE TRES TIPOS DE ALGORITMOS LOS CUALES, CADA UN POR SU PARTE, CONTEMPLAN SOLUCIONES DISTINTAS A PROBLEMAS, A LA VEZ TAMBIEN SE ANALIZA SU EFICIENCIA DE PROCESAMIENTO COMO TAMBIEN SU EFICIENCIA EN MEMORIA, SU COMPLEGIDAD DE IMPLEMENTACION, Y TAMBIEN ALGUNOS EJEMPLOS DE SUS APLICACIONES. LOS ALGORITMOS QUE SON ANALIZADOS SON: PROGRAMACION LINEAL (LP), ARBOL DE EXPANCION MINIMA (MST), Y CLIQUES EN GRAFOS

Este articulo es sometido a revision el dia 29 de Mayo del 2017
Javier Felipe Vasquez Roldan Estudiante de ingenieria de sistemas, Desarrollador movil Linktic S.A.S
(Javier-vasquez@javeriana.edu.co)

II. Arboles de Expansión minima (MST)

El algoritmo del árbol de expansión mínima es un modelo de optimización de redes que consiste en enlazar todos los nodos de la red de forma directa y/o indirecta con el objetivo de que la longitud total de los arcos o ramales sea mínima (entiéndase por longitud del arco una cantidad variable según el contexto operacional de minimización, y que puede bien representar una distancia o unidad de medida).

A. Entendiendo un Árbol de expansión mínima

Se asume que se tiene un grafo conexo y dirigido G=(V,E) con una función de peso w: E -> R y se quiere encontrar el árbol de expansión mínima para G

La estrategia voraz hace crecer el árbol de expansión mínima una arista a la vez y maneja un conjunto de aristas A manteniendo la siguiente invariante de ciclo:

Antes de cada iteración, A es un subconjunto del árbol de expansión mínima

Se determina una arista (u,v) que no viole la invariante en el sentido que AU{(u,v)} es también un subconjunto del árbol de expansión mínima.

```
GENERIC-MST (G; w)
A = 0;
while A no conforme un arbol de expansion minima
      busque una arista (u,v) que este a salvo para A
      A =AU{(u,v)}
return A
```

entonces se afirma que cumpliendo la invariante de ciclo, se genera un grafo aciclico donde sus vertices conectados dan la minima altura del arbol formado por estos, desde el nodo inicial y se tiene como resultado un grafo conexo, no dirigido G es un árbol compuesto por todos los vertices y algunas (quizá todas) de las aristas de G.

Informalmente, un árbol de expansión de G es una selección de aristas de G que forman un árbol que cubre todos los vértices.

B. Algoritmo Fuerza Bruta

La solucion 2 se realiza apartir de una busqueda de todos los posibles casos de un MST del grafo hasta que se llega a uno que cumpla con los requisitos minimos de aristas rojas y azules, en resumidas cuentas se implementa un algoritmo de fuerza bruba el cual:

- Genera todos los grafos posibles con todos los nodos
- Analiza si este tiene ciclos (si no, entonces es un MST)
- Revisa si cumple con el numero de aristas rojas definida por el archivo
- Retorna la solucion

Esta misma solucion se tiene una complejidad de O(2ˆn)

