FALL 2022-2023

IE 303: MODELING AND METHODS IN OPTIMIZATION

**Date:** 16.12.2022

# SECOND COMPUTING PROJECT REPORT


YİĞİT ALİ KARADOĞAN - 21902218

EMRE GÖKTEPE - 21902632

DEPARTMENT OF INDUSTRIAL ENGINEERING

BILKENT UNIVERSITY

06800 ANKARA


**Instructor:** Mustafa Çelebi Pınar


THE FACULTY OF ENGINEERING

BILKENT UNIVERSITY

06800 ANKARA

## PROBLEM DEFINITION

We sell a collection of classical music CDs online courtesy to a web page. There are various guidelines that will govern the selling process. For instance, every prospective buyer submits their valuations online, and they are all compiled in a valuation matrix. Every client is only permitted to purchase one CD (or none at all, as we do not wish to transfer whole CD collection to another collecting individual or store), as CDs are uncommon collection goods. The site is seeking to maximize its overall earnings according to specific guidelines.

Our task is to provide a linear integer programming model for the internet site's decision-making process over which consumers receive CDs and at what price in order to maximize total revenue.

## INTEGER MODEL OF THE PROBLEM

$\underline{Parameters}$:

$$v_{(i,j)} = customer\ i's\ valuation\ for\ cd\ j \qquad \forall i = 1, \cdots n, \forall j = 1, \cdots m$$

$\underline{Decision\ Variables}$:

$$y_{(a,b)} = \begin{cases} 1 & if\ customer\ a\ gets\ cd\ b \\ 0 & o.w. \end{cases} \qquad \forall a = 1, \cdots n, \forall b = 1, \cdots m$$

$$p_b = price\ of\ cd\ j \qquad \forall b = 1, \cdots m$$

$\underline{Model}$:

$$\max \quad \sum_{b=1}^{m} p_b$$

$s.t.$

(1) $$\sum_{a=1}^{n} y_{(a,b)} = 1 \qquad \forall b = 1, \cdots m$$

(2) $$\sum_{b=1}^{m} y_{(a,b)} \leq 1 \qquad \forall a = 1, \cdots n$$

(3) $$-v_{(a,b)} + v_{(a,c)} + p_b - p_c \leq 40 * \left(1 - y_{(a,b)}\right)$$

$$\forall a = 1, \cdots n \,,\ \forall b = 1, \cdots m \,, \forall c = 1, \cdots m \,,\ \forall c \neq b$$

(4) $$-v_{(a,b)} + p_b \leq 20 * \left(1 - y_{(a,b)}\right) \qquad \forall a = 1, \cdots n \,, \forall b = 1, \cdots m$$

(5) $$y_{(a,b)} \in \{0,1\} \qquad \forall a = 1, \cdots n \,, \forall b = 1, \cdots m$$

(6) $$p_b \geq 0 \qquad \forall b = 1, \cdots m$$

## XPRESS CODE OF INTEGER MODEL OF THE PROBLEM

Additional comments of constraints explanations are supplied within code. Further explanation of Big-M type constraints are also supplied.

- Objective Function:

```
! maximize its total revenue
objective:= sum(b in m)p(b)
! objective function is max type
maximize(objective)
```

- Constraint (1):

```
! every CD must be assigned
forall(b in m)
        sum (a in n)y(a, b) = 1
```

- Constraint (2):

```
!every customer can only get one CD (or none at all)
forall(a in n)
        sum (b in m)y(a, b) <= 1
```

- Constraint (3):

```
!If a customer i is assigned a CD j at a price p_j then this must be one,
!of the most advantageous CDs for him/her in terms of utility
forall(a in n, b in m, c in m)
    -v(a,b)+v(a,c)+p(b)-p(c) <= 40*(1-y(a,b))
```

<u>Justification of Big-M:</u> We found the upper bound Big-M 40 as 20+v(i,j). Since the expression v(i,j) cannot exceed 20, we set upper bound to 40.

- Constraint (4):

```
!If a customer i is assigned a CD j at price p_j
!then his/her utility must be non-negative.
forall(a in n, b in m)
    -v(a,b)+p(b) <= 20*(1-y(a,b))
```

<u>Justification of Big-M:</u> Again if customer does not buy a CD, then Big-M consonant 20 is our upper bound since v(i,j) cannot exceed 20.

