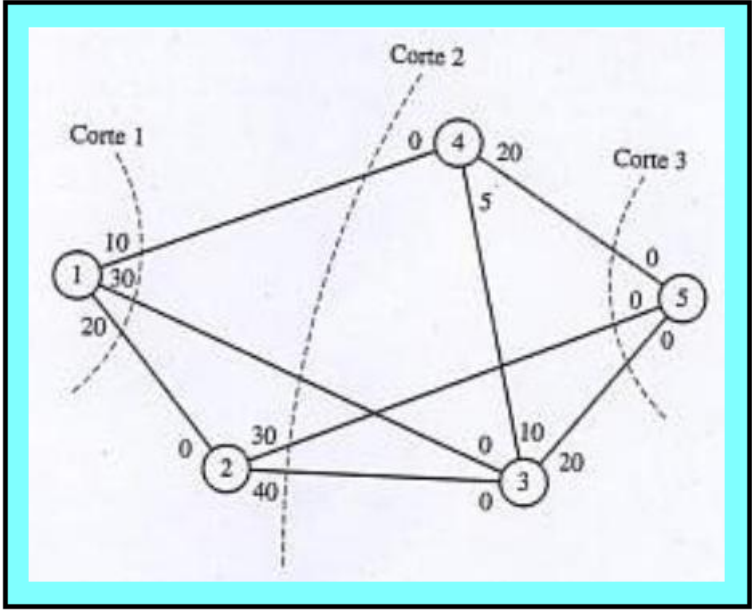


DESARROLLO TEÓRICO.
PROBLEMAS DE MÁXIMO FLUJO



Trabajo realizado por Ballester Matito, Manuel
Molina De Castro, María Elena
Moreno López, Manuel.

1. Problemas de Programación Lineal en Redes

En la introducción hemos dado un repaso de la teoría de grafos o redes, y hemos explicado que nuestro trabajo versa de modelos de optimización sobre estas redes. Hay diversos problemas de que se pueden plantear de sobre redes, los principales son los siguientes:

- Problema de Flujo a Coste Mínimo
- Problema de Camino Mínimo
- Problema de Transporte
- Problema de Máximo Flujo
- No obstante, hay muchos más, como el de árbol de mínima de expansión, o el de optimización de trueques entre tiempo y costo de un proyecto.

En la introducción también acercamos al lector a estos problemas de forma intuitiva y con algunos ejemplos. Ahora en este apartado teórico los definiremos formalmente y los formularemos.

Nos daremos cuenta de que la mayoría de los problemas de redes son casos particulares del Problema de Flujo a Coste Mínimo. Por ello, prestaremos más atención a

A modo de esquema de lo que pretendemos estudiar en este trabajo, podemos hacer la siguiente agrupación por secciones:

- Empezaremos con el Problema de Flujo a Coste Mínimo. En cada subapartado estudiaremos los problemas que se derivan de él. Para cada problema, vamos a dar el objetivo principal, la formulación, y resolveremos un ejemplo de forma intuitiva (para saber interpretar los resultados). Obviamente, no nos centraremos en los modos de resolución de estos problemas, ya que no es el objeto del presente trabajo.
- Ahora sí que estudiaremos la resolución, por el método símplex, implementado en el ordenador, del Problema de Máximo Flujo. Posteriormente lo aplicaremos a un ejemplo, y más tarde veremos las limitaciones de nuestro programa, el cual hemos realizado en R.
- Por último, explicaremos un modo de solucionar los Problemas de Máximo Flujo a mano. Y es que, puesto que resulta bastante lioso trabajar con las tablas del símplex en este tipo de problemas, se ha ideado un algoritmo alternativo. Este servirá para resolver a mano este tipo de problemas, de forma relativamente rápida.

2. Problemas de Flujo a Coste Mínimo

2.1. Modelización y formulación

Ese caso que acabamos de analizar, se modela como un problema de flujo a coste mínimo. En general, son problemas de programación lineal que presentan las siguientes características:

- Tenemos al menos, un nodo origen o fuente, y un nodo destino. El resto pueden considerarse de transbordo.
- Trabajamos sobre un grafo dirigido $G = (V, A)$ (con arcos, no aristas), y que además es conexo (de otra forma, no podríamos enviar el flujo de un lado a otro). Evidentemente, el flujo solamente puede ser enviado en el sentido que indican los arcos.
- Cada uno de los arcos del grafo tiene una capacidad máxima de flujo que puede circular a través de él, y tiene asociado un coste.
- Cada uno de los nodos puede tener una oferta y/o una demanda. Si por ejemplo, tuviera una demanda 3 unidades y oferta 2, a ese nodo debe llegarle 3 unidades, y de él deben salir 2. Por tanto, hay 1 unidad que debe permanecer en el nodo.
- El objetivo es minimizar el coste total de enviar el flujo por la red para satisfacer las demandas.