C. Algoritmo de Prim

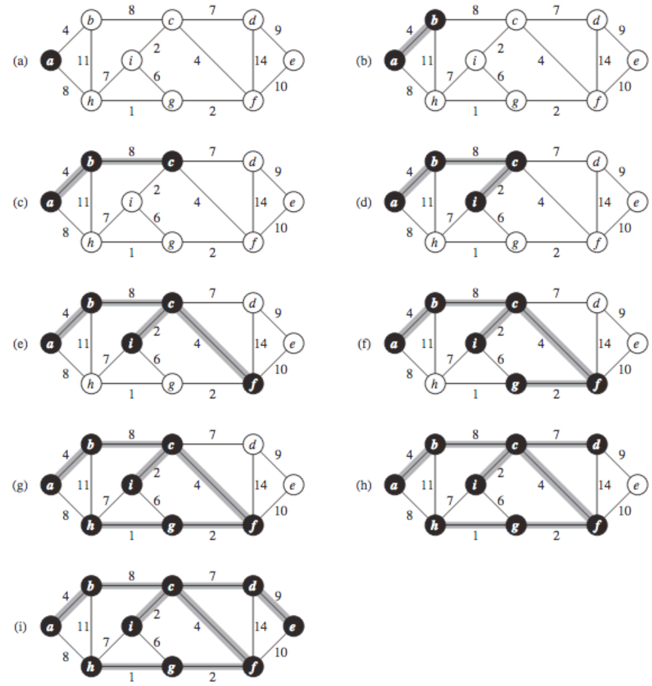
El algoritmo de Primm es un algoritmo que trabaja muy similarmente como trabaja Dijkstra para encontrar los caminos mas cortos en un grafo . El algoritmo de Primm tiene la propiedad de que siempre todas las aristas en el conjunto A forman un único Árbol, se comienza desde un vértice arbitrario y de este va creciendo el árbol de expansión mínima

```
MST-PRIM(G, w, r)
1  for each u ∈ G.V
2      u.key = ∞
3      u.π = NIL
4  r.key = 0
5  Q = G.V
6  while Q ≠ ∅
7      u = EXTRACT-MIN(Q)
8      for each v ∈ G.Adj[u]
9          if v ∈ Q and w(u, v) < v.key
10             v.π = u
11             v.key = w(u, v)
```

Para implementar eficientemente el algoritmo de Prim, necesitamos una forma rápida de seleccionar una nueva arista para añadir al árbol formado por las aristas en A. En el pseudocódigo, el grafo conectado G y la raíz r del árbol de expansión mínimo son entradas para el algoritmo. Durante la ejecución del algoritmo, todos los vértices que no están en el árbol residen en una cola de prioridad mínima Q basada en un atributo clave. Para cada vértice, el atributo v es el peso mínimo de cualquier arista que se conecta a un vértice en el árbol; Por convención, v = infinito si no existe tal borde.

Este es el algoritmos que se implementa para solucionar el problema de la expansión mínima con Primm que se describe paso a paso como el árbol se va creando partir de una regla especifica.

Aquellos vértices que tienen caminos libres y que estén ya dentro del conjunto A, se revisan para ver cual es el de menor peso y se escoge ese camino para agregar otro vértice y así comentar de nuevo el proceso, este paso a paso de describe mejor en la siguiente ilustración



el tiempo de ejecución del algoritmo de primm depende al igual que Kruskal de la implementación pero esta vez de la prioridad de la cola Q, que si la implementamos como una binary min-heap, el tiempo ocupado para realizar las líneas 1-5 es de O(V) , EXTRACT-MIN es de O(Vlg V) y para las líneas 8-11 el ciclo tarda O(V lg V) para resumir el total de esta operación del algoritmo de primm es un tiempo de ejecución de O(R lg V) el cual es asintóticamente el mismo que el de Kruskal

Entrada

La entrada consiste en multiples casos de prueba. Cada caso de prueba consiste en una linea con los enteros P, K donde P representa el numero de localizaciones y K el numero de tiendas a construir. Seguido por P lineas con tres valores: x, y, d donde x y y representa la posicion de la ubicación Pi de la forma Pi = (x, y) y d representa la demanda del sitio Pi . Luego va un enter k que representa el costo de construir cada tienda. 1.4. Salida La salida consiste en en un conjunto de parejas ordenadas de la forma (x, y) donde x y y representan la posicion de la tienda que se construye.

Sample Input.

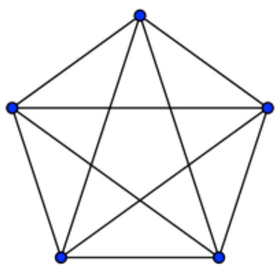
4 1
20 70 50
30 40 30
90 30 70
50 100 60
100

Sample Output.

