UCD MICHAEL SMURFIT GRADUATE BUSINESS SCHOOL

MIS40520: Analytical Business Modelling

M(I)LP Assignment 2

Authors:
Goyal Shruti (16200726)
Gupta Deepak Kumar (16200660)

Lecturer:
Dr Paula Carroll

March 16, 2017

Executive Summary

Optimal Solution for Conference Management

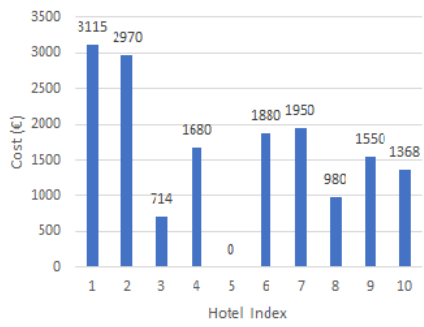
Shruti Goyal, Deepak Kumar Gupta and Dr Paula Carroll

Out of 185 available rooms in 10 four star hotel, 150 rooms allocated allocated to delegates using M(I)LP, Mixed integer linear programming optimization model. For given business problem for minimizing organiser cost while maintain average satisfactory rating of each hotel ,minimum cost found is €16207 subject to given constraints and average customer satisfaction rating is 8.30067 for this solution.

Hotel Index	1	2	3	4	5	6	7	8	9	10	Total
Price (Euro)	89	99	119	112	143	94	130	98	155	152	-
Room Availability	35	30	15	15	15	20	15	10	10	20	185
Room Allocated	35	30	6	15	0	20	15	10	10	9	150
Cost	€3115	€2970	€714	1680	€0	€1880	€1950	€980	€1550	€1368	€16207

Table 1: Number of Rooms allocated to Delegates in each Hotel

(a) Rooms Availability vs Allocated



(b) Total Cost for Each Hotel (€)

Figure 1: Optimal cost and number of rooms allocated

Abstract

Optimal Solution for Conference Management

Shruti Goyal, Deepak Kumar Gupta and Dr Paula Carroll

A summary of major research/methodological contributions. Used in research papers and documents.

1. Background

You have been asked to help organise hotel accommodation for 150 delegates attending a conference in UCD. You use a hotel booking website to extract the data shown in the Table below for ten four star hotels in Dublin city centre. The Table shows the price for a single room per person per night, the customer satisfaction rating and the number of rooms available for each hotel. The conference organisers want to allocate the delegates to hotels at minimum cost while achieving an average customer satisfaction rate of at least 8.3. Formulate an (I)LP model of this problem.

Hotel Index	1	2	3	4	5	6	7	8	9	10
Price (€)	89	99	119	112	143	94	130	98	155	152
Customer rating	7.8	8.3	8	8.7	8	8.1	8.6	8.9	8.9	8.4
Room Availability	35	30	15	15	15	20	15	10	10	20

Table 2: Hotel Data

2. Introduction

Operation research provides mathematical programming techniques for solving real life problems subject to constraints. Whereas linear programming uses the basis mathematical programming for a linear equation objective function $f(z)$ and a set of constraints equations which are set of linear inequalities or equality constraints.

An integer (linear) programming (IP) problem is a linear programming problem in which some or all of the decision variables are restricted to be integer valued. In some real scenarios we can not allocate fractional of something, for example: $3\,\frac{1}{2}$ teachers required for a class of 100 students, but $3\,\frac{1}{2}$ teachers is not possible. That is why we require integer (linear) programming for solving such real life problems.

2.1. Paper 1 Modeling a cruise line revenue management problem

Li, Yihua, Qing Miao, and Bruce X. Wang. "Modeling a cruise line revenue management problem." Journal of Revenue and Pricing Management 13.3 (2014): 247-260. Cruise line is special tourism service has some similarity and dissimilarity with hotel industry. At a company, revenue management of Cruise line still a research topic in Operation Research, as this industry receive little focused compared to airline and hotel industry. Here, Objective is to maximize the Cruise revenue by sales of rooms for each sailing. Pricing for a room in Cruise is per person basis, whereas in Hotel there is marginal difference between single bed room and double room fare. This is because, Cruise has to make additional arrangement per person in terms of food, consumptions etc.

Dynamic Programming is not suitable for this case because it take longer time to process and difficult to manage with multiple dependent constraints. Therefore, Linear (integer) programming is best optimization model for this case using linear objective equations and linear constraints equations. This model can be repeatedly applied for capacity allocation and pricing.