Ahora vamos a tratar de hacer la formulación del mismo:

$$\begin{aligned} \min \quad & \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij} \\ \text{s.a.} \quad & \sum_{j/(i,j) \in A} x_{ij} - \sum_{j/(j,i) \in A} x_{ji} = b_i, \quad i \in \{1, \dots, n\} \\ & 0 \leq x_{ij} \leq u_{ij} \end{aligned}$$

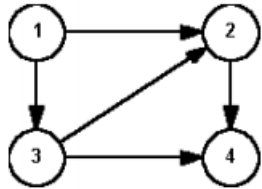
Explicuemos esta formulación:

- Las variables x_{ij} representan la cantidad que pasa por el arco (i, j) . Entonces, la función objetivo busca minimizar la suma de todas esas variables, asignándole a cada una el peso de su costo c_{ij} .
- La primera restricción corresponde con que, para cada nodo $i \in \{1, \dots, n\}$, la suma de todo lo que sale de ese arco, menos todo lo que entra, debe ser igual a lo que se queda en el nodo, b_i . Este valor corresponderá con la oferta menos la demanda, i.e. $b_i = o_i - d_i$.
- La última restricción corresponde con la no negatividad de las variables. Además, cada variable $x_{i,j}$ corresponde con la cantidad de flujo que circula por el arco (i, j) , y ya dijimos que ese arco tendrá una capacidad máxima, la cual estamos llamando $u_{i,j}$.

2.2. Ejemplo

Consideremos el siguiente problema:

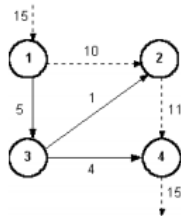
Se deben transportar 15 toneladas de plástico en bola desde la ciudad 1 hasta la 4. Los distintos caminos que tenemos son los siguientes:



Además, disponemos de cierta información recogida en una tabla que mostraremos debajo. En ella, aparece recogido el costo que supone llevar una tonelada por cada camino, y la cantidad mínima y máxima de mercancía que debe circular a través de esos arcos.

Arco(i,j)	Costo Unitario	Flujo Mínimo	Flujo Máximo
(1,2)	3	0	10
(1,3)	2	1	13
(3,2)	3	0	9
(2,4)	5	4	11
(3,4)	6	2	12

Por simple inspección llegamos a una solución factible:



Para resolver el problema podríamos aplicar el algoritmo símplex, implementado en una computadora (como la que aparece en el moodle), o bien aplicar una simplificación del simplex que funciona en estos casos, para hacerlo a mano de forma menos costosa.

A continuación, vamos a ver casos particulares del Problema de Flujo a Coste Mínimo. Estos problemas tienen nombre propio (Camino Mínimo, Transporte,...) debido a la capital importancia de los mismos. Cabe destacar también que no son todos los problemas de redes que existen. No obstante, presentaremos los más famosos.

(*) Otros problemas también interesantes serían, por ejemplo, los mencionados en la Introducción, como el problema del árbol de expansión mínima, o los modelos de optimización de trueques entre tiempo y costo de un proyecto.

2.3. Problemas de Camino Mínimo

Tenemos n lugares en el mapa, que corresponden con los nodos, y habrá ciertos arcos (i, j) que los conecten, luego representan los caminos que podemos coger. Cada uno de estos arcos tendrá asociado su longitud c_{ij} . En este tipo de problemas, supondremos que partimos de un nodo origen, que sin pérdida de generalidad es el primero (1), y un nodo destino, que será el último (n). Supondremos también que al nodo origen no llega ningún arco, y de nodo destino no sale ninguno.

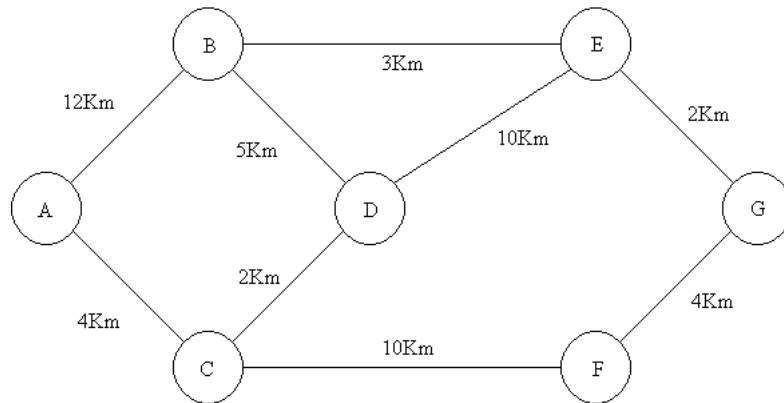
Por tanto, la formulación será:

$$\begin{aligned} \min \quad & \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij} \\ \text{s.a.} \quad & \sum_{j/(i,j) \in A} x_{ij} - \sum_{j/(j,i) \in A} x_{ji} = b_i = 0, \quad i \in \{2, \dots, n-1\} \\ & \sum_{j/(1,j) \in A} x_{1j} = 1, \quad i = 1 \text{ (lo que sale del origen)} \\ & - \sum_{j/(j,n) \in A} x_{jn} = -1, \quad i = n \text{ (lo que entra en el origen)} \\ & x_{ij} \in \{0, 1\} \end{aligned}$$

Vamos a explicar esta formulación:

- Las variables x_{ij} son binarias, e indican valdrá 1 si pasamos por el arco (i, j) y cero en caso contrario. Los costos c_{ij} representan la longitud de ese camino, así que nuestra función objetivo minimiza el camino a tomar.
- La primera restricción corresponde con que, para cada nodo $i \in \{2, \dots, n-1\}$ (ni el origen ni el destino), el número de veces que entramos y salimos de ese arco deben ser las mismas.
- Las dos siguientes restricciones expresan que lo salimos de el primer nodo, y que debemos llegar al último.
- Además, tenemos que x_{ij} son binarias.

