Integer/Binary Integer Programming Presentation

Group 4
Erin Romfo
Anthony Rende
Joe Redmond

Integer Linear Programs

- In an All-Integer Linear Program all the variables are integers.
- In LP Relaxation the integer requirements are removed from the program
- In a Mixed-Integer Linear Program some variables, but not all, are integers.
- In a Binary Integer Linear Program the variables are restricted to a value of 0 or 1.

Some Applications of Integer Linear Programming:

- Capital budgeting – capital is limited and management would like to select the most profitable projects.
- Fixed cost – there is a fixed cost associated with production setup and a maximum production quantity for the products.
- Distribution system design – determine the best plant locations and to determine how much to ship from the plants to distribution centers.

- <u>Location problem </u>– minimum amount of locations to do business and serve the largest area.
- <u>Product design & market share </u>– use the preferences of prospective consumers/buyers to determine what to produce.

All-Integer Problem

To help illustrate this problem, let's use our favorite example of tables and chairs. T&C Company wants to maximize their profits. They make $10 for every table and $3 for every chair. Employee #1 can make 6 tables and 7 chairs, but can't work more than 40 hours. Employee #2 can make 3 tables and 1 chair, but can't work more than 11 hours.

LP Relaxation

Model:

Max $10x_1 + 3x_2$

s.t. $6x_1 + 7x_2 \leq 40$

$3x_1 + x_2 \leq 11$

$x_1, \quad x_2 \geq 0$

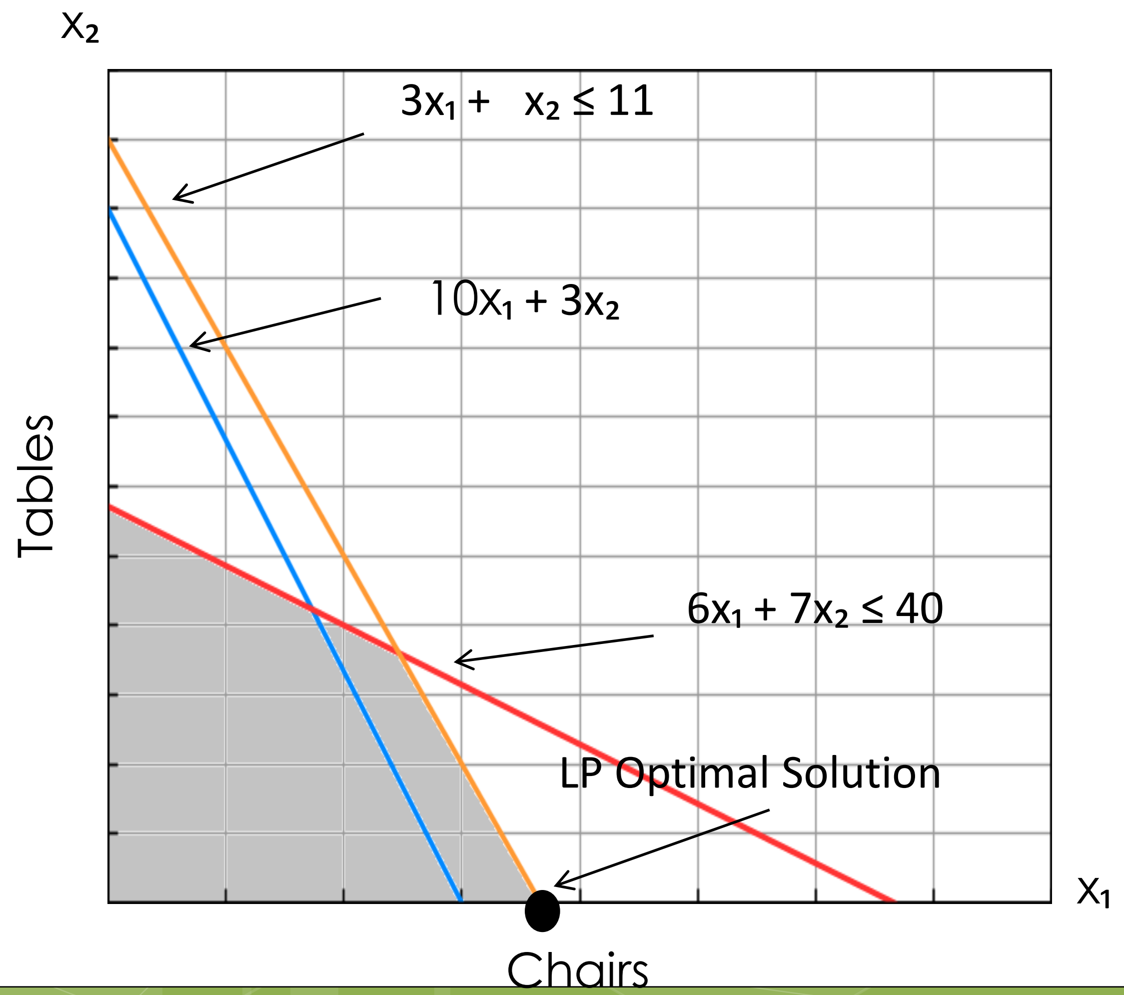
Optimal Solution:

OF = 36.66667

$x_1 = 3.666667$

$x_2 = 0$

Graph of LP Relaxation Problem



Rounding Up and Rounding Down

- In this situation rounding x_1 up from 3.666667 to 4 would give a solution outside the feasible region.
- Rounding down x_1 from 3.666667 to 3 would provide a feasible solution, but not necessarily the optimal solution.

Complete Enumeration of Feasible Solutions

	x_1	x_2	$10x_1 + 3x_2$
1.	0	0	0
2.	1	0	10
3.	2	0	20
4.	3	0	30
5.	0	1	3
6.	1	1	13
7.	2	1	23
8.	3	1	33
9.	0	2	6
10.	1	2	16

	x_1	x_2	$10x_1 + 3x_2$
11.	2	2	26
12.	**3**	**2**	**36**
13.	0	3	9
14.	1	3	19
15.	2	3	29
16.	0	4	12
17.	1	4	22
18.	2	4	32
19.	0	5	15

Calculating the Optimal Solution

So, if we take the original model and add the integer constraint we can find the optimal solution much quicker.

$$\text{Max} \quad 10x_1 + 3x_2$$

$$\text{s.t.} \quad 6x_1 + 7x_2 \leq 40$$

$$3x_1 + x_2 \leq 11$$

$$x_1, \quad x_2 \geq 0 \text{ and integer}$$

Input into LINGO

```
Model:
!Objective Function;
Max = 10*x1 + 3*x2;
!Subject to;
6*x1 + 7*x2 <= 40;
3*x1 + x2 <= 11;
@Gin (x1);
@Gin (x2);
End
```

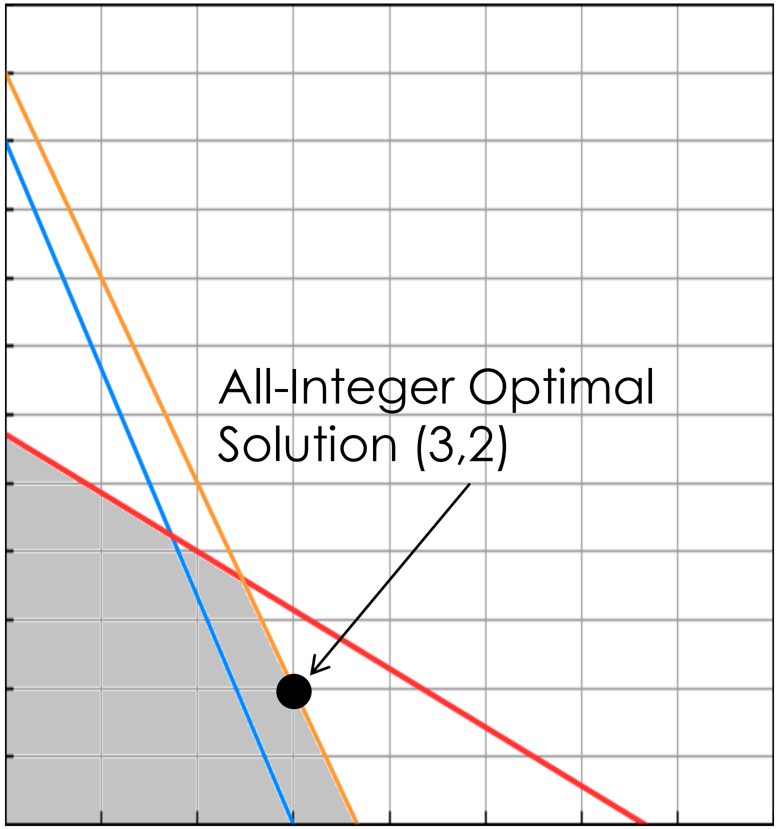
LINGO Results and Graph

Global optimal solution found.

Objective value:	36.00000
Objective bound:	36.00000
Infeasibilities:	0.000000
Extended solver steps:	0
Total solver iterations:	0
Elapsed runtime seconds:	0.05
Model Class:	PILP
Total variables:	2
Nonlinear variables:	0
Integer variables:	2
Total constraints:	3
Nonlinear constraints:	0
Total nonzeros:	6
Nonlinear nonzeros:	0

Variable	Value	Reduced Cost
X1	3.000000	-10.00000
X2	2.000000	-3.000000

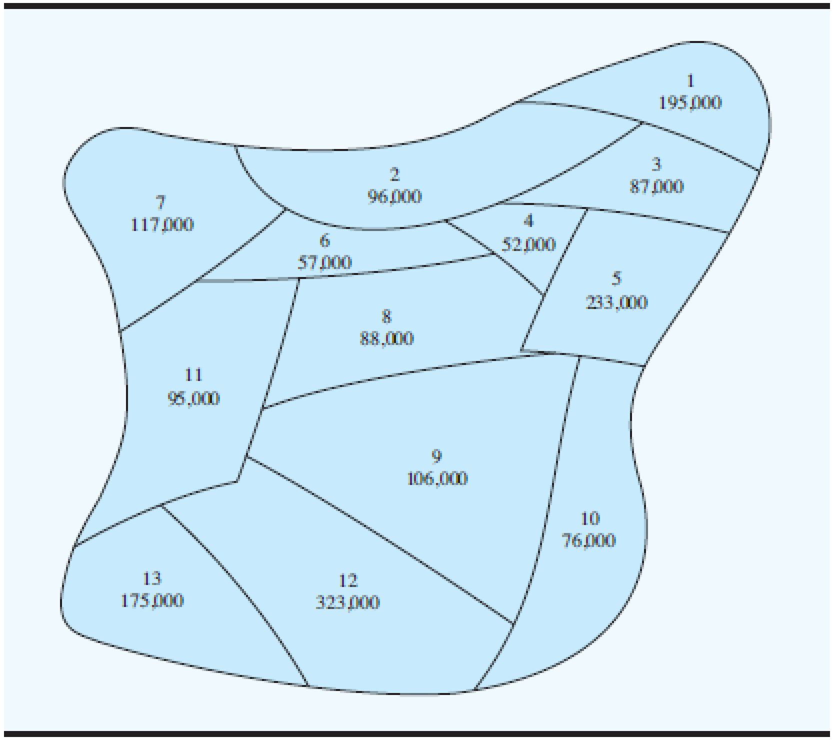
Row	Slack or Surplus	Dual Price
1	36.00000	1.000000
2	8.000000	0.000000
3	0.000000	0.000000



Binary Integer Programming Problem

CHB Inc., is a bank holding company that is evaluating the potential for expanding into a 13-county region in the southwestern part of the state. State law permits establishing branches in any county that is adjacent to a county in which a PPB (principal place of business) is located. The following map shows the 13-county region with the population of each county indicated.

Map



1
195,000
2
96,000
3
87,000
4
52,000
5
233,000
6
57,000
7
117,000
8
88,000
9
106,000
10
76,000
11
95,000
12
323,000
13
175,000

Table of Counties

Counties Under Consideration	Adjacent Counties
1	2,3
2	1,3,4,6,7
3	1,2,4,5
4	2,3,5,6,8
5	3,4,8,9,10
6	2,4,7,8,11
7	2,6,11
8	4,5,6,9,11
9	5,8,10,11,12
10	5,9,12
11	6,7,8,9,12,13
12	9,10,11,13
13	11,12

Decision Variables and Problem Formulation

x_i = County, 1 if established and 0 if not.

