Analisis de Algoritmos

Solución al problema de Demanda Monetaria por una reducción al algoritmo de Ford-Fulkerson en lenguaje de Programación Java

Mario Alexis Guzmán Mosco

11 de junio de 2019

1. Problema de Circulación Monetaria

En el mundo, cada día se realiza una inmensa cantidad de transacciones monetarias ya sea con dinero físico (billetes, monedas, etc) o con dinero líquido. La cantidad y la demanda de dinero que una nación posee, al menos el los paises de occidente, es controlada por un banco central o una entidad gubernamental que funcione como equivalente, por ejemplo, en los Estados Unidos es la Reserva Fereal la encargada de este proceso.

Sin embargo, son los bancos comerciales los encargados de controlar el flujo de dinero disponible dentro de una nación y fuera de ésta cuando se relacionan los paises. Los bacons comerciales funcionan como instituciones financieras que retienden dinero a travéz de depositos de clientes, cuentas de negocios, entre otros servicios. Desde una perspectivas más amplia, sus rol en la cadena de difusión de dinero es ofrecer a los clientes crédito con el fin de ayudar a individuos a hacer grandes o pequeños retiros de dinero.

La reserva federal puede influeciar el abasto de dinero mendiante el control de bancos comerciales cambiando la cantidad de dinero que poseen. Las reservas de dinero indican la cantidad de dinero que un banco comercial debe retener en lugar de prestar. Las tasas bajas aumentan la oferta monetaria, mientras que las tasas altas disminuyen la oferta monetaria.

La cantidad de dinero que deben de poseer los bancos es de gran importancia en la economía pued puede afectar directamente la inflación. La inflación se define de manera çlásicaçomo demasiado dinero disponible para obtener muy pocos bienes. La estrecha oferta de dinero puede limitar la cantidad de negocios que las personas y las empresas pueden llevar a cabo en el mercado económico.

El problema de la circulación monetaria consiste que en una cantidad limitada de bancos, cada uno con una oferta y una demanda monetaria especifica, tratar de encontrar una forma de distribuir el dinero emitido y autorizado por el Estado (en este caso, una reserva federal) de tal manera que el dinero se encuentre optimamente distribuido por los diferentes bancos para evitar escaces o exceso de dinero.

En este proyecto, no se tratará de encontrar una red optima para el problema de circulación monetaria, en su lugar, se intentara aplicar un algoritmo que nos diga si es posible, dado ciertos bancos con cierta demanda u oferta de dinero, formar una red de flujo de dinero para satisfacer las necesidades de dinero en todos los puntos de la red, es decir, se trata de un problema de desición.

2. Circulación con demanda

Una red de flujo se define como una gráfica especial $G = (V, E)$ tal que $\forall e \in E, f(e) \geq 0$ en donde las función $f(e)$ representa el flujo de la arista e. Además, se tiene que hay dos vertices distinguidos s y t, cada vertice de G pertence a alguna ruta de s a t.

El problema de flujo máximo nos indica sobre cual es la mayor cantidad de flujo que se puede transportar de s a t sin violar las restricciones de capacidad, es decir:

- $\forall (u,v) \in V, \quad f(u,v) \leq c(u,v)$
- $f(u,v) = -f(v,u)$
- $\forall u \in V - \{s,t\} \quad \sum_{v \in V} f(u,v) = 0$

El problema de circulación bajo demanda es una modificación al problema original del flujo máximo, en lugar de tener dos vertices distinguidos s y t, puede haber un conjunto finito de fuentes s, y de pozos t. En este caso, los bancos con demanda positiva de dinero actuarian como los vertices del conjunto de pozos, y los de demanda negativa como el conjunto de fuentes.

3. Reducción al Algoritmo de Ford-Fulkerson

El algoritmo de Ford-Fulkerson se propone resolver el problema del flujo máximo tomando como parametros la red de flujos, el vertice fuente, y el vertice pozo.