- Constraint (5):

```
! whether customer gets the CD or not is a binary variable
forall(a in n , b in m) do
    y(a,b) is_binary
    end-do
```

- Creating an uniformly distributed valuation matrix V:

```
! v is the valuations matrix
forall ( a in n, b in m)
! random function' returns a float value between 0 and 1.
    v(a,b) := random*19 + 1
```

## CASE 1: m = 10, n = 20

We have 10 CDs and 20 customers. We set m = 10, and n = 20 in our code in declaration part as follows, the remaining parts are same:

```
!every potential buyer i = 1, . . . , n
n = 1..20
!Your collection has m CDs
m = 1..10
```

Here, there are some example valuations, between 1 and 20, of customer 2 and customer 7 for CDs consisting of 10 pieces:

```
v(2,1)=13.64148873
v(2,2)=3.407516995
v(2,3)=5.120534711
v(2,4)=10.9790554
v(2,5)=11.24086979
v(2,6)=13.13812998
v(2,7)=1.766849591
v(2,8)=5.035434819
v(2,9)=7.274549502
v(2,10)=5.638221333
```

Valuations of customer 2

```
v(7,1)=12.39421651
v(7,2)=16.31896212
v(7,3)=18.61923876
v(7,4)=13.05316648
v(7,5)=11.28947094
v(7,6)=3.830179686
v(7,7)=4.179128442
v(7,8)=16.50720734
v(7,9)=17.13812042
v(7,10)=8.751280994
```

Valuations of customer 7

According to our model, there are some examples of some customers who bought some CDs or not. As you can see, CD 1 sold to customer 14 and customer 2 did not buy anything:

```
y(14,1)=1
y(14,2)=0
y(14,3)=0
y(14,4)=0
y(14,5)=0
y(14,6)=0
y(14,7)=0
y(14,8)=0
y(14,9)=0
y(14,10)=0
```

CD 14 sold to Customer 1

```
y(2,1)=0
y(2,2)=0
y(2,3)=0
y(2,4)=0
y(2,5)=0
y(2,6)=0
y(2,7)=0
y(2,8)=0
y(2,9)=0
y(2,10)=0
```

Customer 2 did not buy

Here, you can see the optimal price for maximum revenue of each CD and total revenue:

```
p(1)=19.34098037
p(2)=16.44210681
p(3)=19.79588179
p(4)=19.72314479
p(5)=18.13616754
p(6)=17.97038955
p(7)=18.13389478
p(8)=17.05924268
p(9)=19.79807512
p(10)=19.57938344
Total Revenue: 185.9792669
```

Prices of each CD and Total Revenue

## CASE 2: m = 15, n = 20

Now, we  have 15 CDs and # of customers did not change. We set m = 15, and n = 20 in our code in declaration part as follows, the remaining parts are same:

```
!every potential buyer i = 1, . . . , n
n = 1..20
!Your collection has m CDs
m = 1..15
```

Here, there are some example valuations, between 1 and 20, of customer 4 and customer 16 for CDs consisting of 15 pieces:

```
v(4,1)=3.810582649
v(4,2)=6.368792397
v(4,3)=3.74613261
v(4,4)=11.03164251
v(4,5)=4.636127549
v(4,6)=11.37297778
v(4,7)=3.733160116
v(4,8)=12.52621316
v(4,9)=11.77531619
v(4,10)=18.77924277
v(4,11)=19.04884212
v(4,12)=5.129547955
v(4,13)=8.274931884
v(4,14)=15.90463674
v(4,15)=12.72346703
```

Valuations of customer 4

```
v(16,1)=18.00945626
v(16,2)=19.21801135
v(16,3)=12.49070172
v(16,4)=7.302356176
v(16,5)=15.65834268
v(16,6)=3.998087083
v(16,7)=17.41609462
v(16,8)=3.799025288
v(16,9)=9.758970659
v(16,10)=11.58745151
v(16,11)=5.496987953
v(16,12)=13.30278562
v(16,13)=2.995466117
v(16,14)=9.76549772
v(16,15)=14.89527224
```

Valuations of customer 16

According to our model, there are some examples of some customers who bought some CDs or not. As you can see, CD 14 sold to customer 15 and customer 8 did not buy anything:

```
y(15,1)=0
y(15,2)=0
y(15,3)=0
y(15,4)=0
y(15,5)=0
y(15,6)=0
y(15,7)=0
y(15,8)=0
y(15,9)=0
y(15,10)=0
y(15,11)=0
y(15,12)=0
y(15,13)=0
y(15,14)=1
y(15,15)=0
```

