Problema de Transporte: Restricción de Fuente Única [*]

Sergio García Prado

sergio@garciparedes.me

1 de abril de 2018

Resumen

[TODO]

1. Introducción

[TODO]

2. Planteamiento: Restricción de Fuente Única

[TODO]

2.1. Modelo 1

[TODO]

2.2. Modelo 2

[TODO]

3. Conclusiones

[TODO]

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

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.

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. Network Flow Transeuro, 2018. https://github.com/garciparedes/network-flow-transeuro.

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