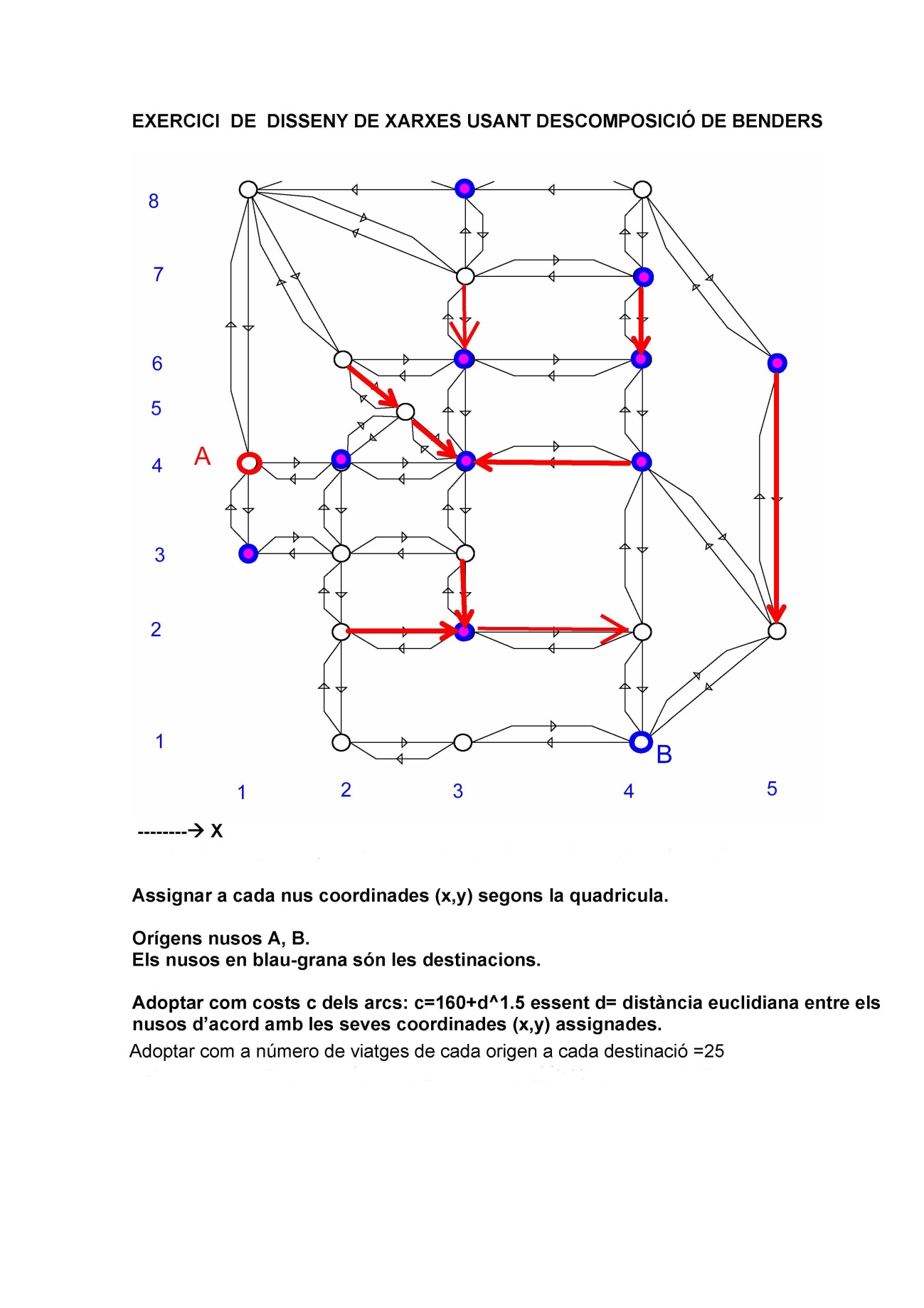
**ASSIGNMENT 1. CUTTING PLANE ALGORITHM (Dantzig)**

The following multicommodity network flow problem must be solved.

using the cutting plane decomposition algorithm by dualizing the joint capacity constraint.

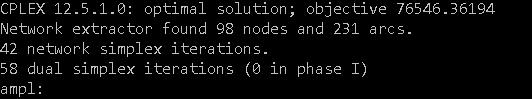


The values used in this assignment will be those of the Benders decomposition exercise. This exercise can be done in groups of two students.

