Problema de Transporte: Restricción de Fuente Única *

Sergio García Prado

sergio@garciparedes.me

1 de abril de 2018

Resumen

[TODO]

1. Introducción

El *Problema de Transporte* se considera bien conocido y existe una extensa literatura dedicada a él dado que surge de manera natural en un gran número de fenómenos de distinta índole. El problema básico se basa en la búsqueda de la mejor planificación de abastecimiento sobre un conjunto de destinos a partir de un conjunto de fuentes de origenes, asumiendo un determinado coste unitario así como unas restricciones de capacidad y demanda para cada punto de origen y destino respectivamente.

Matemáticamente este problema se puede modelar como un problema de programación lineal de la siguiente manera: Asumiremos que existen n puntos de destino cada uno de ellos con una demanda $d_j,\ \forall j \in \{1, ..., n\}$, que deben ser abastecidas a partir de m puntos de origen, con una capacidad denotada por $s_i, \forall i \in \{1, ..., m\}$. Además, existe un determinado coste asociado $c_{i,j}$ por llevar una unidad de producto entre el origen i y el destino j. Para resolver este problema se utiliza una matriz de variables de decisión positivas de dimensión nxm donde la variable x_{ij} determina la cantidad de producto enviado del origen i al destino j. El modelo matemático para este problema se define en la ecuación (1).

$$
\begin{array}{ll}
\text{Minimizar} & \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} \cdot x_{ij} \\
\text{sujeto a} & \sum_{j=1}^{n} x_{ij} \leq s_i, \quad \forall i \in \{1, ..., m\} \\
& \sum_{i=1}^{m} x_{ij} \geq d_j, \quad \forall j \in \{1, ..., n\} \\
& x_{ij} \geq 0, \qquad \forall i \in \{1, ..., m\}, \forall j \in \{1, ..., n\}
\end{array}
\tag{1}
$$

Ecuación 1: Formulación básica del *Problema de Transporte*.

El modelo de transporte de la ecuación (1) tiene una importante propiedad que cumplen algunos problemas de programación lineal denominada *Propiedad de Integridad*, que asegura que si todas las constantes del problema (costes, capacidades y demandas máximas) son enteras, entonces las variables de decisión tomarán valores enteros en en la solución óptima.

Nótese que este problema se puede aplicar a un gran número de situaciones que se dan en la vida real. Desde el caso más destacado, consistente en planificar el abastecimiento de mercancias entre puntos de origen y destino para minimizar costes, a otras como la diversificación de proyectos entre distintos equipos de trabajo en una empresa para reducir costes.

Sin embargo, el problema básico descrito en la ecuación (1) es muy restringido a la hora de modelar situaciones reales más complejas. Por tanto, en este caso se va a describir una ampliación de este que permite limitar el número de fuentes de origen para cada destino (en este caso a una única fuente).

*URL: `https://github.com/garciparedes/single-source-transportation-problem`

La descripción de esta nueva restricción se llevará a cabo en la Sección 2. Después, en la Sección 3 se aplicará el problema a un caso concreto con $m = 8$ orígenes y $n = 12$ destinos. Finalmente, se comentarán distintas cuestiones acerca de esta restricción en la Sección 4. Por último, en el Apéndice A se incluye el código fuente utilizado para la realización de las distintas pruebas, mientras que en el Apéndice B se incluyen los datos concretos utilizados para verificar el correcto funcionamiento del modelo descrito.

2. Planteamiento: Restricción de Fuente Única

[TODO]

2.1. Modelo 1

[TODO]

$$
\begin{array}{lll}
\text{Minimizar} & \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij} \cdot x_{ij} & \\
\text{sujeto a} & \sum_{j=1}^{n} x_{ij} \leq s_i, & \forall i \in \{1, ..., n\} \\
& \sum_{i=1}^{m} x_{ij} \geq d_j, & \forall j \in \{1, ..., n\} \\
& x_{ij} \leq min(s_i, d_j) \cdot y_{ij}, & \forall i \in \{1, ..., m\}, \forall j \in \{1, ..., n\} \\
& \sum_{i=1}^{m} y_{ij} \leq 1, & \forall j \in \{1, ..., n\} \\
& x_{ij} \geq 0, & \forall i \in \{1, ..., m\}, \forall j \in \{1, ..., n\} \\
& y_{ij} \in \{0, 1\}, & \forall i \in \{1, ..., m\}, \forall j \in \{1, ..., n\}
\end{array}
\tag{2}
$$

