ISE 426
Optimization models and applications

Lecture 15 — October 22, 2015

- Integer (Linear) Programming (IP)
- Examples

Reading:

- Winston & Venkataramanan, Chapter 9

Mixed-Integer Linear Programming (MILP) problems

…or more simply, Integer Programming (IP) problems:

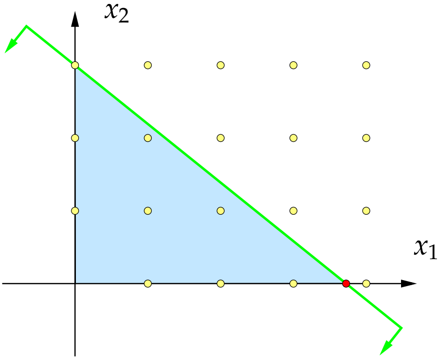
$$
\begin{array}{llllll}
\min & c_1x_1 & +c_2x_2 & \dots & +c_nx_n & & \\
 & a_{11}x_1 & +a_{12}x_2 & \dots & +a_{1n}x_n & \leq b_1 & \\
 & a_{21}x_1 & +a_{22}x_2 & \dots & +a_{2n}x_n & \leq b_2 & \\
 & \vdots & & & & & \\
 & a_{m1}x_1 & +a_{m2}x_2 & \dots & +a_{mn}x_n & \leq b_m & \\
 & & & & x_i & \in \mathbb{Z} & \forall i \in J \subseteq \{1, 2 \dots, n\}
\end{array}
$$

A much more powerful modeling tool than LP:

yes/no decisions ($x_i \in \{0, 1\}$)	nonconvexities
integer quantities	discontinuities
piece-wise linear functions	economies of scale

Much more difficult than LP models: **nonconvex**

Why can't we just round numbers up/down?



$$\begin{array}{lll} \max & x_1 & +1.1x_2 \\ & \frac{x_1}{3.7} & +\frac{x_2}{3} \leq 1 \\ & & x_1, x_2 \in \mathbb{Z}_+ \end{array}$$

- Optimal solution of the LP relaxation: $(3.7, 0)$, obj. f.: 3.7
- Rounded solution: $(3, 0)$, obj. f.:3.0
- Optimal solution of the original problem: $(0, 3)$, obj. f.: 3.3

LP and IP

IP is a bit younger than LP, but is the subject of extensive research and can model countless problems in Industry:

- Airline crew scheduling
- Vehicle Routing
- Financial applications
- Design of Telecommunications networks

IP problems are difficult to solve:

- Require very specialized techniques
  (some use LP relaxations to find lower bounds)
- A good model makes a problem easier (but not easy)

i.e. unlike LP, the way we model an Optimization problem **affects** the chances to solve it

Binary variables, logical operators

- model yes/no decisions: $x_i \in \{0, 1\}$
- $x_i = 0$ if the decision is "no",
- $x_i = 1$ if it is "yes"
- can use logical operators: implications, disjunctions, etc.:
  - Mario or Luigi will have ice cream, but not both:
    $x_{\text{Mario}} + x_{\text{Luigi}} \leq 1$
  - At least one among Mario and Luigi will have ice cream:
    $x_{\text{Mario}} + x_{\text{Luigi}} \geq 1$
  - If Mario has ice cream, then Giovanni will have one too:
    $x_{\text{Mario}} \leq x_{\text{Giovanni}}$
  - Luigi gets ice cream if and only if Paolo does not get any:
    $x_{\text{Luigi}} = 1 - x_{\text{Paolo}}$

Binary variables and operations with sets

Binary variables are useful to model problems on sets. E.g.:

- Choose a subset S of a set A of elements such that S has certain properties (e.g. not more than K elements, etc.)
- Each element $i \in A$ has a cost c_a

$\Rightarrow$ The cost of a solution S is $\sum_{i \in S} c_i$

- Define variable x_i:

$$x_i = \begin{cases} 1 & \text{if } i \in S \\ 0 & \text{otherwise} \end{cases}$$

- now the cost of a solution S is $\displaystyle\sum_{i \in A : x_i = 1} c_i = \sum_{i \in A} c_i x_i$
- define properties similarly, e.g. $|S| \leq K$ is $\displaystyle\sum_{i \in A} x_i \leq K$

Example: Subset Sum

Two brothers, Ludwig and Johann, inherit from their dear uncle a set A of antique objects.