To carry out the assignment follow the next steps:

1. Solve initially the problem without capacities on links, i.e.:

1. Report the solution and optimal function value. Then select four links and set on them a global capacity bound violated by the reported solution of the uncapacitated problem. Edit the file caps.dat with the selected values.



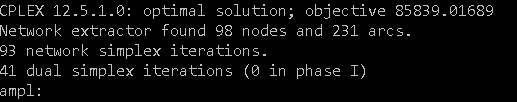
El valor obtenido para el problema sin capacidad tiene asociado un valor de la función objetivo de 76546.4 unidades. Ahora debemos añadir 4 arcos al problema para que este sea un problema con restricción de capacidad. Añadimos los siguientes valores (disponibles en el archivo caps.dat).

Las restricciones violadas se extraen del fichero flows del problema resuelto en el problema sin capacidades.

Se seleccionan los siguientes arcos restringiendo el flujo a 12 en todos ellos. También se presenta el valor obtenido en el problema sin capacidad.

1. {1,4} : 12.00 (antes 16.00)
2. {7,19} : 12.00 (antes 32.00)
3. {11,8} : 12.00 (antes 16.00)
4. {11,22} : 12.00 (antes 64.00)
5. With the joint capacities on the selected links solve the capacitated problem using the AMPL files. (cuttingOGE.mod, .dat, .run, caps.dat). If the capacitated problem results unfeasible, then try with new capacity values.

Lo primero que debemos hacer es resolver el problema para ver que este es factible (en caso de no serlo deberiamos cambiar las restricciones violadas por el problema). El valor obtenido ahora para la función objetivo pasa a ser de 85839.1 unidades.



Al restringir las capacidades se obtiene un valor más elevado para la función objetivo.

1. Implement the cutting plane algorithm. Note that the algorithm requires an initial feasible solution of the capacitated problem. In order to determine one, either a) use the network’s transformation that uses an artificial node and artificial links, being the costs of associated to artificial links the constant “big-M”, or b) use AMPL to find a feasible solution by solving the capacitated problem with some other function (1’s as cost in any link…)

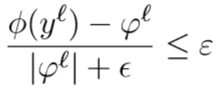
En la primera iteración del master (MP) se inyecta a los nodos origen (3;11) el flujo desde el nodo artificial 1 (llamado art1 en AMPL). Una vez inyectado este flujo, se asignan las extracciones a todos los otros arcos artificiales mediante t[i,l]. Para calcular el problema w incial se asigna al multiplicador mu[2,1], un valor 15 (arbitrario pequeño) ya que esta restricción no esta sujeta a demanda restringida (tiene asociado un flujo elevado ‘big-M’). De esta manera obtenemos una solución inicial.

Tras esta iteración, iniciaremos el algoritmo de planos de corte de Danzig e iremos añadiendo nuevos cortes (variables mu0) hasta hallar un problema en que ningún coste para cualquier x que pudiera entrar al sistema fuera menor a zero. Al hallar este valor aseguramos que nuestra solución no podria mejorar añadiendo ninguno de los otros nodos o arcos de flujo al sistema. Iterativamente este cálculo se realiza evaluando el GAP.

En las siguientes iteraciones se optimizan los multiplicadores mu, entonces estos valores se prueban en w asociandose a unas x. Estos valores se devuelven entonces al problema Master (MP). De vuelta al MP se pruevan estas x y se vuelven a asignar respectivos multiplicadores, ya que, estas variables (mu) son las variables de decisión (o a optimizar) del problema MP. Estas variables permiten resolver el sistema de flujo con la restricción de capacidad.

En resumen, el proceso genera cortes (problema MP) para encontrar multiplicadores (mu) que se van evaluando en el problema “original” (Zk). De esta manera se obtiene una solución factible que se llevará al (MP) para que este obtenga un nuevo multiplicador mu para llevar otra vez al problema “original”.

La convergencia del algorítmo se avalua mediante el siguiente GAP:



Donde hace referencia al problema “original” y el otro valor al que se encarga de ir generando multiplicadores. En este algorítmo el GAP obtenido en cada iteración será mayor o igual a 0, ya que, el problema “original” es siempre mayor o igual que los cortes que proporciona MP.

Donde lo que hacemos es la diferencia entre el problema máster y el sub-problema dividido entre el valor absoluto del sub-problema más una constante arbitraria pequeña. Cuando este valor sea inferior o igual a una constante que tiende a 0, podemos decir que ambos algoritmos han convergido. En esta instancia, al ser pequeña, el GAP se iguala a 0 en 5 iteraciones.

