SOLVING ILP USING GLPK

Specifying integer variables in Mathprog

GLPK integer solver

- GLPK has a very good integer solver.
  - Uses branch-and-bound + Gomory cut techniques
  - We will examine these techniques soon.

- In this lecture,
  - Show how to solve (mixed) integer linear programs
  - Continue to use AMPL format.

- This is the best option for solving ILPs/MIPs

Example-1 (ILP)

$$
\begin{array}{lcccccccc}
\min & x_1 & +x_2 & +x_3 & +x_4 & +x_5 & +x_6 & & \\
 & x_1 & +x_2 & & & & & \geq & 1 \\
 & x_1 & +x_2 & & & & +x_6 & \geq & 1 \\
 & & & x_3 & +x_4 & & & \geq & 1 \\
 & & & x_3 & +x_4 & +x_5 & & \geq & 1 \\
 & & & & x_4 & +x_5 & +x_6 & \geq & 1 \\
 & & x_2 & & & +x_5 & +x_6 & \geq & 1 \\
 & x_1, & x_2, & x_3, & x_4, & x_5, & x_6 & \in & \mathbb{Z}
\end{array}
$$

Specifying variable type

```
var x; # specifies a real-valued decision variable
var y integer; # specifies an integer variable
var z binary; # specifies a binary variable
```

Example – I expressing in AMPL

```
var x{1..6} integer;
# Declare 6 integer variables
minimize obj: sum{i in 1..6} x[i];
c1: x[1] + x[2] >= 1;
c2: x[1] + x[2] + x[6] >= 1;
c4: x[3] + x[4]  >= 1;
c5: x[3]+ x[4] + x[5] >= 1;
c6: x[4] + x[5] + x[6]   >= 1;
c7: x[2] + x[5] + x[6] >= 1;
solve;
display{i in 1..6} x[i];
end
```

$$
\begin{array}{lcccccccc}
\min & x_1 & +x_2 & +x_3 & +x_4 & +x_5 & +x_6 & & \\
 & x_1 & +x_2 & & & & & \geq & 1 \\
 & x_1 & +x_2 & & & & +x_6 & \geq & 1 \\
 & & & x_3 & +x_4 & & & \geq & 1 \\
 & & & x_3 & +x_4 & +x_5 & & \geq & 1 \\
 & & & & x_4 & +x_5 & +x_6 & \geq & 1 \\
 & & x_2 & & & +x_5 & +x_6 & \geq & 1 \\
 & x_1, & x_2, & x_3, & x_4, & x_5, & x_6 & \in & \mathbb{Z}
\end{array}
$$

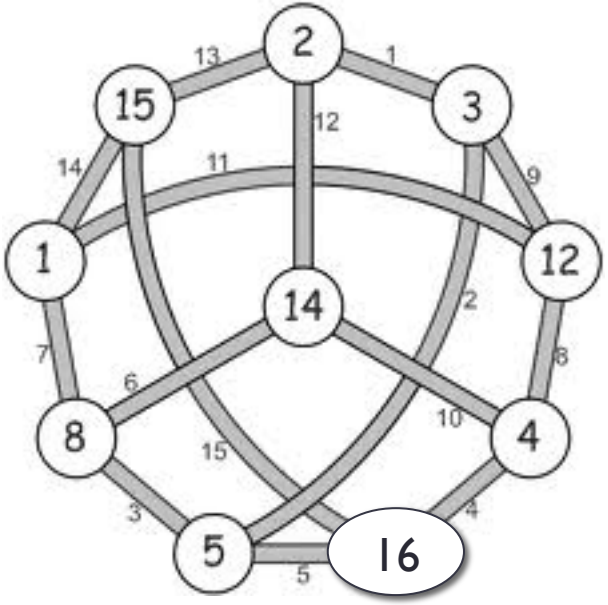
Example-1 Solving using GLPK

- glpsol -- math ip1.math

Display statement at line 25
x[1].val = 0
x[2].val = 1
x[3].val = 0
x[4].val = 1
x[5].val = 0
x[6].val = 0
Model has been successfully processed

Example -2

Vertex Cover Problem



source mathpuzzle.com

Vertex Cover to ILP

- Vertices {1,…, n}
  - Decision variables: $x_1, \ldots, x_n$

$$x_i \in \{0, 1\}$$

$$\begin{aligned} \min \quad & \sum_{i=1}^{n} x_i & \\ \text{s.t.} \quad & 0 \leq x_i \leq 1 & \forall\ i \in V \\ & x_i + x_j \geq 1 & \forall\ (i,j) \in E \\ & x_i \in \mathbb{Z} & \forall\ i \in V \end{aligned}$$

Vertex Cover AMPL (Model + Data)

```
param n;
var x {1..n} binary;
# binary specifies that the variables are binary

set E within {i in 1..n, j in 1..n: i < j};
# specify that the edges will be a set.
# each edge will be entered as (i,j) where i < j

minimize obj: sum{i in 1..n} x[i];
# minimize cost of the cover
s.t.
c{(i,j) in E}: x[i] + x[j] >= 1;

solve;
display{i in 1..n} x[i];
```

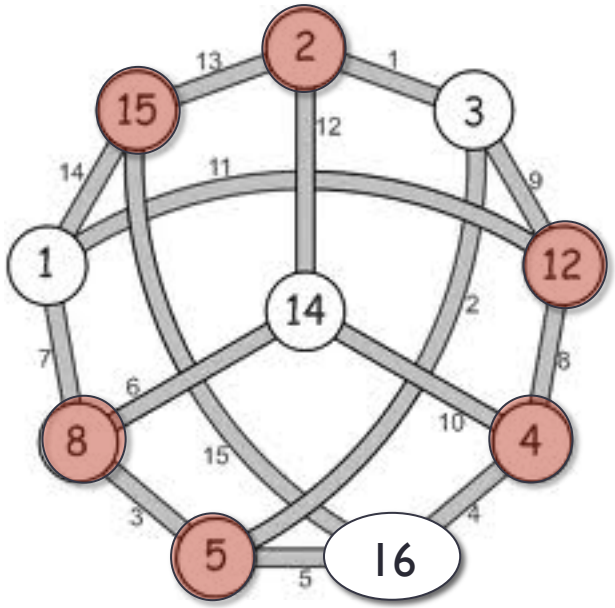
```
data;
param n := 16;

set E :=  (2,3) (3,5) (5,8)
          (4,16) (5,16) (8,14)
        (1,8) (4,12) (3,12) (4,14)
      (1,12) (2,14) (2,15) (1,15) (15,16) ;



end;
```

Running GLPK …



- glpsol -m vertexCover.model

```
x[1].val = 0
x[2].val = 1
x[3].val = 0
x[4].val = 1
x[5].val = 1
x[6].val = 0
x[7].val = 0
x[8].val = 1
x[9].val = 0
x[10].val = 0
x[11].val = 0
x[12].val = 1
x[13].val = 0
x[14].val = 0
x[15].val = 1
x[16].val = 0
```