

Práctica 8

Flujo de valor máximo

En esta cuarta práctica vamos a ver cómo resolver el problema de calcular un flujo de valor máximo. Para ello, tendremos un grafo G y un vector de pesos p que a cada arco e le asignará un peso p_e . En este tema el peso de cada arco representará la capacidad de dicho arco. Tenemos por tanto que encontrar la manera óptima de asignar el flujo, de manera que éste valor sea máximo y que en cada vértice intermedio entre y salga la misma cantidad de flujo.

8.1. Resolución con MatLab

8.1.1. El comando `maxflow`

MatLab dispone del comando `maxflow` que nos permite resolver fácilmente el problema del flujo máximo. Su uso es muy sencillo y solamente hemos de indicarle los vértices inicial y final (los vértices *fuentes* y *salida*, utilizando la misma notación que en las clases de teoría). Si s y t denotan, respectivamente, la fuente y la salida de la red, la expresión a utilizar será:

```
>> maxflow(G,s,t)
```

De esta manera obtendremos el valor del flujo máximo permitido por esa red. Sin embargo, lo más interesante de dicho comando no es que calcule el valor del flujo máximo, sino que nos proporciona el grafo asociado al flujo máximo, esto es, el grafo en el que a cada arco le asigna como peso el valor del flujo, siempre que el flujo sea no nulo. Para ello, utilizamos la siguiente intrucción:

```
>> [flujo,Grafo]=maxflow(G,s,t)
```

Bajo el nombre de `flujo` almacenará el valor del flujo máximo, y valor el valor de `Grafo` almacena el grafo asociado a este flujo. Con este nuevo grafo podemos utilizar todos los comandos vistos en la primera práctica, como por ejemplo los relativos a la representación gráfica.

Además de obtener el flujo máximo y su grafo asociado, también es posible obtener por pantalla el corte de valor mínimo. Para ello debemos añadir dos argumentos de salida, donde se almacenarán los vértices de la parte inicial y final del corte, respectivamente:

```
>> [flujo,Grafo,corteInicial,corteFinal]=maxflow(G,s,t)
```

Ejemplo 8.1. Consideremos el grafo representado en la Figura 8.1. El primer paso será definir este

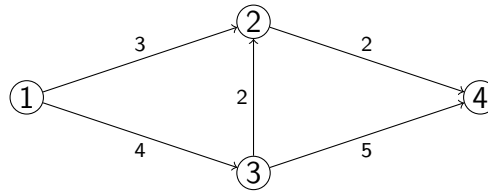


Figura 8.1: Grafo del Ejemplo 8.1.

grafo en MatLab:

```
>> s=[1 1 2 3 3];
>> t=[2 3 4 2 4];
>> cap=[3 4 2 2 5];
>> G=digraph(s,t,cap)
```

G =

digraph with properties:

```
Edges: [5x2 table]
Nodes: [4x0 table]
```

A continuación, para calcular el flujo máximo entre los vértices 1 y 4, así como su grafo y corte mínimos asociados, utilizamos la siguiente instrucción:

```
>> [flujo,Grafo,corteInicial,corteFinal]=maxflow(G,1,4)
```

flujo =

6

Grafo =

digraph with properties:

```
Edges: [4x2 table]
Nodes: [4x0 table]
```

corteInicial =

1

2

corteFinal =

3

4

Vemos por tanto que el valor del flujo máximo es de 6 unidades, y ese flujo máximo se obtiene con el corte formado por los vértices 1 y 2 y los vértices 3 y 4. Vamos a pedir por pantalla los arcos del grafo asociado al flujo máximo para ver de qué manera se reparte el flujo:

```
>> Grafo.Edges
```

ans =

EndNodes	Weight
-----	-----
1 2	2
1 3	4
2 4	2
3 4	4

Eso significa que para obtener el flujo máximo hemos de asignar el valor 2 a los arcos $1 \rightarrow 2$ y $2 \rightarrow 4$ y el valor 4 a los arcos $1 \rightarrow 3$ and $3 \rightarrow 4$. En la Figura 8.2 hemos representado en rojo los valores del flujo máximo y hemos puesto en línea discontinua el corte de valor mínimo:

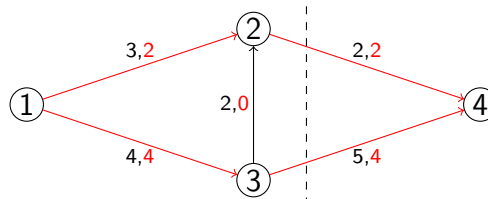


Figura 8.2: Grafo del Ejemplo 8.1 con su flujo máximo (en rojo) y corte mínimo (en línea discontinua).

8.1.2. Opciones del comando `maxflow`

En el apartado anterior ya hemos visto las opciones relativas a los argumentos de salida del comando `maxflow`. Además de estas opciones, también nos permite indicar el tipo de algoritmo que deseamos utilizar:

- El algoritmo utilizado por defecto es el de Boykov-Kolmogorov. Para su uso habría que indicar la opción `searchtrees`.

- b) Una alternativa, solamente válida para grafo orientados, es el uso del algoritmo de Ford-Fulkerson. Para ello deberíamos de indicar la opción `augmentpath`.
- c) Para utilizar una segunda alternativa, también válida únicamente para grafos orientados, deberíamos utilizar la opción `pushrelabel`.

8.1.3. Representación gráfica del flujo de valor máximo

Al igual que hemos visto en las prácticas anteriores, una vez resuelto el problema del flujo máximo podemos utilizar las opciones de las representaciones gráficas vistas en la primera práctica para indicar cuál es el flujo máximo y el corte mínimo. Comenzamos haciendo la representación gráfica del grafo original:

```
>> h=plot(G)
```

A continuación, remarcamos en color (rojo, en este caso) los arcos con un flujo no nulo:

```
>> highlight(h,Grafo,'EdgeColor','red')
```

Además, podemos añadir a esos arcos remarcados el valor de su flujo asociado:

```
>> cap=Grafo.Edges.EndNodes;
>> labeledge(h,cap(:,1),cap(:,2),Grafo.Edges.Weight);
```

Con la primera instrucción almacenamos los vértices inicial y final de cada arco, y con la segunda instrucción asignaremos a cada arco el valor de su flujo. Por último, podemos también distinguir el corte de valor mínimo:

```
>> highlight(h,corteInicial,'NodeColor','blue')
>> highlight(h,corteFinal,'NodeColor','green')
```

Con la primera instrucción dibujaríamos en azul los vértices de la parte inicial del corte de valor mínimo, y con la segunda instrucción dibujamos en verde los vértices de la parte final del corte.

Ejemplo 8.2. *Continuemos con el Ejemplo 8.1. En la Figura 8.2 habíamos representado el flujo máximo y el corte mínimo. Utilizando las instrucciones citadas anteriormente, MatLab relizaría la representación gráfica de la Figura 8.3.*

8.2. Ejemplo ilustrativo

Consideremos un ejemplo del libro *Introduction to Mathematical Programming, Applications and Algorithms*, W.K.Winston, Duxbury Press: la compañía petrolífera Sunco Oil quiere enviar la mayor cantidad posible de petróleo por hora a través de oleoductos desde el vértice llamado F hasta el vértice llamado S (F=fuente, S=salida, utilizando la notación vista en las clases de teoría). A lo

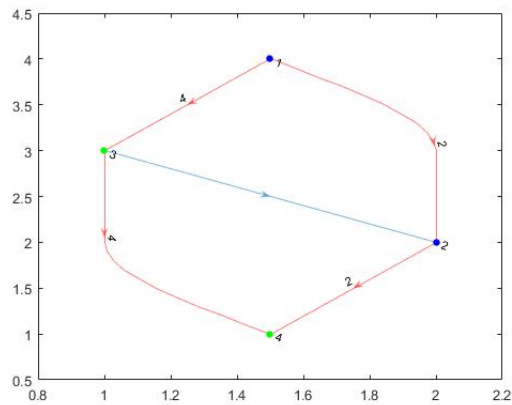


Figura 8.3: Representación gráfica de MatLab del flujo máximo del grafo del Ejemplo 8.1.