Veamos un ejemplo:



Mapa de carreteras

Por ejemplo aquí, en un grafo con pocos nodos, como este, podemos hacer un análisis heurístico. Suponemos que tenemos que llegar de A a G . Y podemos observar que el camino más corto será:

$$A \rightarrow C \rightarrow D \rightarrow B \rightarrow E \rightarrow G$$

El valor de la función objetivo será 16 (número de kilómetros totales recorridos). Y las variables x_{ij} valdrán todas cero, salvo: $x_{AC} = x_{CD} = x_{DB} = x_{BE} = x_{EG} = 1$

2.4. Problemas de Transporte

En este tipo de problemas diferenciamos dos tipos de nodos

- Los nodos orígenes son aquellos que tienen ofertas y no tienen ninguna demanda (luego $b_i = o_i - d_i = o_i$), además no habrá ningún arco dirigido hacia estos nodos, i.e. si i es un nodo de origen,

$$\sum_{j/(j,i) \in A} x_{ji} = 0$$

- De forma similar tenemos los nodos destino, que solamente tienen demanda, en los que se cumple que $b_i = o_i - d_i = -d_i$. De este tipo de nodos, tampoco sale ningún arco, luego si i es un nodo destino,

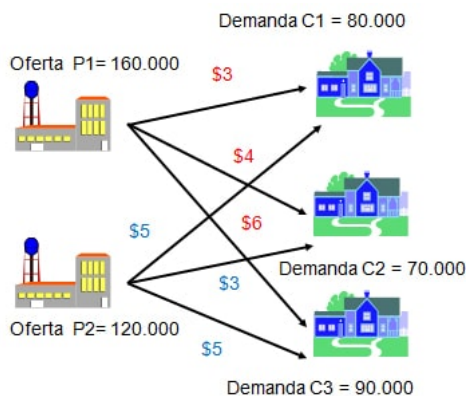
$$\sum_{j/(i,j) \in A} x_{ij} = 0$$

Este tipo de situaciones tiene sentido, por ejemplo, para empresas que deben hacer un suministro a particulares. Podríamos pensar en empresas de distribución como Amazon, que deben enviar sus productos desde ciertas centrales (nodos orígenes) hacia ciertas personas (destino).

Además, con esto que hemos dicho, la formulación nos queda como sigue:

$$\begin{aligned} \min \quad & \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij} \\ \text{s.a.} \quad & \sum_{j/(i,j) \in A} x_{ij} = o_i, \quad i \in \{1, \dots, m\} \text{ (m nodos orígenes, } m < n) \\ & -\sum_{i/(j,i) \in A} x_{ji} = -d_i, \quad i \in \{m+1, \dots, n\} \text{ (n-m nodos destino)} \\ & 0 \leq x_{ij} \leq u_{ij} \end{aligned}$$

Consideremos el siguiente caso particular: Imaginemos que tenemos dos centrales eléctricas, pertenecientes a la misma empresa, que deben suministrar energía a tres casas. Obviamente, la empresa quiere minimizar los costos de esta gestión.



Entonces, podemos obtener una solución factible (que seguramente no coincida con la óptima), como: $x_{P1,C1} = 80$, $x_{P1,C2} = 70$, $x_{P2,C3} = 90$, y el resto de variables a cero. De este modo, la primera central suministra a las dos primeras casas, y la segunda central suministra a la tercera.

2.5. Problemas de Flujo Máximo

Este es el caso que nos ocupa. Dado que ya está explicado en la sección de Introducción y Motivación, no entraremos en más detalles, pero a modo de resumen, podemos decir que en estos problemas tratamos de enviar el máximo flujo posible entre un nodo origen (1) y otro de demanda (n).

Para resolverlo como un problema de Flujo a Coste Mínimo, tendremos que hacer una ligera transformación del problema inicial: Nuestro nuevo conjunto de aristas será: $A' = A \cup \{(n, 1)\}$, donde A es el conjunto de aristas inicial, y el otro es un arco ficticio que une el nodo destino con el inicial.

- Por un lado, vamos a exigir que lo que entra en cada nodo y lo que sale sea igual. Esto se conoce como Ley de Kirchoff.

En un principio, tiene sentido considerar esta propiedad para los nodos $2, 3, \dots, n - 1$, pero no para el origen ni el destino, ya que del primero sale un cierto flujo, y al último le llega. No obstante, gracias a nuestro conjunto de arcos ampliado, cuando el flujo final F llega al nodo destino (n), regresa por nuestro arco ficticio hasta el origen (1) ese mismo flujo F . Por ello, también se cumple la Ley de Kirchoff en los nodos origen y destino.

