

Modelado de flujo máximo de Aguas Residuales

Maximum flow mode of wastewater

Yolanda Moyao Martínez, Gerardo Flores Petlascalco, Meliza Contreras González ,
Beatriz Beltrán Martínez , Pedro Bello López, Miguel Rodríguez Hernández

Facultad de Ciencias de la Computación, Benemérita Universidad Autónoma de Puebla,
Av. San Claudio y 14 Sur, Cd Universitaria, Puebla, Puebla, México, C.P. 72592.

Correo-e: {junio8834, gfloresp93, vikax68, bbeltranmtz, pb5pbello, mikelrodriguez} @gmail.com

PALABRAS CLAVE:

Flujo máximo, algoritmo
Ford-Fulkerson, aguas
residuales.

RESUMEN

En las grandes ciudades un problema es el desalojo de sus aguas residuales por parte de su sistema de drenaje. En muchas ocasiones esta tan mal implementado o es tan obsoleto que no permite el correcto desalojo de los fluentes, es aquí donde soluciones computacionales pueden dar una puerta de escape a este tipo de problemas con muy poco esfuerzo, basta conocer de forma precisa las capacidades de las tuberías de toda su red de drenajes y la ubicación de los vertederos o tuberías con mayor capacidad más próximos. Conociendo lo anterior, se puede crear un algoritmo que permita dirigir esas aguas residuales estancadas a un lugar en el menor tiempo posible por conductos que tendrán limitaciones o capacidades. En esta investigación se implementó computacionalmente el algoritmo "Ford.-Fulkerson" de flujo máximo y se harán algunas observaciones para ver si este es un buen candidato para dar solución al problema expuesto anteriormente.

KEYWORDS:

maximum flow, Ford-
Fulkerson Algorithm,
wastewater

ABSTRACT

In the big cities a problem is the evacuation of its residual waters on the part of its system of drainage. In many occasions it is so badly implemented or it is so obsolete that it does not allow the correct evacuation of the fluents, it is here where computational solutions can give an escape door to this type of problems with very little effort, it is enough to know in a precise way the capacities of the pipes of all your drainage network and the location of the nearest spillways or pipes with greater capacity. Knowing the above, it is possible to create an algorithm that allows directing these stagnant wastewater to a place in the shortest possible time by conduits that will have limitations or capacities. In this investigation the algorithm "Ford.-Fulkerson" of maximum flow was implemented computationally and some observations will be made to see if this is a good candidate to give solution to the problem previously exposed.

Recibido: 07 de agosto de 2018 Aceptado: 20 de mayo de 2019 Publicado en línea: 28 junio de 2019

1. Introducción

En teoría de grafos, un grafo dirigido con pesos también se le conoce con el nombre de red, donde cada arista representa canales o tuberías por los que puede circular algo y sus pesos son la cantidad máxima de tráfico que puede soportar tal canal, este flujo máximo puede o no alcanzarse, la única restricción es que no se supere.

El problema del flujo máximo en una red, consiste en lo siguiente: dado un grafo dirigido con pesos, encontrar la cantidad máxima de flujo que puede circular desde un punto inicial S hasta un punto final T, pasando por nodos intermedios que tendrán dos tipos de flujos, positivos si es que de los nodos salen flujos y negativos si es que entran flujos al nodo. S tiene como restricción de que siempre tendrá solo flujo positivo y T siempre tendrá solo flujo negativo.[1]

Para solucionar este problema de flujo máximo, se han propuesto diversos algoritmos, uno muy conocido y el que se estudiará es el de “Ford-Fulkerson” propuesto en 1956 por Lester Radolph Ford Jr. y Delbert Ray Fulkerson.

El documento está organizado de la siguiente forma. En la sección 2 se presenta el algoritmo de Ford-Fulkerson. En la sección 3 se muestra su implementación. Finalmente en la sección 4 se brindan conclusiones sobre el trabajo.