Ecuación 2: Formulación del *Problema de Transporte de Fuente Única* siguiendo la *Modelización 1.*

2.2. Modelo 2

[TODO]

$$
\begin{array}{lll}
\text{Minimizar} & \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij} \cdot d_j \cdot y_{ij} & \\
\text{sujeto a} & \sum_{j=1}^{n} d_j \cdot y_{ij} \leq s_i, & \forall i \in \{1, ..., n\} \\
& \sum_{i=1}^{m} y_{ij} = 1, & \forall j \in \{1, ..., n\} \\
& y_{ij} \in \{0, 1\}, & \forall i \in \{1, ..., m\}, \forall j \in \{1, ..., n\}
\end{array}
\tag{3}
$$

Ecuación 3: Formulación del *Problema de Transporte de Fuente Única* siguiendo la *Modelización 2.*

3. Resultados

[TODO]

Coste Mínimo $c = 17003$		Destinos $n = 12$											
		1	2	3	4	5	6	7	8	9	10	11	12
Orígenes $m = 8$	1	64	0	0	0	0	3	0	0	0	0	0	109
	2	0	0	0	88	0	0	29	0	0	0	0	0
	3	0	0	0	0	0	0	0	0	82	0	0	0
	4	0	0	0	0	0	0	39	0	0	0	113	0
	5	5	0	0	0	95	0	0	0	0	78	0	0
	6	0	0	95	1	0	9	0	0	0	0	0	0
	7	0	0	0	0	0	15	0	112	0	0	0	0
	8	0	98	0	0	0	77	0	0	0	0	0	0

Tabla 1: Solución óptima para el problema aplicando la relajación lineal de varias fuentes.

Coste Mínimo $c = 21942$		Destinos $n = 12$											
		1	2	3	4	5	6	7	8	9	10	11	12
Orígenes $m = 8$	1	69	0	0	0	0	104	0	0	0	0	0	0
	2	0	0	0	89	0	0	68	0	0	0	0	0
	3	0	0	0	0	0	0	0	0	82	0	0	109
	4	0	0	0	0	0	0	0	0	0	0	113	0
	5	0	0	0	0	95	0	0	0	0	78	0	0
	6	0	0	95	0	0	0	0	0	0	0	0	0
	7	0	0	0	0	0	0	0	112	0	0	0	0
	8	0	98	0	0	0	0	0	0	0	0	0	0

Tabla 2: Solución óptima para el problema aplicando la restricción de fuente única.

4. Conclusiones

[TODO]

Apéndice A Código Fuente

A.1 Problema de Transporte con Fuente Única: Relajación Lineal

```
model "single-source-transportation-relaxation"
    !-------------------------------------
    ! Single Source Transportation - Relaxation
    ! Sergio García Prado - garciparedes.me
    ! April 2018
    !-------------------------------------

    uses "mmxprs";

    declarations
        n, m: integer
    end-declarations
    initializations from "data.dat"
        n m
    end-initializations
    declarations
        origins = 1..m
        destinations = 1..n
        offer: array(origins) of real
        demand: array(destinations) of real
        cost: array(origins, destinations) of real
        x: array(origins, destinations) of mpvar
    end-declarations
    initializations from "data.dat"
        offer demand cost
    end-initializations

    forall(i in origins) do
        res_ori(i) := sum(j in destinations) x(i, j) <= offer(i)
    end-do
    forall(j in destinations) do
        res_dest(j) := sum(i in origins) x(i, j) >= demand(j)
    end-do
    objetive := sum(i in origins, j in destinations) x(i, j) * cost(i, j)
    minimize(objetive)

    writeln("objetive = ", getobjval)
    writeln
    forall(i in origins) do
        writeln
        forall(j in destinations) do
            write(getsol(x(i,j)), "\t")
        end-do
    end-do
end-model
```