Capacity Limitation:
1. Ship do not accept more than 2.5 or 3 person per room
2. Number of total seats on lifeboats
3. Number of rooms available for each room category

Similarity & Dissimilarity with given business problem:
Cruise line revenue management require integer programming as a room can not booked in fractions. Each cruise ship has 800 rooms, and each room is classified into 12 segment according to its size, location, view,

capacity etc. Here, we can relate Hotel Index in Conference management problem with room segments, rating of each hotel room with its segment factors i,e size, location, view and room availability is similar to number of available rooms under each segment. However, Cruise line problem is about maximizing the revenue and pricing, whereas conference management problem about minimizing cost while achieving average customer satisfaction rating.

2.2. Paper 2

Baur, A., Klein, R. & Steinhardt, C. OR Spectrum (2014) 36: 557. doi:10.1007/s00291-013-0338-3 Overview
Methods
LP Model
Similarity & Dissimilarity with given business problem:

3. Methodology

We have total delegates 150 and 10 four star hotels to accommodate guests We need to minimize the cost while achieving minimum average rating:
Lets say HI as $Hotel\ Index$:
Mention about non negative equations as well Objective is:

$$minimize(Cost) \tag{1}$$

Constraint 1:

$$Cost = \sum_{HI=1}^{10} Price(HI) * decVar(HI \tag{2}$$

Constraint 2:

$$Avg.Rating = \frac{\sum_{HI=1}^{10} Rating(HI) * decVar(HI))}{150} \tag{3}$$

Constraint 3:

$$decVar(HI) <= Avail(HI), HI \in \{1, 10\} \tag{4}$$

Hotel Index	LP Model	(I)LP Model
1	35	35
2	30	30
3	6.25	6
4	15	15
5	0	0
6	20	20
7	15	15
8	10	10
9	10	10
10	8.75	9
Total Cost	€16198.8	€16207

Table 3: Comparison of Output LP Model with Integer LP Model

4. Analysis of Results

Clearly, optimized cost is different in LP Model and Integer Linear Programming model. It is due to output value of decision variables in (I)LP are integer number and decision variables in LP model are real number. It is clear From Table 3 that value of decision variable for Hotel Index 3 and 10 is different in LP and (I)LP model. Final optimal cost for conference management has a difference of 0.0506% between LP and (I)LP model.

For given problem of conference management, optimal cost of accommodating 150 guests in hotel is €16207 and achieving average customer satisfaction rating of 8.3. Compare non integer and integer solutions
Why interger programming is required
Also take previous assignment example to contrast the need on MILP

5. Conclusions

Appendix

A. Code

```
model Conference_Management
uses "mmxprs"; !gain access to the Xpress-Optimizer solver
declarations
HI = 1..10                          ! Index range
PRICE: array(HI) of integer           ! Price table
RATING: array(HI) of real !Customer Satsifcation rating
AVAIL: array(HI) of integer !Available Rooms in a hotel
decVar: array(HI) of mpvar            !Decision Variables
end-declarations
Max_Guest := 150

!Read in the data from our text file
initializations from 'Conference_Management.txt'
PRICE
RATING
AVAIL
end-initializations

!procedure to check problem status
procedure print_status
declarations
status: string
end-declarations
case getprobstat of
XPRS_OPT: status:="LP Optimum found"
XPRS_UNF: status:="Unfinished"
XPRS_INF: status:="Infeasible"
XPRS_UNB: status:="Unbounded"
XPRS_OTH: status:="Failed"
else status:="???"
end-case
writeln("Problem status: ", status)
end-procedure

!Minimiz cost
Cost:= sum(i in HI) PRICE(i)*decVar(i)

!Constraints
!Declare that our decision variables are integers
forall (i in HI) do
decVar(i) is_integer
end-do

AvgRating:= (sum(i in HI) RATING(i)*decVar(i))/150 >= 8.3

forall( i in HI)
decVar(i) <= AVAIL(i)
sum(i in HI) decVar(i) = Max_Guest
```

```
!Display output of solution values
procedure print_sol
writeln("Begin running model")
writeln("-----------------------------------------------------------")
print_status
writeln("Cost is: âĈň",getobjval)
writeln("-----------------------------------------------------------")
forall(i in HI)
writeln("Passenger in Hotel_",i," -> ", getsol(decVar(i)))

!write value of AvgRating to output
writeln(getsol(AvgRating))
writeln("-----------------------------------------------------------")
exportprob(EP_MIN,"",Cost)
writeln("-----------------------------------------------------------")
exportprob(1,"Conference_Management",Cost)
writeln("End running model")
!Modify Optimizer control parameter PSEUDOCOST
end-procedure
minimize(Cost)
print_sol
end-model
```

