TEORÍA DE REDES DE TELECOMUNICACIONES

GRADO EN INGENIERÍA TELEMÁTICA
GRADO EN INGENIERÍA EN SISTEMAS DE TELECOMUNICACIÓN

CURSO 2018-2019

Práctica #6. Encaminamiento del tráfico. Formulaciones flujo-enlace

(1 sesión)

Autor:

Pablo Pavón Mariño
Juan Pedro Muñoz Gea
María Victoria Bueno Delgado
Ángel Antonio Pintado Sedano
Juan Carlos Sánchez Aarnoutse

1 Objetivos

Los objetivos de esta práctica son:

1. Crear algoritmos de Net2Plan que resuelvan algunas variantes de problemas de encaminamiento del tráfico, resolviendo formulaciones flujo-camino utilizando la librería *Java Optimization Modeler* (JOM).
2. Conseguir práctica en las diferentes formas de escribir problemas de optimización en JOM, aprovechando sus capacidades de representación vectorial.
3. Explorar el potencial de la representación matricial de las restricciones de encaminamiento en JOM y Net2Plan.

2 Duración

Esta práctica está diseñada para una sesión de dos horas.

3 Evaluación

Esta práctica ha sido diseñada para guiar al estudiante en su aprendizaje en Net2Plan. Las anotaciones que los estudiantes hagan en este documento son para su uso cuando repasen la asignatura y no es necesario que se entregue al profesor para su evaluación.

4 Documentación

Los recursos necesarios para este trabajo de laboratorio son:

- Documentación de la librería JOM (ver `http://www.net2plan.com/jom`).
- La herramienta Net2Plan y su documentación (ver `http://www.net2plan.com/`).
- Instrucciones contenidas en este guión de la práctica.

5 Trabajo previo antes de venir al laboratorio

- Leer las secciones 4.3, 4.6.3 y 4.6.4 de la referencia [1], y los apuntes relacionados con la formulación flujo-enlace.
- Refrescar las lecturas de la documentación de JOM en `http://www.net2plan.com/jom`, y en particular, cómo se manejan los vectores de variables y restricciones.

6 Encaminamiento de Ancho de Banda Mínimo

Estamos interesados en crear algoritmos de Net2Plan que resuelvan el problema de encaminamiento de ancho de banda mínimo utilizando JOM. El algoritmo se define de la siguiente manera:

- Parámetros de entrada (constantes conocidas):
  - $\mathcal{N}$: Conjunto de nodos.
  - $\mathcal{E}$: Conjunto de enlaces. De esta información, $\delta^+(n)$ denota el cunjunto de enlaces salientes del nodo n, y $\delta^-(n)$ el conjunto de enlaces entrantes al nodo n.
  - $u_e, e \in \mathcal{E}$: Capacidad del enlace e.
  - $\mathcal{D}$: Conjunto de demandas unicast.
  - $h_d, d \in \mathcal{D}$: Tráfico ofrecido por la demanda d.
- Variables de decisión:
  - $x_{de}, d \in \mathcal{D}, e \in \mathcal{E}$: Tráfico de la demanda d que atraviesa el enlace e.
- Formulación:

$$\min \quad \sum_{de} x_{de}, \quad \text{sujeto a:} \tag{1a}$$

$$\sum_{e\in\delta^+(n)} x_{de} - \sum_{e\in\delta^-(n)} x_{de} = \begin{cases} h_d, \text{ if } n = a(d) \\ -h_d, \text{ if } n = b(d) \\ 0, \text{ en otro caso} \end{cases} \quad , \quad \forall d \in \mathcal{D}, \forall n \in \mathcal{N} \tag{1b}$$

$$\sum_{d} x_{de} \leq u_e, \quad \forall e \in \mathcal{E} \tag{1c}$$

$$x_{de} \geq 0, \quad \forall d \in \mathcal{D}, e \in \mathcal{E} \tag{1d}$$

La función objetivo (1a) minimiza la cantidad total de tráfico en los enlaces. Las restricciones (1b) son las restricciones de conservación de flujo. Las restricciones (1c) representan que por cada enlace, el tráfico cursado en el enlace es menor o igual que su capacidad (y por lo tanto, ningún enlace está saturado). Finalmente, (1d) evita que un enlace curse una cantidad negativa de tráfico de una demanda, ya que esto no tiene sentido físico.