- each worth a lot of money, c_i for all $i \in A$
- they want to share these objects in a balanced manner

$\Rightarrow$ minimize the difference between their total values

- $S_L \subset A$ contains the objects that Ludwig will get, while $S_J = A \setminus S_L$ are the remaining objects

$\Rightarrow$ minimize

$$\left| \sum_{i \in S_L} c_i - \sum_{i \in S_J} c_i \right|$$

How to model this with IP?

Example: Subset Sum (cont'd)

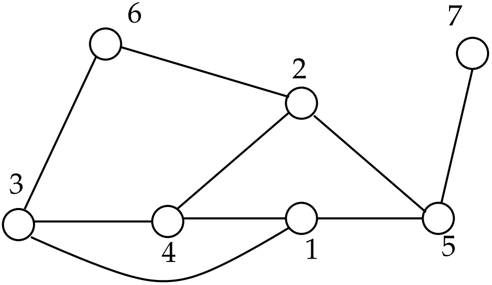
$$x_i = \begin{cases} 1 & \text{if Ludwig gets the } i\text{-th object} \\ 0 & \text{if Johann gets the } i\text{-th object} \end{cases}$$

Then the integer model is

$$\begin{aligned} \min \quad & \left| \sum_{i \in A} c_i x_i - \sum_{i \in A} c_i (1 - x_i) \right| \\ & x_i \in \{0, 1\} \qquad \forall i \in A \end{aligned}$$

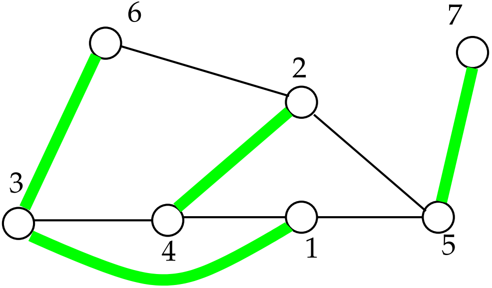
Example: the edge covering problem

In a graph $G = (V, E)$ as in the figure, choose a subset S of edges such that all nodes are "covered" by at least one edge in S. Minimize the number of edges used



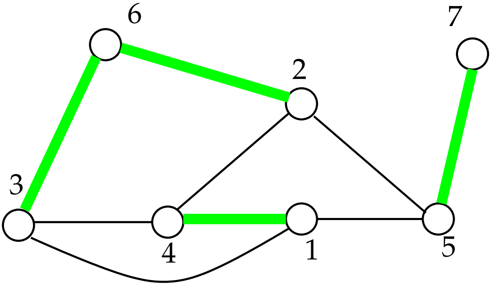
Example: the edge covering problem

In a graph $G = (V, E)$ as in the figure, choose a subset S of edges such that all nodes are "covered" by at least one edge in S. Minimize the number of edges used



Example: the edge covering problem

In a graph $G = (V, E)$ as in the figure, choose a subset S of edges such that all nodes are "covered" by at least one edge in S. Minimize the number of edges used



Edge covering

$$
\begin{array}{lllllllllll}
\min & x_{13} & +x_{14} & +x_{15} & +x_{24} & +x_{25} & +x_{26} & +x_{34} & +x_{36} & +x_{57} & \\
& x_{13} & +x_{14} & +x_{15} & & & & & & & \geq 1 \\
& & & & x_{24} & +x_{25} & +x_{26} & & & & \geq 1 \\
& x_{13} & & & & & & +x_{34} & +x_{36} & & \geq 1 \\
& & x_{14} & & +x_{24} & & & +x_{34} & & & \geq 1 \\
& & & x_{15} & & +x_{25} & & & & +x_{57} & \geq 1 \\
& & & & & & +x_{26} & & +x_{36} & & \geq 1 \\
& & & & & & & & & x_{57} & \geq 1 \\
& x_{13}, & x_{14}, & x_{15}, & x_{24}, & x_{25}, & x_{26}, & x_{34}, & x_{36}, & x_{57} & \in \{0,1\}
\end{array}
$$

Edge covering

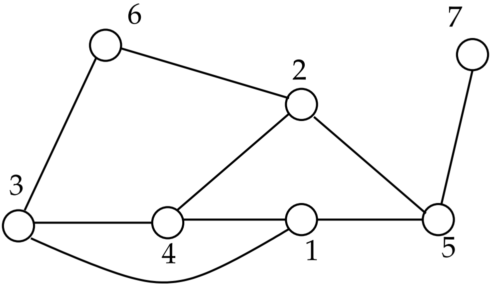
$$
\begin{array}{lll}
\min & \sum_{\{i,j\} \in E} x_{\{i,j\}} & \\
 & \sum_{j \in V : \{i,j\} \in E} x_{\{i,j\}} \geq 1 & \forall i \in V \\
 & x_{\{i,j\}} \in \{0,1\} & \forall \{i,j\} \in E
\end{array}
$$