2. Algoritmo de Ford-Fulkerson

Este algoritmo puede usarse para resolver modelos de transporte de mercancías, flujo de gases y en nuestro caso, flujo de líquidos. Además permite que existan diversas fuentes y sumideros a lo largo de la red, pero el problema puede simplificarse considerando únicamente una fuente y sumidero en común, este algoritmo depende de tres conceptos principales [2]:

Un camino de aumento, es una trayectoria desde el nodo fuente S hasta el nodo sumidero T que puede conducir más flujo.

La capacidad residual es la capacidad adicional de flujo que un arco partiendo de un nodo u hasta un nodo v puede llevar: $Cf(u,v)=c(u,v)-f(v,u)$

Teorema de Ford-Fulkerson: En cualquier red, el flujo máximo que fluye de la fuente al destino es igual a la capacidad de corte mínimo que separa a la fuente del destino.

El algoritmo se muestra a continuación:

Algoritmo Ford-Fulkerson

Ford-Fulkerson (G, S, T)

Inicializar flujo $f(u, v) = 0$ para cada arco (u, v)

Mientras (exista un camino de aumento p desde S hasta T) hacer

1.-

$Cf(p) = \min\{Cf(u, v): (u, v) \text{ está sobre } p\}$

2.- Para cada arco (u, v) está sobre p

$f(u, v) = f(u, v) + Cf(p)$

$f(v, u) = f(v, u) - Cf(p)$

3. Implementación

Para realizar la implementación del algoritmo se usó Python como lenguaje de programación junto con la librería Graphviz para dibujar los grafos que se van obteniendo con las iteraciones. Para meter los datos de los grafos al programa es por medio de un archivo de texto que contiene la siguiente estructura:

1. En el primer renglón se especifica el número de nodos
2. Abajo del número de nodos, se introducen las tulpas respetando la siguiente forma: `Nodo_Origen` `Nodo_Destino` `Flujo`, como se muestra en la Figura 1.

```

ejemplo: Bloc de notas
Archivo Edición Formato Ver
5
<- Número de nodos
1 4 10
1 3 30
1 2 20
2 5 30
2 3 40
3 4 10
4 3 5
3 5 20
4 5 20
    
```

Origen Destino Flujo

Fig. 1. Ejemplo de archivo de texto para procesarlo en python.

Con la información obtenida se podrá crear la matriz de adyacencias que es la parte fundamental para el funcionamiento del algoritmo. Esta matriz tendrá en sus intersecciones únicamente los flujos positivos y estos se actualizarán de forma automática cada vez que haya una iteración. Además servirán para poder crear los grafos que nos vayan mostrando la evolución de una red cuando sus flujos se van modificando y por último servirán de punto de partida al algoritmo “Búsqueda a lo

profundo” para determinar si aún hay caminos que vayan de S a T que permitan iterar una vez más para aumentar el flujo o cortarlo si ya no existe más caminos, un grafo de ejemplo se muestra en la Figura 2.

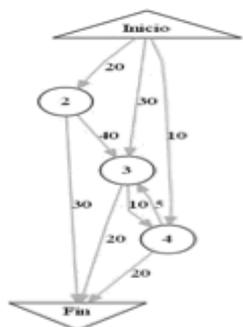


Fig. 2. Grafo de ejemplo.

Para auxiliar en la búsqueda de caminos, se usa el algoritmo “Búsqueda a lo profundo” donde su única tarea será informar si aún hay caminos de S a T, en caso de que ya no haya ninguno simplemente envía una señal de falso y el programa termina. Para funcionar hace uso de la matriz de adyacencias y un arreglo de nodos visitados. De forma general la implementación queda de la siguiente manera, se usaron dos matrices, la primera es la matriz de fuentes, donde únicamente serán flujos positivos y es la que se usará para crear los caminos p en las iteraciones, la otra matriz, llamada residual, contendrá los flujos negativos de cada nodo, ninguna tendrá valores estáticos pues se irán actualizando constantemente.