- Por otro lado, buscaremos maximizar el flujo de nuestro arco ficticio, es decir, el que pasa desde el nodo destino (n) hasta el (1). Como hemos impuesto la Ley de Kirchoff, este será el mismo que pase a través de cierto camino (a determinar) desde (1) hasta (n)

Por ello, en nuestra función objetivo, x_{n1} será la única variable a la que le daremos peso. Luego nuestra función objetivo sería, $max.x_{n1}$. No obstante, como el problema de Flujo a Coste Mínimo es de minimizar, lo expresaremos de forma equivalente como $min - x_{n1}$.

Con esto que hemos comentado, la formulación del problema nos queda como:

$$\begin{aligned} \min \quad & -x_{n1} \\ \text{s.a.} \quad & \sum_{j/(i,j) \in A} x_{ij} - \sum_{j/(j,i) \in A} x_{ji} = 0, \quad i \in \{1, \dots, n\} \\ & 0 \leq x_{ij} \leq u_{ij} \end{aligned}$$

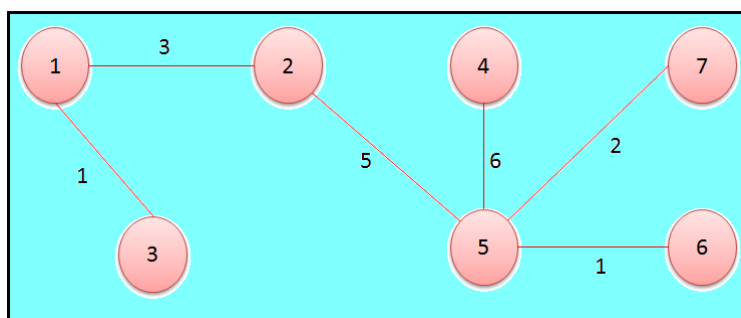
- En la primera restricción exigimos la Ley de Kirchoff. Para ello, tomamos $b_i = 0$, para todos los nodos $i \in \{1, 2, \dots, n\}$.
- La segunda restricción tiene que ver con la capacidad. Como en los casos anteriores, cada arco (i, j) tienen una capacidad u_{ij} , y la cantidad que pase a través de ellos debe ser positiva. Cabe destacar que nuestro arco ficticio no debe limitar la solución del problema. Por ello, siempre tomaremos como capacidad máxima de este arco un número muy grande (por ejemplo, la suma de todas las capacidades sería suficiente, pues nunca el flujo final será mayor a esta cantidad).

Al igual que en los casos anteriores, consideraremos un ejemplo trivial para que podamos ver la solución por simple inspección:

Los astronautas de la ISRO, enviados al espacio en la nave ISRO Orbital Vehicle, se plantean el siguiente dilema: Quieren despresurizar una de las cabinas lo más rápido posible.

Por ello, se plantean el problema de sacar el máximo flujo aire posible a través de ciertos conductos (nuestros arcos), que van desde dicha cabina (nodo origen), hasta una cabina que conecta con el espacio (nodo final). Cada uno de esos conductos tiene una capacidad máxima del volumen de aire que puede pasar por ella, por cada segundo.

Se preguntan qué conductos deben activar para que pase a través de ellos el máximo flujo de aire posible. Veamos el gráfico que han montado.



Evidentemente, si (1) es el nodo origen de la cabina a despresurizar, y en (7) está la cabina que conecta con el espacio, el flujo máximo de aire pasará en un segundo será de 2 unidades de volumen, y pasará por los arcos (1, 2), (2, 5), (5, 7).

En nuestro problemas, nos encontraríamos la solución $x_{12} = 2, x_{25} = 2, x_{57} = 2$, la variable asociada al nodo fictible es $x_{71} = 2$, y el resto de variables valdrían cero. Puede comprobarse fácilmente que cumplen la restricción primera restricción (Ley de Kirchoff), ya que, por ejemplo, para el nodo $i = 1, x_{12} - x_{71} = 0$. Con respecto a la segunda restricción, claramente son no negativas y no sobrepasan las capacidades máximas establecidas.

2.6. Resultados

Debemos hacer tres puntualizaciones:

- Por un lado, ¿qué pasa con el ejemplo anterior, en el que el grafo es no dirigido? Pues podemos considerar, en lugar de una arista, dos arcos, uno para cada sentido. Debemos hacerlo así porque desconocemos el sentido en el que se moverá el flujo.

Con esto, podemos ver que toda esta teoría se aplica también a los grafos no dirigidos.

- Analizando el problema genérico de Flujo a Mínimo Coste, nos podemos preguntar sobre la factibilidad del mismo (así como de todos los sub-problemas resultantes). Existe una propiedad básica que nos da una **condición necesaria** para que el problema sea factible. Esta es, que para que

un problema sea factible, debe cumplirse que $\sum_{i=1}^n b_i = 0$.

Esto quiere decir que la demanda y la oferta es la misma. Y cuando ocurre, decimos que el problema está balanceado. De todas formas, de no estarlo, siempre podemos balancearlo añadiendo un nodo ficticio más (de origen o demanda, según convenga).