(50, 70)

III. Cliques en Grafos

En Teoría de grafos un clique en un grafo G es un conjunto de vértices V tal que para todo par de vértices de V, existe una arista que las conecta. En otras palabras, un clique es un sub-grafo en que cada vértice está conectado a cada otro vértice del sub-grafo, es decir, todos los vértices del sub-grafo son adyacentes. Esto equivale a decir que el sub-grafo inducido por V es un Grafo completo. El tamaño de un clique es el número de vértices que contiene.



Un clique es un grafo dirigido G=(V,E) tal que V, C V y todos los pares de vértices esta conectados, en otras palabras un clique es un sub-grafo completo de G
Este problema del clique es un problema NP-Completo

A. Camino Hamiltoniano – Ciclo Hamiltoniano

Un grafo G tiene un ciclo de Hamilton utilizando la arista uv y sólo si el grafo G es obtenido por G mediante la sustitución de la arista por un par de vértices de grado 1, uno conectado a u y el otro conectado a v, tiene un camino de Hamilton. Por lo tanto, al tratar esta sustitución para todas las aristas incidentes hasta cierto vértice seleccionado de G, el problema del ciclo Hamiltoniano puede ser resuelto como máximo por V cálculos en la mayoría de los caminos Hamiltonianos, donde V es el número de vértices en el grafo.

B. El problema

El problema del vendedor ambulante (TSP) plantea la siguiente pregunta: "Dada una lista de ciudades y las distancias entre cada par de ciudades, ¿cuál es la ruta más corta posible que visita cada ciudad exactamente una vez y regresa a la ciudad de origen?"

C. Sample Input.

4 2

D. Sample Output.

8
['0111', '0001', '0010', '0100', '1110', '1101', '1011', '1000']

References

R. L. Graham and Pavol Hell. "On the History of the Minimum Spanning Tree Problem". 1985.

https://www.ingenieriaindustrialonline.com/herramientas-para-el-ingeniero-industrial/investigaci%C3%B3n-de-operaciones/teor%C3%ADa-de-redes/

1. **Cormen, T.; Leiserson, Ch.; Rivest, R., (2009). Introduction to Algorithms, The MIT Press.**
2. *Michael R. Garey and David S. Johnson (1979), Computers and Intractability: A Guide to the Theory of NP-Completeness, W.H. Freeman,*

Basic format for books:
J. K. Author, "Title of chapter in the book," in *Title of His Published Book, x*th ed. City of Publisher, (only U.S. State), Country: Abbrev. of Publisher, year, ch. *x*, sec. *x*, pp. *xxx–xxx.*
Examples:

[1] G. O. Young, "Synthetic structure of industrial plastics," in *Plastics,* 2nd ed., vol. 3, J. Peters, Ed. New York, NY, USA: McGraw-Hill, 1964, pp. 15–64.
[2] W.-K. Chen, *Linear Networks and Systems*. Belmont, CA, USA: Wadsworth, 1993, pp. 123–135.

Basic format for books (when available online):
J. K. Author, "Title of chapter in the book," in *Title of Published Book*, *x*th ed. City of Publisher, State, Country: Abbrev. of Publisher, year, ch. *x*, sec. *x*, pp. *xxx–xxx*. [Online]. Available: http://www.web.com
Examples:

[3] G. O. Young, "Synthetic structure of industrial plastics," in Plastics, vol. 3, Polymers of Hexadromicon, J. Peters, Ed., 2nd ed. New York, NY, USA: McGraw-Hill, 1964, pp. 15-64. [Online]. Available: http://www.bookref.com.
[4] *The Founders' Constitution*, Philip B. Kurland and Ralph Lerner, eds., Chicago, IL, USA: Univ. Chicago Press, 1987. [Online]. Available: http://press-pubs.uchicago.edu/founders/
[5] The Terahertz Wave eBook. ZOmega Terahertz Corp., 2014. [Online]. Available: http://dl.z-thz.com/eBook/zomega_ebook_pdf_1206_sr.pdf. Accessed on: May 19, 2014.
[6] Philip B. Kurland and Ralph Lerner, eds., *The Founders' Constitution.* Chicago, IL, USA: Univ. of Chicago Press, 1987, Accessed on: Feb. 28, 2010, [Online] Available: http://press-pubs.uchicago.edu/founders/