Las pruebas fueron hechas usando el grafo de la Figura 3, en esta parte el criterio para ir eligiendo el camino p de los nodos fue elegir el camino de (u,v) que tenga el flujo positivo más alto, el nodo fuente S será el 1, mientras que el nodo vertedero T será el 4, el resultado final se muestra en la Figura 4.

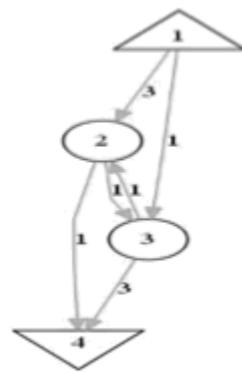


Fig. 3. Grafo de prueba.

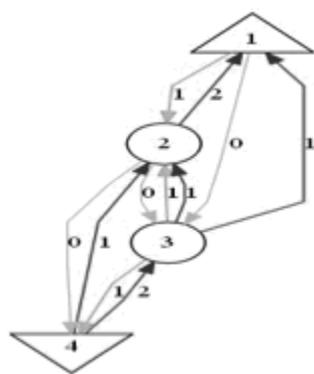


Fig. 4. Grafo final con la aplicación del algoritmo.

4. Conclusiones

En la prueba realizada con el grafo anterior, los resultados arrojan que existe una diferencia de flujos máximos dependiendo del criterio de selección de caminos usados, aunque si aplicamos el algoritmo usando el criterio del mínimo en una red de drenaje para Aguas Residuales afectaría el rendimiento y por ende no se recomienda usar. Otra diferencia es en la cantidad de Iteraciones que se efectúan, pero depende de la red modelada pues algunos casos usando el máximo serán menos iteraciones que usando el mínimo, pueden ser iguales o el mínimo tenga más iteraciones que el máximo.

El algoritmo “Ford-Fulkerson” para flujo máximo en una red, es una muy buena alternativa para mejorar el flujo de Aguas residuales en una Red, sin embargo se requiere tener en cuenta estas recomendaciones:

- Se debe conocer la capacidad de cada arista de la red y las ubicaciones de los vertederos.
- Usar el criterio del máximo para elegir el camino ya que en algunos casos si se elige el

mínimo puede verse afectado el rendimiento de la red.

Referencias

1. García Mateos, G.; Giménez Cánovas, D. *Algoritmos y Estructura de Datos*. Murcia: Diego Marin. 2003
2. Taha, H., *Investigación de operaciones*. Pearson. 2010.

Semblanza de los autores



M.C. Yolanda Moyao Martínez es egresada de la Maestría en Ciencias de la Computación, actualmente es un estudiante del programa de doctorado LKE, su principal área de investigación es la optimización combinatoria y la generación de algoritmos de conteo.



Lic. Gerardo Flores Petlacalco actualmente realiza estudios en la Ingeniería en Ciencias de la Computación, sus temas de interés son las aplicaciones móviles, web y programación en redes sociales y el procesamiento de lenguaje natural.



M.C. Meliza Contreras González realizó su Maestría en Ciencias de la Computación en la BUAP, en el área de computación matemática, actualmente realiza estudios en el Doctorado en Ingeniería del Lenguaje y del Conocimiento, sus temas de interés son el procesamiento del lenguaje natural, la economía del comportamiento, las teorías de aprendizaje y los procesos de razonamiento.



M.C. Beatriz Beltrán Martínez realizó su Maestría en Ciencias de la Computación en la BUAP, en el área de reconocimiento de patrones, actualmente realiza estudios en el Doctorado en Ingeniería del Lenguaje y del Conocimiento, trabaja principalmente en el área de recuperación de información, procesamiento de lenguaje natural y reconocimiento de patrones, En estas líneas tiene diferentes publicaciones.



M.C. Pedro Bello López es egresado de la Maestría en Ciencias de la Computación, actualmente es un estudiante del programa de doctorado LKE, su principal área de investigación es la teoría de grafos y la revisión de creencias.



M.C. Yolanda Moyao Martínez es egresada de la Maestría en Ciencias de la Computación, actualmente es un estudiante del programa de doctorado LKE, su principal área de investigación es la optimización combinatoria y la