

Revisión del Model

Yimin Pan, L12

Revisado a: Pablo Carbonell (Problema 15)

November 7, 2019

1 Juegos de Prueba para el Modelo Revisado

1. Juego de prueba 1

El primer juego de prueba consiste en un problema de 3 franjas horarias y 3 compañías. Los otros parametros, tales como precios que paga la compañía por el paquete, los minutos que pide la compañía para cada franja horaria... són generados aleatoriamente.

En este caso vemos que las compañías ofrecen mucho más que si vendemos los minutos por separado. Por ejemplo, para k_0 podemos ver que vendiendo los minutos que pide esta compañía a minoristas, obtenemos mucho menos: $42*0.04+56*1.43+64*1.21 \approx 158$, que es menor que los 72947 que ofrece k_0 . Igual para las otras compañías, siempre y cuando recordando escoger la compañía que más paga en relación a los minutos que pide. En este caso k_1 .

La solución que muestra el solver coincide con lo que habíamos pensado, acepta el paquete de k_1 y los minutos restantes se lo vende a minorista. Por lo que obtenemos una función objetivo de 73238.

2. Juego de prueba 2

En el segundo juego de prueba intentaremos balancear un poco la oferta de las compañías, que no haya tanta diferencia. El número de compañía no afecta mucho en ver si el modelo funciona o no, por lo que lo dejamos igual. En cuanto al precio para los minoristas deberá ser baja, ya que sinó la solución es trivial, lo vendemos todo a los minoristas.

La solución óptima debería ser vender a k_0 y k_2 ya que ambos piden 40 minutos para cada franja, por lo que suman 80 en cada franja, pagando $240+240 = 480$ que es mayor que los 340 que ofrece k_1 por 90 en cada franja.

```

1  set Kompanies := K0 K1 K2;
2  set Periods := P0 P1 P2;
3
4  param Bundle_price :=
5  K0 72947
6  K1 470631
7  K2 854927
8  ;
9
10 param Bundle_minutes: K0 K1 K2 :=
11 P0 42 56 64
12 P1 42 99 84
13 P2 78 77 13
14 ;
15
16 param Period_price :=
17 P0 0.04067369732333814
18 P1 1.4309955623200288
19 P2 1.2145677510765052
20 ;
21
22 param Period_minutes :=
23 P0 214
24 P1 59
25 P2 292
26 ;
27

```

Figure 1: Juego 1

```

ampl: solve;
Gurobi 5.1.0: optimal solution; objective 73238.2403
ampl: display choose_bundle;
choose_bundle [*] :=
K0 1
K1 0
K2 0
;

ampl: display minutes_period;
minutes_period [*] :=
P0 172
P1 17
P2 214
;

```

Figure 2: Solución Juego 1

La solución que muestra el solver es correcta, con la estrategia indicada anteriormente obtenemos $480 + 1 \cdot 20 \cdot 3 = 540$, igual a la función objetivo del solver.

```

set Kompanies := K0 K1 K2;
set Periods := P0 P1 P2;

param Bundle_price :=
K0 240
K1 340
K2 240
;

param Bundle_minutes: K0 K1 K2 :=
P0 40 90 40
P1 40 90 40
P2 40 90 40
;

param Period_price :=
P0 1.0
P1 1.0
P2 1.0
;

param Period_minutes :=
P0 100
P1 100
P2 100
;

```

Figure 3: Juego 2

2 Aspectos Mejorables del Modelo

Dado que se trata de problemas simples, el modelo se ajusta bastante bien a lo que pide el enunciado. Como hemos visto en los juegos de prueba anteriores la solución que muestra el solver coincide con lo que pensamos.

Un aspecto a mencionar es que, no se sabe por qué razón a pesar de que la variable $choose_{bundle}$ se ha definido como binario en el modelo, si no entramos el comando *optionsolvergurobi*; en AMPL, lo cual elegimos un solver en específico. Esa variable puede adquirir valores reales, y eso no es posible para este problema, o vendemos todo el paquete a una compañía, o no lo vendemos.

```

ampl: solve;
Gurobi 5.1.0: optimal solution; objective 540
ampl: display choose_bundle;
choose_bundle [*] :=
K0 1
K1 0
K2 1
;

ampl: display minutes_period;
minutes_period [*] :=
P0 20
P1 20
P2 20
;

```

Figure 4: Solución juego 2

3 Variante del Modelo

La modificación que tiene se ha hecho respecto al modelo anterior son:

- Se ha eliminado la variable *minutes_period* que indica los minutos restantes despues de venderlos a las companias, ya que esta variables se puede calcular simplemente restando para cada periodo los minutos que dispones con los vendidos a las companias.
- Una variable *moneyFromCompanies* que indica la ganancia que proviene de las companias.
- Una variable *moneyFromRetail* que indica los beneficios que se ha obtenido vendiendo los minutos de cada periodo separado.
- La función objetivo *revenue* ahora se calcula como la suma de *moneyFromCompanies* y *moneyFromRetail*.

```

set Kompanies;
set Periods;

param Bundle_price {Kompanies} >= 0 default 0;
param Bundle_minutes {Periods, Kompanies} >= 0 default 0;
param Period_price {Periods} >= 0 default 0;
param Period_minutes {Periods} >= 0;

var choose_bundle {Kompanies} binary;

var moneyFromCompanies =
    sum{k in Kompanies} ( Bundle_price[k]*choose_bundle[k] );

var moneyFromRetail =
    sum{i in Periods} ((Period_minutes[i] - sum{k in Kompanies} ( Bundle_minutes[i,k]*choose_bundle[k])) * Period_price[i]);

maximize revenue : moneyFromRetail+moneyFromCompanies;

s.t. time_not_negative{i in Periods} :
    Period_minutes[i] - sum{k in Kompanies} ( Bundle_minutes[i,k]*choose_bundle[k] ) >= 0;

```

Figure 5: Variante del modelo

En teoría con este variante deberiamos poder maximizar la ganancia total, venderemos los minutos a las companias si es rentable y actualizamos los minutos

que dispones. La restricción de no tener tiempo de franja negativa, hace que no podamos vender a las compañías cuando no tenemos suficiente tiempo como lo que piden. Dicho esto, deberíamos poder obtener resultados iguales que el modelo original con los juegos de prueba que hemos hecho.

```
ampl: solve;
MINOS 5.5: ignoring integrality of 1 variables
MINOS 5.5: optimal solution found.
1 iterations, objective 73238.2403
ampl: display moneyFromCompanies;
moneyFromCompanies = 72947

ampl: display moneyFromRetail;
moneyFromRetail = 291.24
```

Figure 6: Solución juego 1 por modelo 2

```
ampl: solve;
Gurobi 5.1.0: optimal solution; objective 540
ampl: display moneyFromCompanies;
moneyFromCompanies = 480

ampl: display moneyFromRetail;
moneyFromRetail = 60
```

Figure 7: Solución juego 2 por modelo 2

Como podemos observar los resultados coincide con la solución que daba el modelo original y lo que habíamos argumentado en el apartado de juego de pruebas. Dicho esto, se vé que este variante del modelo es correcto.