Algorithm Ford–Fulkerson

Inputs Given a Network $G = (V, E)$ with flow capacity *c*, a source node *s*, and a sink node *t*
Output Compute a flow *f* from *s* to *t* of maximum value

1. $f(u,v) \leftarrow 0$ for all edges (u,v)
2. While there is a path *p* from *s* to *t* in G_f, such that $c_f(u,v) > 0$ for all edges $(u,v) \in p$:
    1. Find $c_f(p) = \min\{c_f(u,v) : (u,v) \in p\}$
    2. For each edge $(u,v) \in p$
        1. $f(u,v) \leftarrow f(u,v) + c_f(p)$ (*Send flow along the path*)
        2. $f(v,u) \leftarrow f(v,u) - c_f(p)$ (*The flow might be "returned" later*)

Figura 1: Algoritmo de Ford-Fulkerson en pseudo-código

Sin embargo, es posible resolver la decibilidad de la existencia de una circulación bajo demanda a trávez de este algoritmo. La reducción consiste en crear un nodo s, y un nodo t que funcionen como fuente y pozo de la red. s se conecta a todos los vertices con capacidad negativa, mientras que t a los de capacidad positiva. Esto, evidentemente forma una Red de flujo que tiene asociada un flujo máximo que va de s a t.

4. Programación en Java

El proposito de esta implementación es el apovechar el resultado que indica que la suma de las capacidades en una red de flujo con demanda G es igual al flujo máximo en una Red G' construida a partir de G. Esta implementación recoge desde la línea de comandos los vertices y las aritas de la gráfica G.

Cada vertice en la gráfica representa un banco mientras que la demanda pordría representar en un entero, la demanda o la oferta en millones que el banco nececita, mientras que el flujo de las aristas representa la cantidad con la que estan autorizados a pagar de uno a otro.

4.1. Compilación y entrada:

Compilación desde la terminal: situarse en el directorio del proyecto y ejecutar
javac *.java
ejecución: java Main

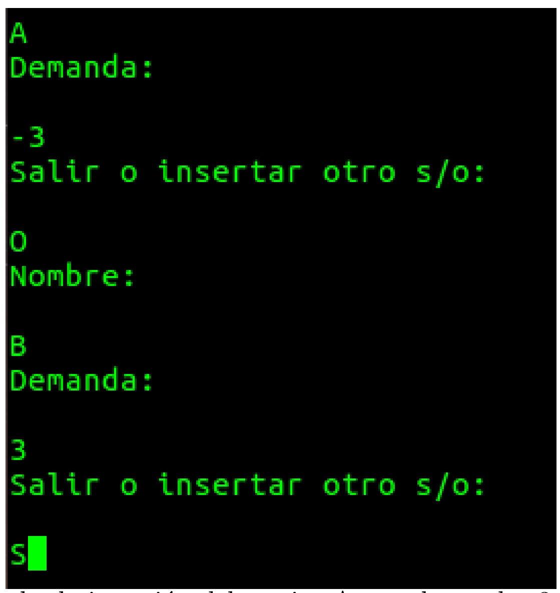

Figura 2: Ejemplo de inserción del vertice A con demanda -3 y vertice B con demanda 3

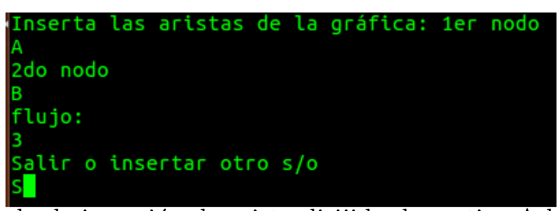

Figura 3: Ejemplo de inserción de arista dirijida de vertica A hacía B con flujo 3

5. Fuentes bibliograficas:

Kleingberg,Jon y Tardos, Eva. (2006). Algorithm Desing (2ª ed.). Boston, Estados Unidos: Pearson.

Mankiw, N. Gregory. Principios de Economía. Mc Graw Hill.