B. Input and Output

B.1. Case 1: Original Data (10 Variables)

Input

```
! Data file for `20. Conference Management Assignment - 2'

PRICE: [ 89, 99, 119, 112,143,94,130,98,155,152] !Constraint coefficients

RATING: [7.8,8.3,8.0,8.7,8.0,8.1,8.6,8.9,8.9,8.4] !Constraint coefficients

AVAIL:[35,30,15,15,15,20,15,10,10,20] !Values of the constraints
```

Output

```
Begin running model
-----------------------------------------------------------
Problem status: LP Optimum found
Cost is: âĈň16207
-----------------------------------------------------------
Passenger in Hotel_1 -> 35
Passenger in Hotel_2 -> 30
Passenger in Hotel_3 -> 6
Passenger in Hotel_4 -> 15
Passenger in Hotel_5 -> 0
Passenger in Hotel_6 -> 20
Passenger in Hotel_7 -> 15
Passenger in Hotel_8 -> 10
Passenger in Hotel_9 -> 10
Passenger in Hotel_10 -> 9
0.000666667
```

```
---------------------------------------------------------
\ Using Xpress-MP extensions
Minimize
 89 decVar(1) + 99 decVar(2) + 119 decVar(3) + 112 decVar(4) + 143 decVar(5) +
94 decVar(6) + 130 decVar(7) + 98 decVar(8) + 155 decVar(9) + 152 decVar(10)

Subject To
_R1: decVar(1) + decVar(2) + decVar(3) + decVar(4) + decVar(5) + decVar(6) +
decVar(7) + decVar(8) + decVar(9) + decVar(10) = 150
AvgRating: 0.052 decVar(1) + 0.0553333 decVar(2) + 0.0533333 decVar(3) + 0.058 decVar(4) +
0.0533333 decVar(5) + 0.054 decVar(6) + 0.0573333 decVar(7) + 0.0593333 decVar(8) +
0.0593333 decVar(9) + 0.056 decVar(10) >= 8.3

Bounds
decVar(1) <= 35
decVar(2) <= 30
decVar(3) <= 15
decVar(4) <= 15
decVar(5) <= 15
decVar(6) <= 20
decVar(7) <= 15
decVar(8) <= 10
decVar(9) <= 10
decVar(10) <= 20

Integers
decVar(1) decVar(2) decVar(3) decVar(4) decVar(5) decVar(6) decVar(7) decVar(8)
decVar(9) decVar(10)

End
---------------------------------------------------------
End running model
```

B.2. Case 2: Scale Up to 20 Variables

Input

```
! Data file for `20. Conference Management Assignment - 2'

PRICE: [ 89, 99, 119, 112,143,94,130,98,155,152,55,68,99,140,87,75,78,110,96,74] !Constraint coeffici

RATING: [7.8,8.3,8.0,8.7,8.0,8.1,8.6,8.9,8.9,8.4,9.5,7.6,9.5,7.9,8.3,8.2,9.9,8.8,8.0,8.9] !Constrai

AVAIL:[35,30,15,15,15,20,15,10,10,20,2,5,6,7,9,1,2,5,4,6] !Values of the constraints
```