Este proceso permite obtener soluciones a problemas formados por numerosas constricciones y variables.

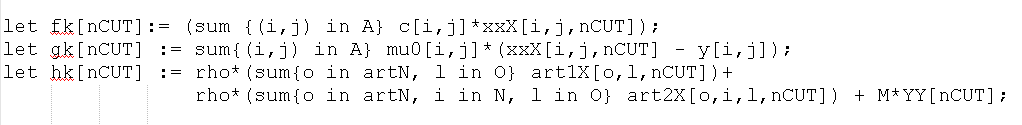
A continuación, se muestra para cada iteración del algoritmo los siguientes indicadores o valores:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Iter** | **Zk** | **Wk** | **CTXk+1** | **Xk+1 <= d** | **GAP** |
| 1 | 240000 | -64161 | 85839 | xx[1,4] - y[1,4] = -12  xx[7,19] -y[7,19] = 0  xx[11,8] - y[11,8] = 0  xx[11,22] - y[11,22] = 0 | 1.27 |
| 2 | 85839 | -63292.7 | 90868.3 | xx[1,4] - y[1,4] = 0  xx[7,19] - y[7,19] = -12  xx[11,8] - y[11,8] = 0  xx[11,22] - y[11,22] = 0 | 1.73 |
| 3 | 85839 | -60772.7 | 93388.3 | xx[1,4] - y[1,4] = 0  xx[7,19] - y[7,19] = -12  xx[11,8] - y[11,8] = -12  xx[11,22] - y[11,22] = 0 | 1.71 |
| 4 | 85839 | -57172.7 | 96988.3 | xx[1,4] - y[1,4] = 0  xx[7,19] - y[7,19] = -12  xx[11,8] - y[11,8] = -12  xx[11,22] - y[11,22] = -12 | 1.67 |
| 5 | 85839 | 85839 | 93388.3 | xx[1,4] - y[1,4] = 0  xx[7,19] - y[7,19] = -12  xx[11,8] - y[11,8] = -12  xx[11,22] - y[11,22] = 0 | 0 |

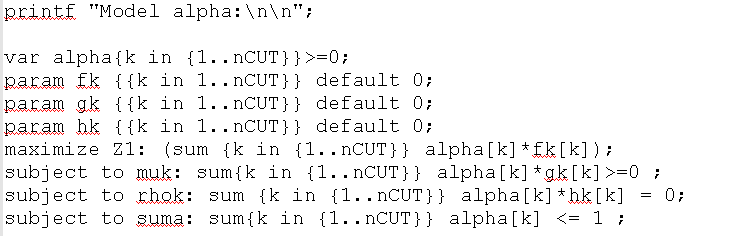
La información presentada en la 5ª columna indica si la restricción de capacidad fijada es factible/infactible. Será una solución infactible para ese arco sí el valor > 0.

El valor del Zk rápidamente se optimiza (hallándose en la segunda iteración el valor óptimo). El valor w (wk) toma valores negativos hasta que en la 5ª iteración acaba coincidiendo con el óptimo, siendo el valor del GAP = 0.

En cada iteración se guardan como parámetros las soluciones asociadas a f(k), h(k) y g(k). Donde cada parámetro contendrá:



Una vez obtenido el GAP = 0, se formula el siguiente problema para obtener una combinación lineal de α candidatas (1 para cada iteración) a la solución del problema Zk. A continuación se muestra el código AMPL usado para este modelo:



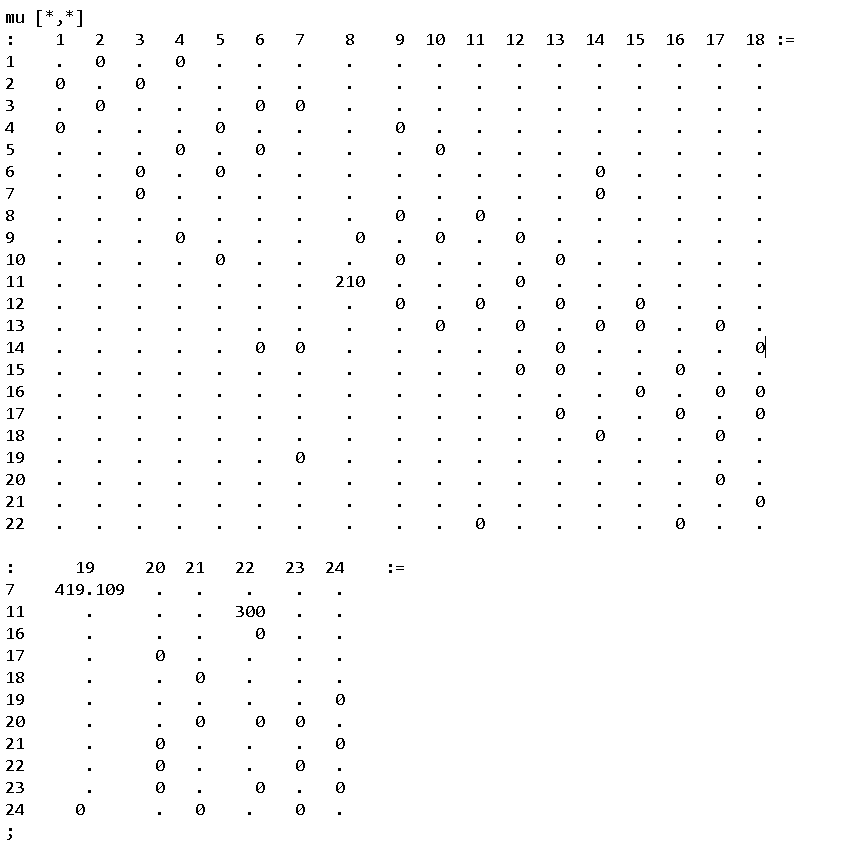
El mejor valor se obtiene en el segundo corte, asignando valor 1. Por lo que el valor de Z1 es equivalente a la solución obtenida anteriormente mediante el algoritmo de planos. También comentar que en el algoritmo la solución de Zk es óptima también a partir de la segunda iteración.

***Nota: en el siguiente apartado se pueden ver la solución de el flujo para cada origen, el flujo total y los multiplicadores mu obtenidos en la solución óptima del algoritmo de Dantzig.***

ANNEXO

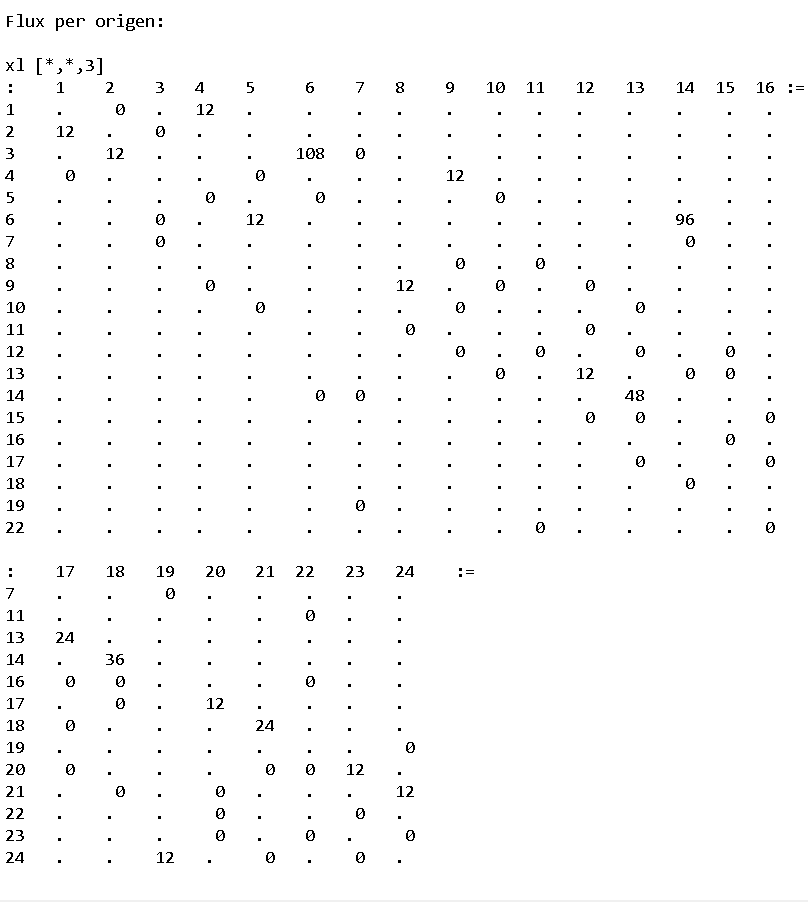
A continuación, se añaden los valores obtenidos para la variable mu y la variable xl. Se presentan los outputs obtenidos en la útlima iteración debido a la extensión del output.

**Multiplicadores mu**

****

**Variables respuesta xl (marcan la capacidad final que pasa por el flujo por cada orígen)**

Orígen 3:

****

Orígen 11:

