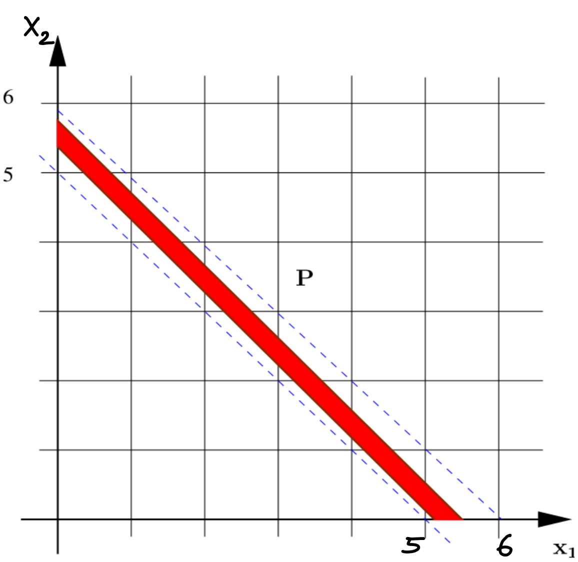
MATH 567: Lecture 2 (01/14/2025)

Today: * general forms of IP
* IP formulations

To round off (no pun intended!) the discussion on possibly rounding fractional solutions from LP relaxations to solve integer programs, we present another "extreme" example.

Consider the polytope P shown here, which is defined by the following constraints.

$$\left\{\begin{array}{l} 106 \le 21x_1 + 19x_2 \le 113 \\ 0 \le x_1, x_2 \le 6 \end{array}\right\} (P)$$



As is evident from the figure, $P \cap \mathbb{Z}^2 = \emptyset$!
The idea of rounding for <u>any IP</u> defined on P is moot here.

for any objective function

We will return to such examples later on.

General Forms of integer (linear) Programs

Mixed integer Program (MIP)

$$\begin{aligned} \max \quad & z = \bar{c}^T \bar{x} + \bar{d}^T \bar{y} \\ \text{s.t.} \quad & A\bar{x} + B\bar{y} \leq \bar{b} \\ & \bar{x} \in \mathbb{Z}^{n_1}_{\geq 0}, \; \bar{y} \in \mathbb{R}^{n_2}_{\geq 0} \end{aligned} \qquad \text{(MIP)}$$

Pure integer program (IP)

$$\begin{aligned} \max \quad & z = \bar{c}^T \bar{x} \\ \text{s.t.} \quad & A\bar{x} \leq \bar{b} \\ & \bar{x} \in \mathbb{Z}^{n}_{\geq 0} \end{aligned} \qquad \text{(IP)}$$

Special case of IP: Binary IP (BIP)

$$\begin{aligned} \max \quad & z = \bar{c}^T \bar{x} \\ \text{s.t.} \quad & A\bar{x} \leq \bar{b} \\ & \bar{x} \in \{0,1\}^{n} \end{aligned} \qquad \text{(BIP)}$$

Q When do we insist $x_i \in \mathbb{Z}$?

Should we build a new dorm? $\Rightarrow$ $x \in \{0,1\}$

fractional value does not make sense

insist on it.

How many rooms should we build?

$y = 385.6$ → 386, → 385 } both are okay in the big scheme of things.

So, just set $y \in \mathbb{R}_{\geq 0}$.

We will now look at several BIP and MIP formulation problems. It is important to remember that one **need not write these formulations in standard form**. Later on, when we describe algorithms to solve these problem instances, it will make sense to describe them for problems in standard form. Similarly, when we study the geometry or other properties of the associated polytope, we will do so using a standard form. In fact, it is better to write formulations in non-standard form if they are more readable!

We will introduce AMPL, which is a state-of-the-art **modeling software**. The function of such a software is to convert formulations written in non-standard form to, sometimes more concise, standard form. Then AMPL sends the standard form problem to a solver, e.g., Gurobi, which runs linear-algebra based algorithms to solve the same. AMPL then "interprets" the solution to the standard form problem back to the original form before displaying the same.