Output

```
Begin running model
---------------------------------------------------------
Problem status: LP Optimum found
Cost is: âĈň14061
---------------------------------------------------------
Passenger in Hotel_1 -> 35
Passenger in Hotel_2 -> 30
Passenger in Hotel_3 -> 0
```

```
Passenger in Hotel_4 -> 15
Passenger in Hotel_5 -> 0
Passenger in Hotel_6 -> 20
Passenger in Hotel_7 -> 0
Passenger in Hotel_8 -> 10
Passenger in Hotel_9 -> 0
Passenger in Hotel_10 -> 0
Passenger in Hotel_11 -> 2
Passenger in Hotel_12 -> 5
Passenger in Hotel_13 -> 6
Passenger in Hotel_14 -> 0
Passenger in Hotel_15 -> 9
Passenger in Hotel_16 -> 1
Passenger in Hotel_17 -> 2
Passenger in Hotel_18 -> 5
Passenger in Hotel_19 -> 4
Passenger in Hotel_20 -> 6
0.0306667
---------------------------------------------------------
\ Using Xpress-MP extensions
Minimize
 89 decVar(1) + 99 decVar(2) + 119 decVar(3) + 112 decVar(4) + 143 decVar(5) +
94 decVar(6) + 130 decVar(7) + 98 decVar(8) + 155 decVar(9) + 152 decVar(10) +
55 decVar(11) + 68 decVar(12) + 99 decVar(13) + 140 decVar(14) + 87 decVar(15) +
75 decVar(16) + 78 decVar(17) + 110 decVar(18) + 96 decVar(19) + 74 decVar(20)

Subject To
_R1: decVar(1) + decVar(2) + decVar(3) + decVar(4) + decVar(5) + decVar(6) +
decVar(7) + decVar(8) + decVar(9) + decVar(10) + decVar(11) + decVar(12) + decVar(13) +
decVar(14) + decVar(15) + decVar(16) + decVar(17) + decVar(18) + decVar(19) +
decVar(20) = 150
AvgRating: 0.052 decVar(1) + 0.0553333 decVar(2) + 0.0533333 decVar(3) + 0.058 decVar(4) +
0.0533333 decVar(5) + 0.054 decVar(6) + 0.0573333 decVar(7) + 0.0593333 decVar(8) +
0.0593333 decVar(9) + 0.056 decVar(10) + 0.0633333 decVar(11) + 0.0506667 decVar(12) +
0.0633333 decVar(13) + 0.0526667 decVar(14) + 0.0553333 decVar(15) + 0.0546667 decVar(16) +
0.066 decVar(17) + 0.0586667 decVar(18) + 0.0533333 decVar(19) + 0.0593333 decVar(20) >= 8.3

Bounds
decVar(1) <= 35
decVar(2) <= 30
decVar(3) <= 15
decVar(4) <= 15
decVar(5) <= 15
decVar(6) <= 20
decVar(7) <= 15
decVar(8) <= 10
decVar(9) <= 10
decVar(10) <= 20
decVar(11) <= 2
decVar(12) <= 5
decVar(13) <= 6
decVar(14) <= 7
decVar(15) <= 9
```

```
decVar(17) <= 2
decVar(18) <= 5
decVar(19) <= 4
decVar(20) <= 6

Integers
decVar(1) decVar(2) decVar(3) decVar(4) decVar(5) decVar(6) decVar(7) decVar(8)
decVar(9) decVar(10) decVar(11) decVar(12) decVar(13) decVar(14) decVar(15)
decVar(16) decVar(17) decVar(18) decVar(19) decVar(20)

End
---------------------------------------------------------
End running model
```

B.3. Case 2: Scale Up to 20 Variables

Input

```
! Data file for `20. Conference Management Assignment 2'

PRICE: [ 89, 99, 119, 112,143,94,130,98,155,152,55,68,99,140,87,75,78,110,96,74,65,5,40,30,30,17,15,8

RATING: [7.8,8.3,8.0,8.7,8.0,8.1,8.6,8.9,8.9,8.4,9.5,7.6,9.5,7.9,8.3,8.2,9.9,8.8,8.0,8.9,8.1,8.3,8.

AVAIL:[35,30,15,15,15,20,15,10,10,20,2,5,6,7,9,1,2,5,4,6,2,5,4,6,1,2,3,17,2,5] !Values of the constr
```