Min $x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8 + x_9 + x_{10} + x_{11} + x_{12} + x_{13}$

s.t.
$x_1 + x_2 + x_3 \geq 1$
$x_1 + x_2 + x_3 + x_4 + x_6 + x_7 \geq 1$
$x_1 + x_2 + x_3 + x_4 + x_5 \geq 1$
$x_2 + x_3 + x_4 + x_5 + x_6 + x_8 \geq 1$
$x_3 + x_4 + x_5 + x_8 + x_9 + x_{10} \geq 1$
$x_2 + x_4 + x_6 + x_7 + x_8 + x_{11} \geq 1$
$x_2 + x_6 + x_7 + x_{11} \geq 1$
$x_4 + x_5 + x_6 + x_8 + x_9 + x_{11} \geq 1$
$x_5 + x_8 + x_9 + x_{10} + x_{11} + x_{12} \geq 1$
$x_5 + x_9 + x_{10} + x_{12} \geq 1$
$x_6 + x_7 + x_8 + x_9 + x_{11} + x_{12} + x_{13} \geq 1$
$x_9 + x_{10} + x_{11} + x_{12} + x_{13} \geq 1$
$x_{11} + x_{12} + x_{13} \geq 1$
$x_i = 0,1$

LINGO Model

```
Objective Function;
Min = x1 + x2 + x3 + x4  + x5 + x6 + x7 + x8 + x9 + x10 + x11 + x12 + x13;
!Subject to;
x1 + x2 + x3 >=1;
x1 + x2 +x3 + x4 + x6 + x7 >= 1;
x1 + x2 + x3 + x4 + x5 >= 1;
x2 + x3 + x4 + x5 + x6 + x8 >= 1;
x3 + x4 + x5 + x8 + x9 + x10 >= 1;
x2 + x4 + x6 + x7 + x8 + x11 >= 1;
x2 + x6 + x7 + x11 >= 1;
x4 + x5 + x6 + x8 + x9 + x11 >= 1;
x5 + x8 + x9 + x10 + x11 + x12 >= 1;
x5 + x9 + x10 + x12 >= 1;
x6 + x7 + x8 + x9 + x11 + x12 + x13 >= 1;
x9 + x10 + x11 + x12 + x13 >= 1;
x11 + x12 + x13 >= 1;
@Bin (x1);
@Bin (x2);
@Bin (x3);
@Bin (x4);
@Bin (x5);
@Bin (x6);
@Bin (x7);
@Bin (x8);
@Bin (x9);
@Bin (x10);
@Bin (x11);
@Bin (x12);
@Bin (x13);
End
```

LINGO Results

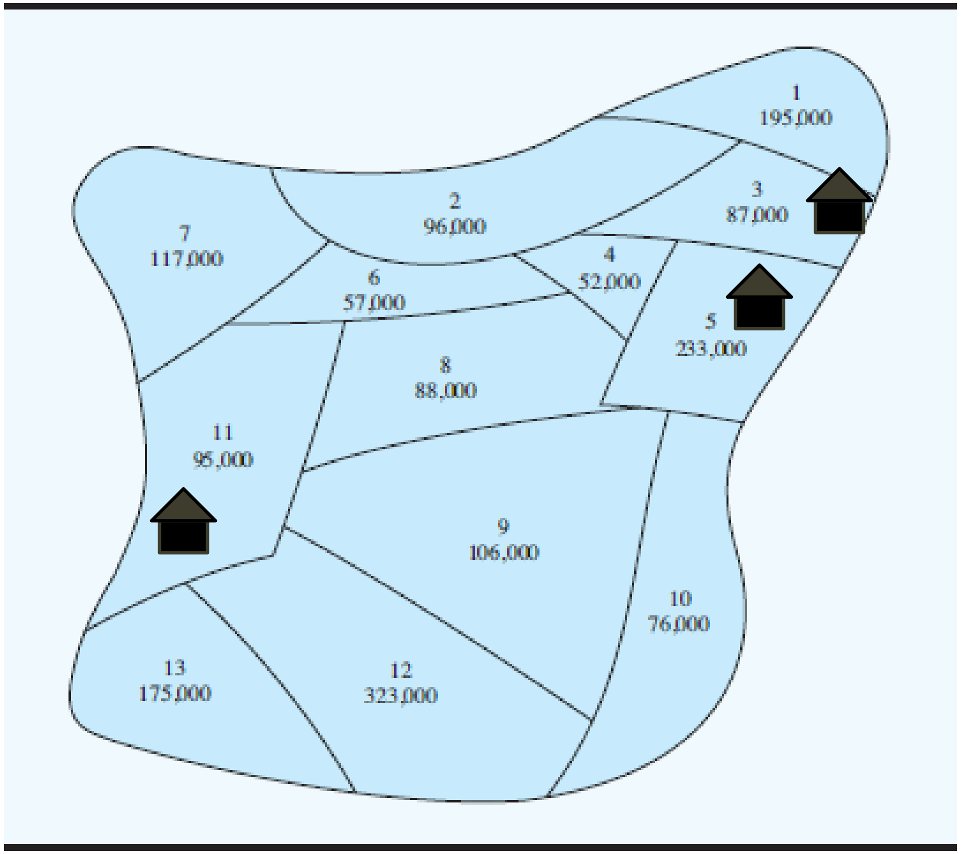
Global optimal solution found.