largo de este trayecto, el petróleo podrá pasar las estaciones 1, 2 y/o 3. En la Figura 8.4 hemos representado gráficamente la estructura de estas estaciones, así como de los vértices inicial F y final S. Cada uno de los arcos tiene asociado un número que indica la capacidad de petróleo (en millones de barriles por hora) que se pueden bombear a lo largo del arco. El objetivo es por tanto maximizar la cantidad de petróleo enviado desde F hasta S e indicar cómo deberá hacerse dicho envío.

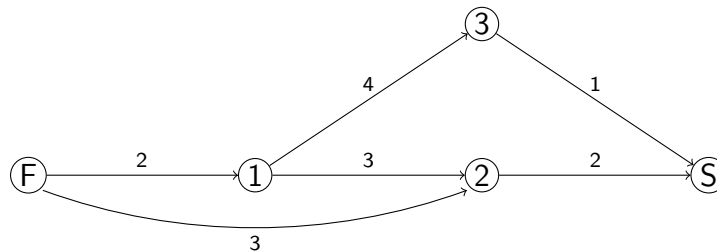


Figura 8.4: Distribución de las estaciones petrolíferas de la compañía Sunco Oil con la capacidad de cada oleoducto.

Como hemos hecho en otras ocasiones, el primera paso será definir el grafo asociado a las estaciones petrolíferas en MatLab:

```
>> s=[1 1 2 2 3 4];
>> t=[2 3 3 4 5 5];
>> capacidad=[2 3 3 4 2 1];
>> G=digraph(s,t,capacidad,{'F','Estación 1','Estación 2','Estación 3','S'})
```

G =

digraph with properties:

Edges: [6x2 table]

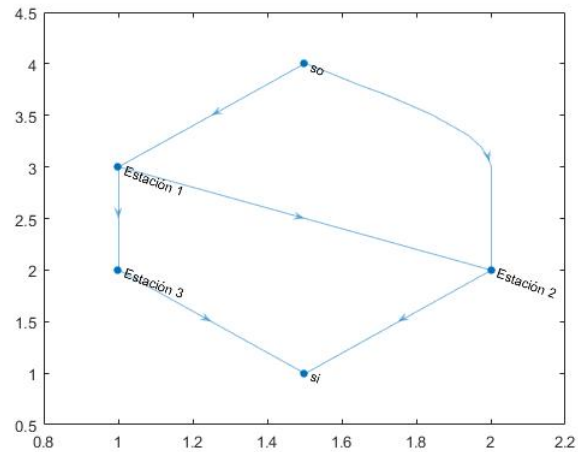


Figura 8.5: Representación gráfica de MatLab del sistema de oleoductos de la compañía Sunco Oil.

```
Nodes: [5x0 table]
h=plot(G)
```

Con esta última instrucción hemos solicitado la representación gráfica realizada por MatLab, que la podemos ver en la Figura 8.5.

En términos de grafos y redes, el problema se traduce en encontrar el flujo máximo a esta red:

```
>> flujo=maxflow(G,'F','S')
```

```
ans =
```

```
3
```

Con esta orden obtenemos el valor del flujo máximo, que es de 3 unidades, es decir, de 3 millones de barriles a la hora. Sin embargo, esta información es pobre, puesto que no nos dice cómo hemos de repartir el flujo. A continuación vamos a utilizar la misma instrucción `maxflow` pero solicitando tanto el grafo asociado al flujo máximo como el corte mínimo:

```
>> [flujo,Grafo,corteInicio,corteFinal]=maxflow(G,'F','S')
```

```
flujo =
```

```
3
```

```
Grafo =
```

```
digraph with properties:
```

```

Edges: [5x2 table]
Nodes: [5x0 table]

corteInicial =

    'F'
    'Estación 1'
    'Estación 2'
    'Estación 3'

corteFinal =
    'S'
>> Grafo.Edges

ans =