(*) Cabe destacar, que en el caso particular del problema de transporte, esta condición es también suficiente. Luego basta ver que el problema está balanceado, para tener una solución. No obstante, en general, esto no es así, pues el hecho de que el problema tenga soluciones dependerá de qué arcos estén presentes en las redes y de sus capacidades.

- Otro resultado interesante es la **propiedad de las soluciones enteras**. Esta nos dice que si el problema es entero puro, la solución del mismo también lo será.

3. Profundizando en el modelo de máximo flujo

3.1. Desarrollo del algoritmo

Vamos a analizar el algoritmo exacto que hemos usado para implementar nuestro código en R.

Para resolver este problema desde R, debemos llamar al paquete LpSolve, que es el que resuelve estos problemas usando el método símplex.

Nuestro código se divide en tres secciones:

- La parte uno es para guardar los datos en un vector llamado *data*. También se introduce la información y los paquetes. No merece más atención.
- La parte dos es donde se formula el problema. Al final de la misma, llamamos a la función *lp* para que nos resuelva el PPL y guardamos la solución en la variable *sol*.
- Por último, la parte tres nos imprime la solución, por un lado, en forma de tabla, y por otro, nos dibuja el grafo resultante.

Explicaremos el punto dos, que es el que requiere más atención.

Como sabemos, los PPL se pueden expresar de forma matricial como:

$$\begin{array}{ll} \min & c \cdot x \\ \text{s.a.} & A'x = b' \\ & 0 \leq x_{ij} \end{array}$$

Donde en nuestro caso, $x^t = (x_{11}, x_{12}, \dots, x_{1h}, \dots, x_{h1}, x_{h2}, \dots, x_{hh})$. Considerando que hay h nodos, y que x_{ij} es la variable que nos dirá el flujo que tenemos que llevar por el arco (i, j) .

- Entonces, x es un vector $1 \times h^2$, luego el vector de costos c debe ser $h^2 \times 1$. Para que cumpla el modelo de Flujo Máximo explicado arriba, debemos formular $\min. -x_{h1}$, o equivalentemente, $\max. x_{h1}$. Optamos por esta segunda opción, y definimos el vector c repleto de ceros, a excepción de su componente $((h - 1) * h + 1)$ -ésima, que la rellenaremos con un uno.
- Teníamos como restricción de los PPL que $A'x = b'$. No obstante, nosotros tenemos dos restricciones en el problema de Maximo Flujo, luego debemos subdividir nuestra matriz A' y el vector columna b' en dos:
 - La primera era la Ley de Kirchoff, i.e

$$\sum_{j/(i,j) \in A} x_{ij} - \sum_{j/(j,i) \in A} x_{ji} = 0$$

para todo $i \in \{1, \dots, h\}$. Esto lo podemos expresar como en forma matricial como $Ax = a$, donde A es una matriz de h restricciones

o renglones (filas), y tantas columnas como variables, i.e h^2 . Estará formada por ceros, unos y menos unos. Además, a es un vector columna de h ceros.

Debemos tener también en consideración que en esa misma A hay que introducir el nodo ficticio (lo cual se hace, en el programa, debajo del bucle).

- La segunda restricción son las desigualdades que limitan la capacidad. En general, teníamos que $x_{ij} \leq u_{ij}$. Para lo cual hace falta una matriz Id identidad $h^2 \times h^2$ de forma que $Id \times x \leq u$, donde u es un vector columna de h^2 variables.

No obstante, la mayoría de las variables no son reales, puesto que no corresponden con arcos del problema real, luego valdrán cero. Así que parece más lógico exigir sólo la desigualdad a las m variables reales (anteriormente se ha calculado que hay m arcos en los datos). Así que nos quedamos con la forma matricial $Bx \leq u$, siendo B de dimensiones $m \times h^2$ y b un vector columna de m variables.

Ojo, debemos tener en cuenta que tenemos un arco ficticio más, luego añadimos en B y b otro renglón, correspondiente a que $x_{h1} \leq M$, para una M muy grande. En este caso, hemos tomado M como la suma de las capacidades que aparecen en el grafo. Esto lo hacemos, en el programa, fuera del bucle.

Por último, en la función lp , introducimos el objetivo (maximizar), el vector de costos c , nuestra matriz $(A')^t = [A|B]$, el vector columna $(b')^t = [a|b]$, y las restricciones (para las h primeras el igual $=$, y para las $m + 1$ siguientes la desigualdad \leq).