Objective value:	3.000000
Objective bound:	3.000000
Infeasibilities:	0.000000
Extended solver steps:	0
Total solver iterations:	0
Elapsed runtime seconds:	0.05
Model Class:	PILP
Total variables:	13
Nonlinear variables:	0
Integer variables:	13
Total constraints:	14
Nonlinear constraints:	0
Total nonzeros:	80
Nonlinear nonzeros:	0

Variable	Value	Reduced Cost
X1	0.000000	1.000000
X2	0.000000	1.000000
X3	1.000000	1.000000
X4	0.000000	1.000000
X5	1.000000	1.000000
X6	0.000000	1.000000
X7	0.000000	1.000000
X8	0.000000	1.000000
X9	0.000000	1.000000
X10	0.000000	1.000000
X11	1.000000	1.000000
X12	0.000000	1.000000
X13	0.000000	1.000000

Row	Slack or Surplus	Dual Price
1	3.000000	-1.000000
2	0.000000	0.000000
3	0.000000	0.000000
4	1.000000	0.000000
5	1.000000	0.000000
6	1.000000	0.000000
7	0.000000	0.000000
8	0.000000	0.000000
9	1.000000	0.000000
10	1.000000	0.000000
11	0.000000	0.000000
12	0.000000	0.000000
13	0.000000	0.000000
14	0.000000	0.000000

Map of Branches to be Built



What if only one branch could be built?

Min $195{,}000y_1 + 96{,}000y_2 + 87{,}000y_3 + 52{,}000y_4 + 233{,}000y_5 + 57{,}000y_6 + 117{,}000y_7 + 88{,}000y_8 + 106{,}000\,y_9 + 76{,}000y_{10} + 95{,}000y_{11} + 323{,}000y_{12} + 175{,}000y_{13}$

s.t.

$x_1 + x_2 + x_3 \geq 1 - y_1$

$x_1 + x_2 + x_3 + x_4 + x_6 + x_7 \geq 1 - y_2$

$x_1 + x_2 + x_3 + x_4 + x_5 \geq 1 - y_3$

$x_2 + x_3 + x_4 + x_5 + x_6 + x_8 \geq 1 - y_4$

$x_3 + x_4 + x_5 + x_8 + x_9 + x_{10} \geq 1 - y_5$

$x_2 + x_4 + x_6 + x_7 + x_8 + x_{11} \geq 1 - y_6$

$x_2 + x_6 + x_7 + x_{11} \geq 1 - y_7$

$x_4 + x_5 + x_6 + x_8 + x_9 + x_{11} \geq 1 - y_8$

$x_5 + x_8 + x_9 + x_{10} + x_{11} + x_{12} \geq 1 - y_9$

$x_5 + x_9 + x_{10} + x_{12} \geq 1 - y_{10}$

$x_6 + x_7 + x_8 + x_9 + x_{11} + x_{12} + x_{13} \geq 1 - y_{11}$

$x_9 + x_{10} + x_{11} + x_{12} + x_{13} \geq 1 - y_{12}$

$x_{11} + x_{12} + x_{13} \geq 1 - y_{13}$

$x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8 + x_9 + x_{10} + x_{11} + x_{12} + x_{13} = 1$

x_i and $y_i = 0,1$

LINGO Results

Global optimal solution found.
Objective value: 739000.0
Objective bound: 739000.0
Infeasibilities: 0.000000
Extended solver steps: 0
Total solver iterations: 13
Elapsed runtime seconds: 0.06