Graphs and AMPL

```
param n; # number of nodes
set   V=1..n; # set of nodes
set   E within {i in V, j in V: i<j};
  # subset of set of node pairs

var x {E} binary;

minimize numEdges: sum {(i,j) in E} x [i,j];

covering {i in V}:
  sum {j in V: (i,j) in E} x[i,j] +
  sum {j in V: (j,i) in E} x[j,i] >= 1;

data;

param n := 7;
set E := (1,3) (1,4) (1,5) (2,4) (2,5)
         (2,6) (3,4) (3,6) (5,7);
```

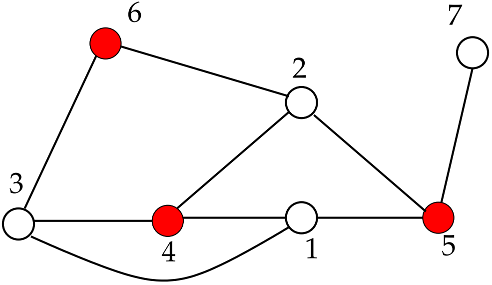
Example: the node packing (or stable set) problem

In a graph $G = (V, E)$ as in the figure, choose a subset S of nodes such that no two nodes i and j in S are adjacent, i.e. share an edge $\{i, j\}$. In other words, if both i and j are included in S, then there must be no edge $\{i, j\}$. Maximize the number of nodes used.



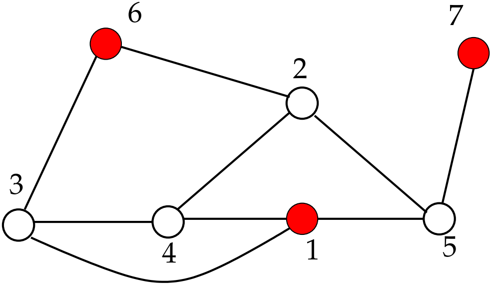
Example: the node packing (or stable set) problem

In a graph $G = (V, E)$ as in the figure, choose a subset S of nodes such that no two nodes i and j in S are adjacent, i.e. share an edge $\{i, j\}$. In other words, if both i and j are included in S, then there must be no edge $\{i, j\}$. Maximize the number of nodes used.



Example: the node packing (or stable set) problem

In a graph $G = (V, E)$ as in the figure, choose a subset S of nodes such that no two nodes i and j in S are adjacent, i.e. share an edge $\{i, j\}$. In other words, if both i and j are included in S, then there must be no edge $\{i, j\}$. Maximize the number of nodes used.



Node packing

$$\begin{array}{lccccccccc}
\max & x_1 & +x_2 & +x_3 & +x_4 & +x_5 & +x_6 & +x_7 & \\
 & x_1 & & +x_3 & & & & & \leq 1 \\
 & x_1 & & & +x_4 & & & & \leq 1 \\
 & x_1 & & & & +x_5 & & & \leq 1 \\
 & & x_2 & & +x_4 & & & & \leq 1 \\
 & & x_2 & & & +x_5 & & & \leq 1 \\
 & & x_2 & & & & +x_6 & & \leq 1 \\
 & & & x_3 & +x_4 & & & & \leq 1 \\
 & & & x_3 & & & +x_6 & & \leq 1 \\
 & & & & & x_5 & & +x_7 & \leq 1
\end{array}$$

Node packing

$$\begin{array}{lll} \max & \sum_{i \in V} x_i & \\ & x_i + x_j \leq 1 & \forall \{i,j\} \in E \\ & x_i \in \{0,1\} & \forall i \in V \end{array}$$

Node packing in AMPL

```
param n; # number of nodes
set   V=1..n; # set of nodes
set   E within {i in V, j in V: i<j};
  # subset of set of node pairs

var x {V} binary;

maximize numNodes: sum {i in V} x [i];

packing {(i,j) in E}: x [i] + x [j] <= 1;

data;

param n := 7;
set E := (1,3) (1,4) (1,5) (2,4) (2,5)
         (2,6) (3,4) (3,6) (5,7);
```

Binary variables and *fixed charge* quantities

Binary variables are also useful to model economies of scale:

- fixed transaction costs in financial optimization
- fixed costs for using a facility/plant/service

Example: "Every truck in the fleet costs \$25,000/year for maintenance plus \$1.26 per mile. Determine the trucks to be used and the distance they will travel."