A.2 Problema de Transporte con Fuente Única: Modelo 1

```
model "single-source-transportation-model-1"
    !-------------------------------------
    ! Single Source Transportation - Model 1
    ! Sergio García Prado - garciparedes.me
    ! April 2018
    !-------------------------------------

    uses "mmxprs";

    declarations
        n, m: integer
    end-declarations
    initializations from "data.dat"
        n m
    end-initializations
    declarations
        origins = 1..m
```

```
        destinations = 1..n
        offer: array(origins) of real
        demand: array(destinations) of real
        cost: array(origins, destinations) of real
        x: array(origins, destinations) of mpvar
        y: array(origins, destinations) of mpvar
    end-declarations
    initializations from "data.dat"
        offer demand cost
    end-initializations

    forall(i in origins, j in destinations) do
        y(i, j) is_binary
        res_logic(i, j) := x(i, j) <= minlist(offer(i), demand(j)) * y(i, j)
    end-do
    forall(i in origins) do
        res_ori(i) := sum(j in destinations) x(i, j) <= offer(i)
    end-do
    forall(j in destinations) do
        res_dest(j) := sum(i in origins) x(i, j) >= demand(j)
        res_single(j) := sum(i in origins) y(i, j) <= 1
    end-do
    objetive := sum(i in origins, j in destinations) x(i, j) * cost(i, j)
    minimize(objetive)

    writeln("objetive = ", getobjval)
    writeln
    forall(i in origins) do
        writeln
        forall(j in destinations) do
            write(getsol(x(i,j)), "\t")
        end-do
    end-do

end-model
```

A.3 Problema de Transporte con Fuente Única: Modelo 2

```
model "single-source-transportation-model-2"
    !-------------------------------------
    ! Single Source Transportation - Model 2
    ! Sergio García Prado - garciparedes.me
    ! April 2018
    !-------------------------------------

    uses "mmxprs";

    declarations
        n, m: integer
    end-declarations
    initializations from "data.dat"
        n m
    end-initializations
    declarations
        origins = 1..m
        destinations = 1..n
        offer: array(origins) of real
        demand: array(destinations) of real
        cost: array(origins, destinations) of real
        y: array(origins, destinations) of mpvar
    end-declarations
    initializations from "data.dat"
        offer demand cost
    end-initializations

    forall(i in origins, j in destinations) do
        y(i, j) is_binary
    end-do
    forall(i in origins) do
        res_ori(i) := sum(j in destinations) demand(j) * y(i, j) <= offer(i)
    end-do
    forall(j in destinations) do
        res_dest(j) := sum(i in origins) y(i, j) = 1
    end-do
    objetive := sum(i in origins, j in destinations) demand(j) * y(i, j)* cost(i, j)
```

```
    minimize(objetive)

    writeln("objetive = ", getobjval)
    writeln
    forall(i in origins) do
        writeln
        forall(j in destinations) do
            write(getsol(y(i,j)) * demand(j), "\t")
        end-do
    end-do
end-model
```

Apéndice B Datos

```
! Problema de transporte con fuente única
! Datos de un ejemplo con m = 8 y n = 12.
! Hay que resolver el problema de transporte normal y el problema con fuente única
! calculando el incremento en el coste total
m: 8
n: 12
offer:[176 163 192 152 178 105 127 175]

demand:[69 98 95 89 95 104 68 112 82 78 113 109]

cost:[
 6 78 31 54 56 34 83 76 74 67 62 46
 96 30 94 6 59 99 34 86 41 77 89 95
 84 63 41 94 63 57 55 76 3 95 54 62
 65 94 23 56 99 70 5 71 68 97 7 53
 8 44 89 56 14 70 81 97 59 43 80 96
 98 78 7 4 32 37 35 93 59 74 56 52
 4 12 32 9 29 7 18 17 34 15 61 57
 99 1 67 82 24 12 72 53 52 44 78 49]

! Solución: el incremento en el costo es de 4939 ( pasa de 17003 a 21942)
```

Referencias

[FIC] FICO Xpress. Xpress-Mosel. http://www.maths.ed.ac.uk/hall/Xpress/FICO_Docs/mosel/mosel_lang/dhtml/moselref.html/.

[GP18] Sergio García Prado. Single Source Transportation Problem, 2018. https://github.com/garciparedes/single-source-transportation-problem.

[SA18] Jesús Sáez Aguado. Programación Entera, 2017/18. Facultad de Ciencias: Departamento de Estadística e Investigación Operativa.