7 Algoritmo Net2Plan

El alumno debe desarrollar un algoritmo Net2Plan para resolver el problema (1) siguiendo los siguientes pasos:

1. Copiar el fichero `AlgorithmTemplate.java` que se encuentra en el Aula Virtual y renombrarlo como `FlowLinkUnicast.java`.
2. El algoritmo no tiene parámetros de entrada.
3. Este algoritmo funcionará para cualquier tipo de encaminamiento del diseño de entrada. Se tiene que eliminar cualquier encaminamiento cursado de la red (rutas en *source routing*, o reglas de encaminamiento en el encaminamiento salto a salto). Se puede hacer llamando al método `removeAllUnicastRoutingInformation` del objeto de entrada `NetPlan`.

4. Crear un objeto del tipo `OptimizationProblem` (de nombre p. e. `op`).
5. Añadir las variables de decisión del problema, con nombre `x_de`: una variable por demanda y enlace. El valor mínimo de las variables se establece a cero, el máximo a *Double.MAX_ VALUE*.
6. Establecer la función objetivo del problema. Utilizar la función `sum` de JOM.
7. Utilizar un bucle for doble con una iteración por nodO de la red, y una iteración interna por demanda, para añadir las restricciones (1b). Para añadir las restricciones de conservación de una demanda `d` y nodo `n`:
   - Establecer los parámetros de entrada JOM:
     - `deltaPlus` con los índices de los enlaces de salida del nodo actual. Para ello, utilizar el método *getOutgoingLinks* del objeto nodo para obtener los enlaces de salida, y el método *NetPlan.getIndexes* para convertir la colección de enlaces a sus índices.
     - `deltaMinus` con los índices de los enlaces entrantes del nodo actual. Para ello, utilizar el método *getIncomingLinks* del objeto nodo para obtener los enlaces entrantes, y el método *NetPlan.getIndexes* para convertir la colección de enlaces a sus índices.
     - `h_d` con el tráfico ofrecido de la demanda actual.
     - `d` con el índice de la demanda actual.
   - Establecer las restricciones utilizando la función `sum`, sobre `x_de`, pero restringiendo la suma a los elementos en la fila `d` y las columnas en `deltaPlus` o `deltaMinus`.
8. Utilizar un bucle for con una iteración por enlace para añadir las restricciones (1c). Dentro de cada iteración del bucle se añade una restricción. Para añadir la restricción de un enlace `e`:
   - Establecer un parámetro de entrada JOM de nombre `u_e`, con un valor igual a la capacidad actual del enlace, y el parámetro de entrada `e` con el índice actual del enlace.
   - Establecer la restricción utilizando la función `sum`, sobre todas las demandas (utilizando la palabra clave JOM `all` en la coordenada de la demanda), y el enlace de índice `e`.
9. Llamar al solver para obtener una solución numérica.
10. Recuperar la solución primal obtenida, y convertirla a un objeto *DoubleMatrix2D* utilizando el método *view2D*. Un objeto de la clase *DoubleMatrix2D* cotiene una matriz de dos dimensiones y permite hacer operaciones con ella de forma eficiente.
11. Utilizar el método de *NetPlan* llamado *setRoutingFromDemandLinkCarriedTraffic*. Este método crea automáticamente las rutas en la red que son consistentes con lo que aparece en la matriz de dos dimensiones x_{de} calculada. Notar que en los valores x_{de} obtenidos, cada coordenada (d, e) contiene la cantidad de tráfico de d en el enlace e, y *no* la fracción respecto a h_d. Esto es importante cuando se llama al método *setRoutingFromDemandLinkCarriedTraffic*. Ya que esta formulación no puede crear bucles (las soluciones con bucles nunca son óptimas, ya que son estrictamente peores que las mismas rutas sin bucles), establecer la opción que las elimina automáticamente a *false*.

7.1 Comprobar el algoritmo

Cargar la red `example7nodesWithTraffic.n2p`. El algoritmo debe generar una solución con una cantidad de ancho de banda consumido en los enlaces de 155.5.

Si reducimos la capacidad de los enlaces a 15 unidades (en lugar de 50), la solución óptima consumirá 174.1 unidades.

Si reducimos la capacidad de los enlaces a 10 unidades, el problema será no factible (no hay suficiente capacidad en los enlaces para cursar el tráfico de ninguna forma).