3.2. Resolviendo un ejemplo con el programa

Consideremos el ejemplo más típico: Tenemos una serie de torres eléctricas (que serán los nodos $G1, \dots, G6$) y ciertos cables de alta tensión (con límite de carga máxima) que permiten la circulación de la corriente entre estas torres. Los datos vienen expresados en esta hoja de cálculo de LibreOffice (en extensión .csv) como:

ejemplo1.csv - LibreOffice Calc

Archivo Editar Ver Insertar Formato Hoja Datos Herramientas V

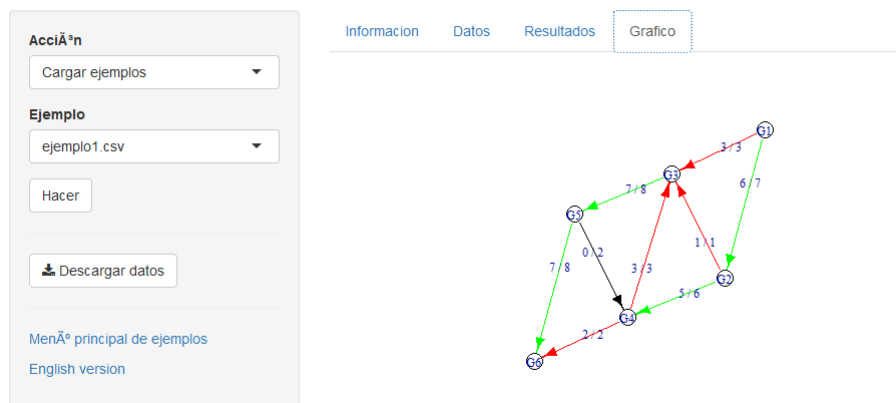
Liberation Sans 10

A1 nombre

	A	B	C	D	E	F
1	nombre	origen	destino	capacidad		
2	G1	1	2	7		
3	G2	1	3	3		
4	G3	2	4	6		
5	G4	2	3	1		
6	G5	3	5	8		
7	G6	5	4	2		
8		4	3	3		
9		4	6	2		
10		5	6	8		
11						
12						

Aplicamos nuestro programa, y obtenemos como resultado el grafo siguiente:

Maximun Flow



(**) Más adelante mostramos también la solución en forma de tabla.

3.3. Limitaciones del programa en R

Como todos los programas, este también tiene sus limitaciones. Es por tanto importante conocerlas para evitar soluciones erróneas, o incluso pensar que no funciona correctamente. Vamos a enumerar algunas de ellas, que podrán incluso visualizarse con el ejemplo, de forma que el usuario pueda identificarlas:

- La primera de ellas viene explicada en la información del programa. Se trata de que consideraremos que el nodo origen del grafo es el origen del primer arco que se introduce en la tabla. Así como el nodo destino es el

destino del último arco que se introduce.

Puede acarrear un problema si el usuario, después de meter los datos y obtener la solución, añade otro arco más a los datos, de forma que en este nuevo y último arco, su destino no corresponda realmente con el nodo destino del grafo. Esto podría llevar a problemas. Así que se recomienda que, aunque se introzcan o borren nuevos datos, que se organicen de forma que en el último arco que aparezca en las datos, el destino coincida con el destino del grafo en nuestro problema.

- Otro problema suele darse cuando ponemos como nombre de los nodos, en lugar de caracteres, números. Por eso, si los nodos son 1, 2, 3, ..., como en el caso anterior, se recomienda usar los nombres $G1, G2, G3, \dots$
- Por último, se ha detectado un error, que no hemos sido capaz de corregir, y puede dar lugar a fallos a la hora de interpretar la solución del problema. Si nos fijamos en la tabla resultante del problema anterior, nos queda:

Información	Datos	Resultados	Grafico			
El flujo total es 9.						
La solución es:						
	G1	G2	G3	G4	G5	G6
G1		6	3			9
G2			1	5		
G3					7	
G4			3			2
G5					7	
G6	9					

Ahí aparece una arista que no debe estar: $(G1, G6)$, por la cual se supone que pasan 6 unidades del flujo máximo. Este arco realmente no existe, entonces...¿Por qué aparece esta solución?

Muy bien, como hemos visto antes, en la formulación del problema, como ese arco no existe, deberíamos poner su capacidad en $u_{1n} = 0$, obligando a que $x_{in} = 0$.

No obstante, la segunda restricción que aparece en nuestro programa es $Bx \leq b$. Y para disminuir el número de variables de nuestro problema, solo metemos en B las restricciones donde $u_{ij} > 0$, es decir, la de los arcos que realmente aparecen en los datos. Luego B tendrá tan solo $m + 1$

renglones (o filas, o restricciones); siendo m el número de arcos reales que aparecen en los datos, más el ficticio. En lugar de esto, para hacer la formulación correcta, debería tener h^2 renglones (siendo h el número de nodos).

Sin embargo no es óptimo trabajar con tantas, y además es inútil meter restricciones sobre estas variables, ya que no influyen en la función objetivo. Por eso, no es de sorprender que en estos casos siempre aparezca este arco $(1, n)$ con un flujo distinto a cero.

No obstante, si miramos en la tabla los arcos que nos interesan (los que hemos metido en los datos, los que realmente existen), estos están correctos. Debemos por tanto ignorar ese dato de más. E igualmente, al imprimir el grafo resultante, puesto que es un nodo ficticio, que no existe, no nos lo dibuja, y nos da también el grafo resultante correcto.

3.4. Una curiosidad

Vamos a plantear una situación que se nos ocurrió a la hora de hacer el trabajo, y que hemos sabido resolver exitosamente.