```

EndNodes		Weight
-----		-----
'F'	'Estación 1'	1
'F'	'Estación 2'	2
'Estación 1'	'Estación 3'	1
'Estación 2'	'S'	2
'Estación 3'	'S'	1

Con la última instrucción utilizada, sabemos cómo hemos de distribuir el flujo para obtener el valor del flujo máximo, que ya sabemos que es de 3 millones de barriles a la hora:

- Desde la estación de origen se enviará un millón de barriles a la hora a la estación 1 y dos millones a la estación 2.
- Desde la estación 1 se enviará un millón de barriles a la hora a la estación 3.
- Desde la estación 2 se enviarán dos millones de barriles a la hora a la estación final.
- Desde la estación 3 se enviará un millón de barriles a la hora a la estación final.

Si queremos la representación gráfica de MatLab del grafo asociado al flujo máximo, podemos utilizar las siguientes instrucciones:

```

>> h.EdgeLabel={};          # Con esta instrucción eliminamos los valores de la
                                # capacidad de cada arco.
>> highlight(h,Grafo,'EdgeColor','r') # Con esta instrucción remarcamos los
                                # arcos con valores de flujo no nulo.
>> cap=Grafo.Edges.EndNodes;        # Con esta instrucción guardamos en el
                                # vector cap los vértices inicial y

```

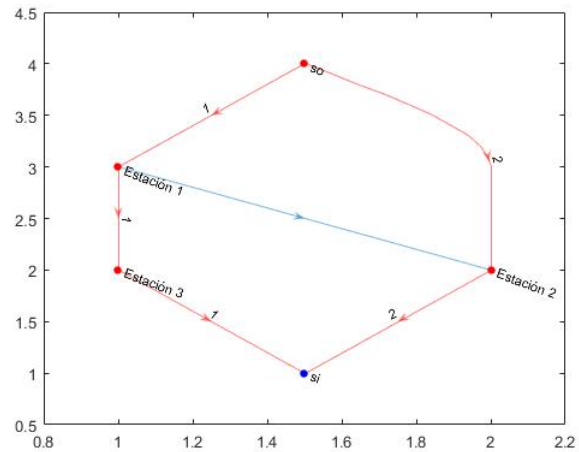


Figura 8.6: Representación gráfica de MatLab de grafo asociado al flujo máximo en el sistema de oleoductos de la compañía Sunco Oil.

```
# final de cada arco.
>> labeledge(h,cap(:,1),cap(:,2),Grafo.Edges.Weight)
>> highlight(h,corteInicial,'NodeColor','red')
>> highlight(h,corteFinal,'NodeColor','blue')
```

Las dos últimas instrucciones dibujarán los vértices de la parte inicial de corte mínimo en rojo y los vértice de la parte final del corte en azul. La representación final de MatLab la podemos ver en la Figura 8.6. Asimismo, en la Figura 8.7 hemos representado el flujo máximo de acuerdo a la representación original, dibujando en rojo los arcos utilizados y añadiendo el rojo el valor del fujo. Además, la línea discontinua representa el corte de valor mínimo.

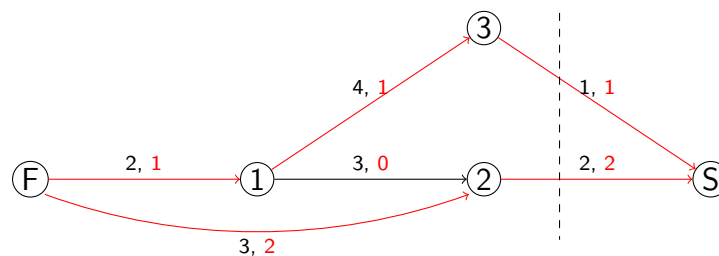


Figura 8.7: Distribución del flujo máximo en las estaciones petrolíferas de la compañía Sunco Oil con la capacidad de cada oleoducto.