Quiz 1. ¿El tráfico circula por los caminos más cortos para todas las demandas en el primer caso ($u_e = 50$)? ¿Se mantiene para $u_e = 15$?

8 Variaciones del problema

Quiz 2. Modificar la formulación (1) de tal forma que ahora las variables x_{de} son renombradas como $\hat{x}_{de}$, y miden la **fracción** ($\in [0;1]$) del tráfico de la demanda d, que es cursado a través del enlace e. Es decir, $x_{de} = h_d \hat{x}_{de}$.

- Variables de decisión:
  - $\hat{x}_{de}, d \in \mathcal{D}, e \in \mathcal{E}$: Fracción $\in [0, 1]$ del tráfico ofrecido de la demanda d que circula a través del enlace e
- Formulación:

$$\min \quad \sum_{de} h_d \hat{x}_{de}, \quad \text{sujeto a:} \tag{2a}$$

$$\sum_{e \in \delta^+(n)} \hat{x}_{de} - \sum_{e \in \delta^-(n)} \hat{x}_{de} = \begin{cases} 1, \text{ if } n = a(d) \\ -1, \text{ if } n = b(d) \\ 0, \text{ en otro caso} \end{cases} \quad , \quad \forall d \in \mathcal{D}, \forall n \in \mathcal{N} \tag{2b}$$

$$\sum_{d} h_d \hat{x}_{de} \le u_e, \quad \forall e \in \mathcal{E} \tag{2c}$$

$$\hat{x}_{de} \ge 0, \quad \forall d \in \mathcal{D}, e \in \mathcal{E} \tag{2d}$$

Rehacer el algoritmo con las nuevas variables de decisión (cambiando su nombre en JOM a `xx_de`). Para la multiplicación en la función objetivo utilizar la expresión JOM:

```
sum (h * xx_de)
```

donde:

- `h` es un vector fila con tantas coordenadas como demandas, conteniendo el tráfico ofrecido de la demanda. El vector del tráfico ofrecido se puede obtener en Net2Plan utilizando el método de la clase `NetPlan`:

  ```
  getVectorDemandOfferedTraffic
  ```

- Entonces, `h * xx_de` es un vector por multiplicación de matrices, que produce un vector con una coordenada por enlace, conteniendo el tráfico cursado del enlace.
- `sum (h * xx_de)` es entonces la suma del tráfico cursado en todos los enlaces

En su turno, para las restricciones de capacidad de los enlaces, utilizar la expresión `h * xx_de (all,e)` para obtener el tráfico cursado en el enlace `e`. Notar que es la multiplicación del vector `h` con una coordenada por demanda, con el vector `xx_de(all,e)`, con una coordenada por demanda también, conteniendo los valores $\hat{x}_{de}$ en el enlace `e`.

Quiz 3. En la formulación (2), notar que si las variables de decisión `xx_de` se restrigen a ser enteros (es decir, o 0 o 1), entonces el encaminamiento se restringe a ser no bifurcado. Añadir un parámetro de entrada al problema llamado `isNonBifurcated`, con valor por defecto `false`. Utilizar este parámetro para permitir al usuario elegir si el encaminamiento está restringido o no a ser bifurcado. Comprobar que la topología de ejemplo elegida en los tests no tiene solución cuando las capacidades de los enlaces se establece a 15 unidades, cuando el encaminamiento se restringe a ser no bifurcado (mientras que hay una solución cuando el encaminamiento es bifurcado).

Quiz 4. Implementar un algoritmo Net2Plan que resuelva una formulación flujo-camino para encontrar el encaminamiento que maximice el ancho de banda disponible en el cuello de botella. Tomar la formulación de los apuntes de clase, o de la sección 4.3 de [1]:

$$\max \quad u, \quad \text{sujeto a:} \tag{3a}$$

$$\sum_{e \in \delta^+(n)} x_{de} - \sum_{e \in \delta^-(n)} x_{de} = \begin{cases} h_d, \text{ si } n = a(d) \\ -h_d, \text{ if } n = b(d) \\ 0, \text{ en otro caso} \end{cases} \quad , \quad \forall d \in \mathcal{D}, \forall n \in \mathcal{N} \tag{3b}$$

$$\sum_d x_{de} \leq u_e - u, \quad \forall e \in \mathcal{E} \tag{3c}$$

$$x_{de} \geq 0, \quad \forall d \in \mathcal{D}, e \in \mathcal{E} \tag{3d}$$

Comprobar que la solución que maximiza el ancho de banda disponible en el cuello de botella puede ser diferente de la que minimiza el ancho de banda consumido medio.