Imaginemos que tenemos el Problema de Máximo Flujo usual, pero ahora queremos añadir una nueva restricción: en lugar de exigir la no negatividad de las variables, podemos exigir que estas sean mayor o igual que cierta cantidad. Esto es, estamos obligando a que por algunos arcos pase, forzosamente, cierta cantidad mínima. Luego la segunda restricción del problema, que la teníamos como:

$$0 \leq x_{ij} \leq u_{ij}$$

Se nos convertiría en

$$c_{ij} \leq x_{ij} \leq u_{ij}$$

Nos podríamos preguntar qué utilidad tiene este supuesto en el caso real. Pues bien, imaginemos que estamos maximizando el flujo de agua residual que pasa por las tuberías. Si no exigimos que por todas las tuberías siempre pase cierta cantidad de flujo, probablemente habrá algunas por la que no pase el flujo. Esto podría ocasionar dificultades: la oxidación de las tuberías, el anidamiento de ciertas alimañas (como las ratas) en estas, el estancamiento del agua en las mismas (favoreciendo el crecimiento de bacterias y otros patógenos), la obstrucción de esas tuberías...

Como estas, puede haber muchas otras situaciones en las que nos convenga exigir, por ejemplo, que por todos los arcos circule cierta cantidad de flujo.

Y ahora que hemos expuesto el problema, podemos preguntarnos por la resolución del mismo. Esta consiste en tomar una nueva variable $x'_{ij} = x_{ij} - c_{ij}$. Por tanto, obtenemos un nuevo problema con la segunda restricción de la forma

$$0 \leq x'_{ij} \leq u_{ij} - c_{ij}.$$

Una vez resulto el problema alternativo, deshacemos el cambio de variable, y obtenemos $x_{ij} = x'_{ij} + c_{ij}$, que cumple el requisito que buscábamos inicialmente.

4. Un algoritmo alternativo de resolución

Siempre podemos resolver el problema como hemos hecho hasta ahora, i.e. aplicando el método simplex usual. No obstante, para los Problemas de Máximo Flujo existe una simplificación del método simplex, que da lugar a un algoritmo alternativo de resolución.

Lo interesante de este algoritmo es que es muy gráfico, y permite resolver a mano problemas que, de tratar de hacerlos con las tablas del simplex, no serían resueltos en un tiempo razonable.

Vamos a explicar la resolución por el método de Cortes Mínimo de Ford Fulker-son.

Los pasos a seguir son los siguientes:

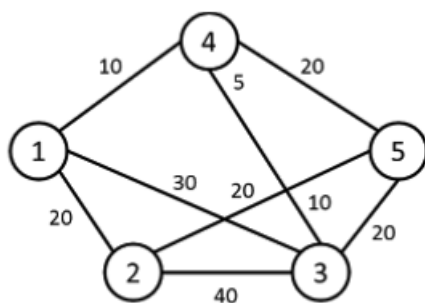
- Se identifica el nodo origen y destino.
- Se parte desde el nodo de origen y se escoge el arco que posea mayor flujo.
- Se identifican los nodos de transbordo.
- Repetir como si el nodo intermediario fuera el nodo origen.
- Se calcula "k" (como explicaremos más adelante) y las nuevas capacidades.
- Dado el resultado, se cambian las capacidades y se repite el mismo procedimiento desde el inicio.

Debemos tener en cuenta en lo que sigue el siguiente formulario:

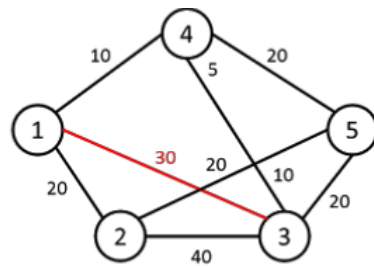
$$C_{i,j,ji} = (C_i - k, C_j + k)$$

Donde C es la capacidad y k el flujo mínimo que puede pasar por el nodo ese.

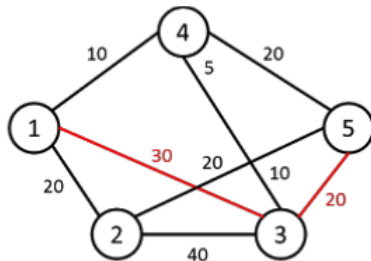
Vamos a ilustrar todos los pasos con un ejemplo. Hallemos el flujo máximo del siguiente grafo, siendo (1) el nodo origen y (5) el de destino:



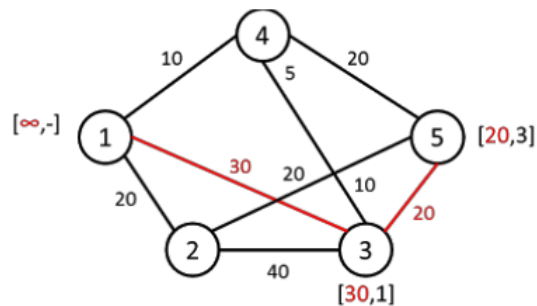
Se escoge desde el nodo de origen aquel flujo que sea el mayor, en este caso es 30, y va dirigido al nodo número 3.



Se identifica el nodo de transbordo como $[30,1]$, 30 es la capacidad, y 1 es el nodo del cual proviene la capacidad y luego repetimos todo el proceso, como si el nodo intermediario fuese el nodo de origen. Se tiene como flujo mayor 20 del nodo numero 3 al nodo número 5, con el nodo de transbordo como $[20,5]$.



Ahora que hemos llegado al nodo de destino, procedemos a calcular "k" las capacidades nuevas.



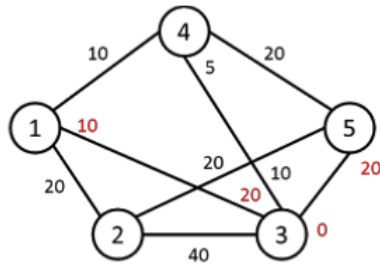
Las nuevas capacidades son:

$$k = \min(\text{inf}, 30, 20) = 20$$

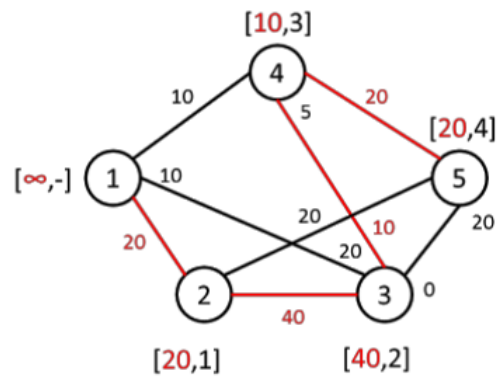
$$C_{13,31} = (30 - 20, 0 + 20) = (10, 20)$$

$$C_{35,53} = (20 - 20, 0 + 20) = (0, 20)$$

Luego de haber calculado las nuevas capacidades, es necesario reemplazarlas:



Se realiza el proceso otra vez, haciendo la ruta con los mayores flujos:



Los cálculos son:

$$k = \min(\text{inf}, 20, 40, 10, 20) = 10$$

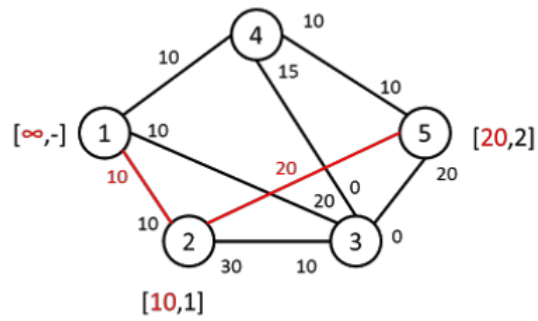
$$C_{12,21} = (10, 10)$$

$$C_{23,32} = (30, 10)$$

$$C_{34,43} = (0, 15)$$

$$C_{45,54} = (10, 10)$$

Volvemos a hacer el proceso y escogemos el camino 1,2. Como se puede observar si se tomara rumbo del nodo 2 al nodo 3 terminaría trancado, obligándose a volver al nodo origen, por lo que se toma el camino 2,5.



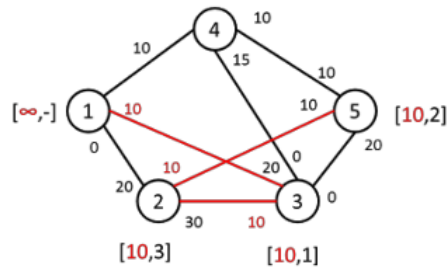
Los nuevos cálculos son

$$k = \min(\text{inf}, 10, 20) = 10$$

$$C_{12,21} = (0, 20)$$

$$C_{25,52} = (10, 10)$$

Se actualizan las capacidades y procedemos a resolver de nuevo. Esta vez agarraremos el camino de 1,3.



Las nuevas capacidades son:

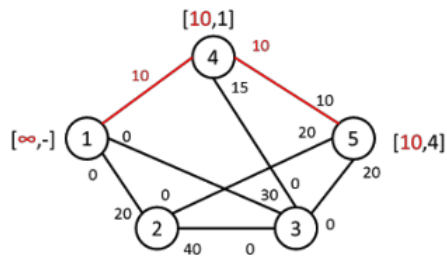
$$k = \min(\text{inf}, 10, 10, 10) = 10$$

$$C_{13,31} = (0, 30)$$

$$C_{23,32} = (40, 40)$$

$$C_{25,52} = (0, 20)$$

Y por último el camino 1,4:



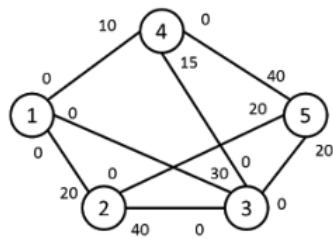
Las nuevas capacidades son:

$$k = \min(\text{inf}, 10, 10) = 10$$

$$C_{14,41} = (0, 10)$$

$$C_{45,54} = (0, 40)$$

Reemplazando las nuevas capacidades, nos queda de la siguiente forma, las capacidades del nodo de origen quedan como 0, por lo cual seguimos a sumar a todas las k y ahí conseguimos el flujo máximo.



El flujo máximo es la sumatoria de todas las k , luego será $F = 20 + 10 + 10 + 10 + 10 = 60$.