CD 14 sold to Customer 15

```
y(8,1)=0
y(8,2)=0
y(8,3)=0
y(8,4)=0
y(8,5)=0
y(8,6)=0
y(8,7)=0
y(8,8)=0
y(8,9)=0
y(8,10)=0
y(8,11)=0
y(8,12)=0
y(8,13)=0
y(8,14)=0
y(8,15)=0
```

Customer 8 did not buy

Here, you can see the optimal price for maximum revenue of each CD and total revenue:

```
p(1)=18.59883754
p(2)=19.21801135
p(3)=18.42074923
p(4)=19.4662852
p(5)=18.39092092
p(6)=19.13123613
p(7)=17.46116978
p(8)=18.54425509
p(9)=18.46636092
p(10)=18.77924277
p(11)=19.72836952
p(12)=18.93175878
p(13)=19.60159721
p(14)=17.66437877
p(15)=17.74448817
Total Revenue: 280.1476614
```

Prices of each CD and Total Revenue

## CASE 3: m = 19, n = 20

Now, we  have 19 CDs and # of customers did not change. We set m = 19, and n = 20 in our code in declaration part as follows, the remaining parts are same:

```
!every potential buyer i = 1, . . . , n
n = 1..20
!Your collection has m CDs
m = 1..19
```

Here, there are some example valuations, between 1 and 20, of customer 6 and customer 18 for CDs consisting of 19 pieces:

```
v(6,1)=1.959449019
v(6,2)=2.675674235
v(6,3)=15.83120302
v(6,4)=4.726927687
v(6,5)=2.471341804
v(6,6)=15.41777171
v(6,7)=13.48911115
v(6,8)=6.056864814
v(6,9)=6.87290612
v(6,10)=6.730543216
v(6,11)=6.700874333
v(6,12)=17.59491693
v(6,13)=8.401930337
v(6,14)=15.47364023
v(6,15)=12.15361644
v(6,16)=8.029143154
v(6,17)=16.23082193
v(6,18)=10.51846012
v(6,19)=1.677126599
```

Valuations of customer 6

```
v(18,1)=6.281692814
v(18,2)=10.29740457
v(18,3)=17.98738828
v(18,4)=2.977526132
v(18,5)=9.269588715
v(18,6)=15.07014834
v(18,7)=3.548093451
v(18,8)=1.780636282
v(18,9)=8.485060708
v(18,10)=9.389615685
v(18,11)=12.45504904
v(18,12)=18.49708155
v(18,13)=13.30992863
v(18,14)=17.8654098
v(18,15)=13.40469597
v(18,16)=7.402147838
v(18,17)=12.90486668
v(18,18)=14.57107735
v(18,19)=5.508564637
```

Valuations of customer 18

According to our model, there are some examples of some customers who bought some CDs or not. As you can see, CD 13 sold to customer 5 and the only customer did not buy anything is 15:

```
y(5,1)=0
y(5,2)=0
y(5,3)=0
y(5,4)=0
y(5,5)=0
y(5,6)=0
y(5,7)=0
y(5,8)=0
y(5,9)=0
y(5,10)=0
y(5,11)=0
y(5,12)=0
y(5,13)=1
y(5,14)=0
y(5,15)=0
y(5,16)=0
y(5,17)=0
y(5,18)=0
y(5,19)=0
```

CD 13 sold to Customer 5

```
y(15,1)=0
y(15,2)=0
y(15,3)=0
y(15,4)=0
y(15,5)=0
y(15,6)=0
y(15,7)=0
y(15,8)=0
y(15,9)=0
y(15,10)=0
y(15,11)=0
y(15,12)=0
y(15,13)=0
y(15,14)=0
y(15,15)=0
y(15,16)=0
y(15,17)=0
y(15,18)=0
y(15,19)=0
```

The only customer who did not buy any CD

Here, you can see the optimal price for maximum revenue of each CD and total revenue:

```
p(1)=14.39357621
p(2)=19.16809291
p(3)=17.08522365
p(4)=16.31607839
p(5)=18.5941988
p(6)=16.33430331
p(7)=17.13535122
p(8)=17.5856722
p(9)=15.925297
p(10)=17.61745764
p(11)=15.05581539
p(12)=17.59491693
p(13)=19.40861918
```

:ps://neos-server.org/neos/jobs/12530000/125

:.12.2022 22:19

```
p(14)=17.64059543
p(15)=16.71258775
p(16)=16.18812185
p(17)=16.45875933
p(18)=19.92824541
p(19)=17.19222398
Total Revenue: 326.3351366
```

Prices of each CD and Total Revenue

As it can be seen in all three cases than as the number of CDs available to sell increases, total revenue also increases.