Output

```
Begin running model
---------------------------------------------------------
Problem status: LP Optimum found
Cost is: âĈň11395
---------------------------------------------------------
Passenger in Hotel_1 -> 35
Passenger in Hotel_2 -> 3
Passenger in Hotel_3 -> 0
Passenger in Hotel_4 -> 0
Passenger in Hotel_5 -> 0
Passenger in Hotel_6 -> 20
Passenger in Hotel_7 -> 0
Passenger in Hotel_8 -> 10
Passenger in Hotel_9 -> 0
Passenger in Hotel_10 -> 0
Passenger in Hotel_11 -> 2
Passenger in Hotel_12 -> 5
Passenger in Hotel_13 -> 6
Passenger in Hotel_14 -> 0
Passenger in Hotel_15 -> 9
Passenger in Hotel_16 -> 1
Passenger in Hotel_17 -> 2
Passenger in Hotel_18 -> 0
Passenger in Hotel_19 -> 4
Passenger in Hotel_20 -> 6
Passenger in Hotel_21 -> 2
Passenger in Hotel_22 -> 5
```

```
Passenger in Hotel_23 -> 4
Passenger in Hotel_24 -> 6
Passenger in Hotel_25 -> 1
Passenger in Hotel_26 -> 2
Passenger in Hotel_27 -> 3
Passenger in Hotel_28 -> 17
Passenger in Hotel_29 -> 2
Passenger in Hotel_30 -> 5
0.0713333
---------------------------------------------------------
\ Using Xpress-MP extensions
Minimize
 89 decVar(1) + 99 decVar(2) + 119 decVar(3) + 112 decVar(4) + 143 decVar(5) +
94 decVar(6) + 130 decVar(7) + 98 decVar(8) + 155 decVar(9) + 152 decVar(10) +
55 decVar(11) + 68 decVar(12) + 99 decVar(13) + 140 decVar(14) + 87 decVar(15) +
75 decVar(16) + 78 decVar(17) + 110 decVar(18) + 96 decVar(19) + 74 decVar(20) +
65 decVar(21) + 5 decVar(22) + 40 decVar(23) + 30 decVar(24) + 30 decVar(25) +
17 decVar(26) + 15 decVar(27) + 89 decVar(28) + 10 decVar(29) + 20 decVar(30)

Subject To
_R1: decVar(1) + decVar(2) + decVar(3) + decVar(4) + decVar(5) + decVar(6) +
decVar(7) + decVar(8) + decVar(9) + decVar(10) + decVar(11) + decVar(12) + decVar(13) +
decVar(14) + decVar(15) + decVar(16) + decVar(17) + decVar(18) + decVar(19) +
decVar(20) + decVar(21) + decVar(22) + decVar(23) + decVar(24) + decVar(25) +
decVar(26) + decVar(27) + decVar(28) + decVar(29) + decVar(30) = 150
AvgRating: 0.052 decVar(1) + 0.0553333 decVar(2) + 0.0533333 decVar(3) + 0.058 decVar(4) +
0.0533333 decVar(5) + 0.054 decVar(6) + 0.0573333 decVar(7) + 0.0593333 decVar(8) +
0.0593333 decVar(9) + 0.056 decVar(10) + 0.0633333 decVar(11) + 0.0506667 decVar(12) +
0.0633333 decVar(13) + 0.0526667 decVar(14) + 0.0553333 decVar(15) + 0.0546667 decVar(16) +
0.066 decVar(17) + 0.0586667 decVar(18) + 0.0533333 decVar(19) + 0.0593333 decVar(20) +
0.054 decVar(21) + 0.0553333 decVar(22) + 0.0546667 decVar(23) + 0.0566667 decVar(24) +
0.0593333 decVar(25) + 0.0633333 decVar(26) + 0.052 decVar(27) + 0.0606667 decVar(28) +
0.054 decVar(29) + 0.0546667 decVar(30) >= 8.3

Bounds
decVar(1) <= 35
decVar(2) <= 30
decVar(3) <= 15
decVar(4) <= 15
decVar(5) <= 15
decVar(6) <= 20
decVar(7) <= 15
decVar(8) <= 10
decVar(9) <= 10
decVar(10) <= 20
decVar(11) <= 2
decVar(12) <= 5
decVar(13) <= 6
decVar(14) <= 7
decVar(15) <= 9
decVar(17) <= 2
decVar(18) <= 5
decVar(19) <= 4
```

```
decVar(20) <= 6
decVar(21) <= 2
decVar(22) <= 5
decVar(23) <= 4
decVar(24) <= 6
decVar(26) <= 2
decVar(27) <= 3
decVar(28) <= 17
decVar(29) <= 2
decVar(30) <= 5

Integers
decVar(1) decVar(2) decVar(3) decVar(4) decVar(5) decVar(6) decVar(7) decVar(8)
decVar(9) decVar(10) decVar(11) decVar(12) decVar(13) decVar(14) decVar(15)
decVar(16) decVar(17) decVar(18) decVar(19) decVar(20) decVar(21) decVar(22)
decVar(23) decVar(24) decVar(25) decVar(26) decVar(27) decVar(28) decVar(29)
decVar(30)

End
----------------------------------------------------------
End running model
```