Model Class: PILP

Total variables: 26
Nonlinear variables: 0
Integer variables: 26

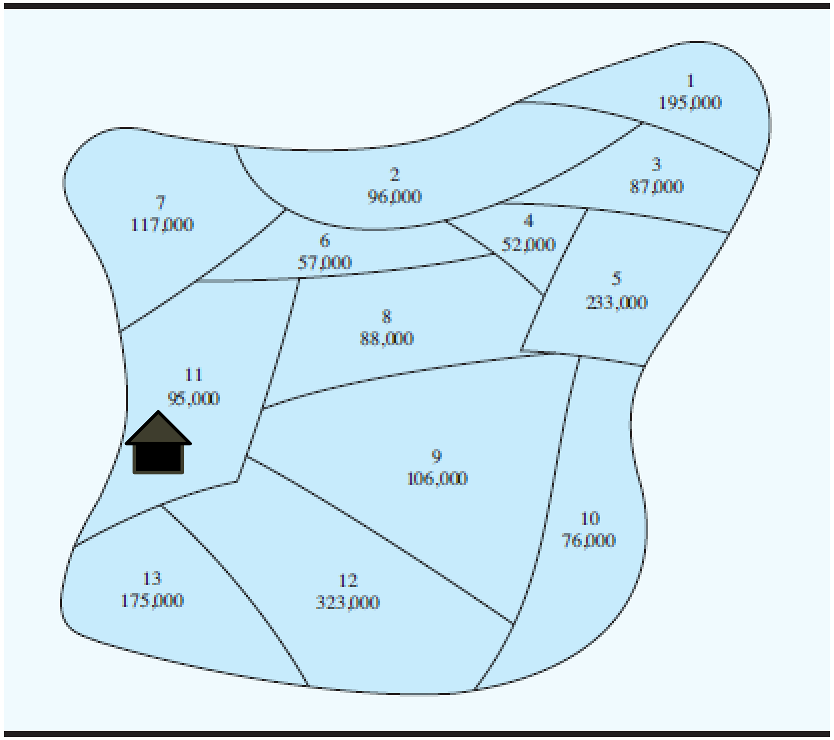
Total constraints: 15
Nonlinear constraints: 0

Total nonzeros: 106
Nonlinear nonzeros: 0

Variable	Value	Reduced Cost
Y1	1.000000	195000.0
Y2	1.000000	96000.00
Y3	1.000000	87000.00
Y4	1.000000	52000.00
Y5	1.000000	233000.0
Y6	0.000000	57000.00
Y7	0.000000	117000.0
Y8	0.000000	88000.00
Y9	0.000000	106000.0
Y10	1.000000	76000.00
Y11	0.000000	95000.00
Y12	0.000000	323000.0
Y13	0.000000	175000.0
X1	0.000000	0.000000
X2	0.000000	0.000000
X3	0.000000	0.000000
X4	0.000000	0.000000
X6	0.000000	0.000000
X7	0.000000	0.000000
X5	0.000000	0.000000
X8	0.000000	0.000000
X9	0.000000	0.000000
X10	0.000000	0.000000
X11	1.000000	0.000000
X12	0.000000	0.000000
X13	0.000000	0.000000

Row	Slack or Surplus	Dual Price
1	739000.0	-1.000000
2	0.000000	0.000000
3	0.000000	0.000000
4	0.000000	0.000000
5	0.000000	0.000000
6	0.000000	0.000000
7	0.000000	0.000000
8	0.000000	0.000000
9	0.000000	0.000000
10	0.000000	0.000000
11	0.000000	0.000000
12	0.000000	0.000000
13	0.000000	0.000000
14	0.000000	0.000000
15	0.000000	0.000000

Total Population Served



Conclusion

The problems that have been shown only represent a couple of ways that Integer and Binary Integer Programming can be used in real world applications. There are so many ways to use this programming it would be impossible to illustrate them all!

The End