$\Rightarrow$ Use a binary variable $x_i = \begin{cases} 1 & \text{if truck } i \text{ is used} \\ 0 & \text{otherwise} \end{cases}$

… and a continuous variable y_i = miles traveled by truck i

The objective function will look like $\$25,000x_i + \$1.26y_i$.
(Implicit) constraint: if truck i travels half a mile, it has to be bought: $y_i > 0 \Rightarrow x_i = 1$, or

$$y_i \le Mx_i$$

where M is an adequate constant.

Example: Production planning with fixed costs

A small firm produces plastic for the car industry.

- At the beginning of the year, it knows exactly the demand d_i of plastic for every month i.
- It also has a maximum production capacity of P

  and an inventory capacity of C.
- The inventory is empty on 01/01

  and has to be empty again on 12/31
- production has a monthly per-unit cost c_i

\+ a monthly **fixed** cost f_i if the machinery for producing plastic is started in month i

What do we produce at each month to minimize total production cost while satisfying demand?

Production planning with fixed costs

- How much to produce each month i: $x_i,\ i = 1, 2 \ldots, 12$
- The inventory level at the end of each month (demand is satisfied by part of production **and** part of inventory): $y_i,\ i = 0, 1, 2 \ldots, 12$
- Why 0? Because we need to constrain the inventory on 01/01 (as well as on 12/31)

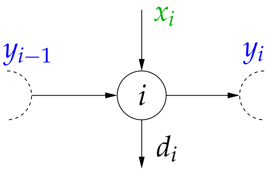
\+ a binary variable $z_i = \begin{cases} 1 & \text{if machine started in month } i \\ 0 & \text{otherwise} \end{cases}$

Production planning. What constraints?

- Production capacity constraint: $x_i \leq P \quad \forall i = 1, 2 \ldots, 12$
- Beginning/end of year: $y_0 = 0, \quad y_{12} = 0$
- What goes in must go out…

\+ production in month i only occurs[1] when the machine is used in i, i.e. $x_i > 0 \Rightarrow z_i = 1$:

$$x_i \leq P z_i$$



$$\Rightarrow \quad x_i + y_{i-1} = d_i + y_i \quad \forall i = 1, 2 \ldots, 12$$

[1] That is, it can be nonzero and at most P

Production planning model

$$
\begin{array}{lll}
\min & \sum_{i=1}^{12}(c_i x_i + f_i z_i) & \\
 & x_i + y_{i-1} = d_i + y_i & \forall i = 1, 2 \ldots, 12 \\
 & 0 \leq x_i \leq P z_i & \forall i = 1, 2 \ldots, 12 \\
 & y_i \geq 0, z_i \in \{0, 1\} & \forall i = 1, 2 \ldots, 12 \\
 & y_0 = y_{12} = 0 &
\end{array}
$$

Note: $x_i \leq P z_i$ implies $x_i \leq P$ as $z_i \leq 1$

Production planning model

```
set Months     = 1..12;
set MonthsPlus = 0..12;

param cost       {Months};
param fixed_cost {Months};
param demand     {Months};
param Capacity;

var production {Months}     >= 0;
var inventory  {MonthsPlus} >= 0;
var use        {Months} binary;

minimize prodCost:
  sum {i in Months} (cost       [i] * production [i] +
                     fixed_cost [i] * use        [i]);
```

Production planning model

```
conservation {i in Months}:
  production [i] + inventory [i-1] =
  demand     [i] + inventory [i];

startMachine {i in Months}:
  production [i] <= Capacity * use [i];

Jan1Inv:  inventory  [0] = 0;
Dec31Inv: inventory [12] = 0;
```

Problem data

```
param Capacity = 120;

param cost :=
1 14        2 19        3 15        4 11
5 10        6  7        7  4        8  5
9  7       10 10       11 11       12 13;

param demand :=
1 110       2  70       3  85       4 90
5 140       6  90       7  40       8 80
9 100      10 105      11 140      12 80;

param fixed_cost :=
1 435       2  470       3  425       4 390
5 340       6  290       7  240       8 280
9 300      10  345      11  390      12 380;
```

An “adequate” constant???

- Constraints such as $y_i \leq Mx_i$ are formally correct, but they are **horrible** for an MILP solver. The bigger M, the uglier.
- We still must ensure that y_i can take any feasible value.

$\Rightarrow$ If you know an explicit upper bound on y_i, use it.

- Physical constraint: for a truck traveling at 65mph, 24 hours a day, 365 days, $M = 569,400$mi (very ugly).
- “The company only works on weekdays 9am-5pm, drivers have an hour lunch break. Loading and unloading a truck takes 30 minutes”

$\Rightarrow$ 65mph$\times$(8h$-$1h-2×0.5h)/day$\times$200days/yr$= 78,000$mi