9 Forma matricial de las restricciones del problema (opcional)

9.1 Restricciones de conservación de flujo - caso fraccional

Las restricciones de conservación de flujo en (2b) (utilizando variables $\hat{x}_{de}$) consisten en una restricción por demanda, y por nodo de red ($|\mathcal{D}| \times |\mathcal{E}|$ restricciones).

Todas las restricciones se pueden representar por una única igualdad matricial como sigue:

$$A_{ne} \times x'_{de} = A_{nd} \tag{4}$$

donde:

- A_{ne} es la matriz de incidencia de enlaces. Es una matriz $|\mathcal{N}| \times |\mathcal{E}|$ con una fila por nodo y una columna por enlace. La coordenada (n, e) es 1 si el enlace e es un enlace saliente de n, -1 si es un enlace entrante a n, y 0 en otro caso.
  - La matriz de inicencia de enlaces se puede obtener como una matriz dispersa en Net2Plan utilizando el método de la clase `NetPlan`:

    `getMatrixNodeLinkIncidence`
- x_{de} es una matriz $|\mathcal{D}| \times |\mathcal{E}|$, y x'_{de} su transpuesta, con las variables de decisión.
- A_{nd} es la matriz de incidencia de demandas. Es una matriz $|\mathcal{N}| \times |\mathcal{D}|$ con una fila por nodo, y una columna por demanda. La coordenada (n, d) es 1 si la demanda d es una demanda saliente de n, -1 si es una demanda entrante a n, y 0 en otro caso.
  - La matriz de incidencia de demandas se puede obtener como una matriz dispersa en Net2Plan utilizando el método de la clase `NetPlan`:

    `getMatrixNodeDemandIncidence`

9.2 Restricciones de capacidad de los enlaces - caso fraccional

Las restricciones de capacidad de los enlaces en (2) (utilizando las variables $\hat{x}_{de}$) consisten en una restricción por enlace ($|\mathcal{E}|$ restricciones):

Todas las restricciones se pueden representar por una única desigualdad vectorial como sigue:

$$h \times x_{de} \leq u \tag{5}$$

donde:

- h es un vector fila $1 \times |\mathcal{D}|$ con el tráfico ofrecido de cada demanda.
  - El vector de tráfico ofrecido se puede obtener en Net2Plan utilizando el método de la clase `NetPlan`:

    `getVectorDemandOfferedTraffic`
- u es un vector fila $1 \times |\mathcal{E}|$ con la capacidad de cada enlace.
  - El vector de capacidades de los enlaces se puede obtener en Net2Plan utilizando el método de la clase `NetPlan`:

    `getVectorLinkCapacity`

Quiz 5. Reescribir las restricción de conservación de flujo y de capacidad de los enlaces en Quiz 3 utilizando su forma matricial. Recuerde que el operador de JOM `*` implementa la multiplicación de matrices estándar.

Quiz 6. Reescribir las restricciones de conservación de flujo y capacidad de los enlaces en (1) (con variables x_{de}) utilizando expresiones matriciales con JOM.

- Para las restricciones de conservación de flujo, utilizar la expresión:

  $$A_{ne} \times x'_{de} = A_{nd} \times \mathbf{diag}(h)$$

  donde $\mathbf{diag}(h)$ es una matriz diagonal con los tráficos ofrecidos de las demandas en la diagonal.
- Para las restricciones de capacidad de los enlaces utilizar la expresión:

  $$\sum_d x_{de} \leq u$$

  y notar que la expresión JOM `sum (x_de , 1)`, suma a lo largo de las filas de la matriz, y produce un vector $1 \times |\mathcal{E}|$ con el tráfico cursado en cada enlace.

10 Trabajo para casa después de la sesión de laboratorio

El estudiante debería completar todos los *Quizs* que no haya podido finalizar durante la sesión en el laboratorio.

Bibliography

[1] *P. Pavón Mariño, "Optimization of computer networks. Modeling and algorithms. A hands-on approach", Wiley 2016.*