We will first introduce several instances of the BIP, and then the MIP. Later on, we will describe some unified procedures to model most situations using only binary variables, or using binary and continuous variables. Further on, we will discuss how to "compare" formulations – as it turns out, there are multiple ways to formulate the same situation, and one formulation might be "tighter" than the rest.

BIP formulations

1. Assignment Problem

n persons, n jobs, c_{ij} = cost of person i doing job j

Goal: Assign each person to a job so that total cost is minimum.

Step 1 decision variables (d.v.s)

Let $x_{ij} = \begin{cases} 1 & \text{if person } i \text{ does job } j, \\ 0 & \text{o.w.} \end{cases}$ n^2 vars.

o.w. → "otherwise"

Step 2: Constraints

$$\sum_{j=1}^{n} x_{ij} = 1 \quad \forall i \ (i=1,\dots,n) \quad \text{(person } i \text{ gets one job)}$$

$$\sum_{i=1}^{n} x_{ij} = 1 \quad \forall j \ (j=1,\dots,n) \quad \text{(job } j \text{ gets one person)}$$

$$x_{ij} \in \{0,1\} \quad \forall i,j$$

Step 3 Objective function

$$\min \ z = \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}x_{ij} \quad \text{(total cost)}$$

In summary

$$\min \ z = \sum_i\sum_j c_{ij}x_{ij} \quad \text{(total cost)}$$

s.t.

$$\sum_j x_{ij} = 1 \ \forall i \quad \text{(person } i \text{ gets 1 job)}$$

$$\sum_i x_{ij} = 1 \ \forall j \quad \text{(job } j \text{ gets 1 person)}$$

$$x_{ij} \in \{0,1\} \ \forall i,j$$

As you do more formulations, you will naturally go straight to the compact form, rather than write out the detailed steps. But do define the d.v.s first in all cases!

2. The 0-1 knapsack problem

* n projects.
* total budget b.
* cost of project j is a_j.
* value of project j is c_j.

It is assumed that you cannot undertake a fraction, e.g., 0.4., of a project.

Decide which projects to choose within the budget such that total value is maximized.

d.v's $x_j = 1$ if project j is selected, 0 o.w.

Constraints
$$\sum_{j=1}^{n} a_j x_j \le b \qquad \text{(budget)}$$
$$x_j \in \{0,1\} \;\; \forall j \qquad \text{(binary vars).}$$

objective function
$$\max \sum_{j=1}^{n} c_j x_j.$$

$$\begin{aligned} \max \quad & \sum c_j x_j \\ \text{s.t.} \quad & \sum a_j x_j \overset{=}{\le} b \\ & x_j \in \{0,1\} \;\; \forall j. \end{aligned}$$

equality knapsack problem: have to use up all the available budget

In feasibility knapsack, we want to find $x_j \in \{0,1\}$ s.t. $\sum a_j x_j = b$. (or, more generally, $\{b' \le \underbrace{\bar{a}^T \bar{x}}_{\sum a_j x_j} \le b\}$)

There is no objective function specified.

Additional Restrictions

1. If project 2 is chosen, so must be project 5.

$$x_5 \geq x_2$$

If $x_2 = 1$, then we have $x_5 \geq 1$, which forces $x_5 = 1$, as $x_5 \in \{0,1\}$.
But if $x_2 = 0$, we get $x_5 \geq 0$, which is redundant.

Notice that the reverse implication that "if project 5 is chosen, then so must project 2" is not forced by this constraint. Indeed, if $x_5 = 1$, we get $x_2 \leq 1$, which is redundant.

Note that $x_2 = x_5$ is also not correct here, as that constraint models "either pick both projects 2 and 5, or neither one."

2. Must
Can choose exactly ~~(at most)~~ 2 out of projects 1, 3, 6, 7.
Must at least

$$x_1 + x_3 + x_6 + x_7 \;\overset{=}{\underset{\geq}{\leq}}\; 2$$

It is important to realize that these constraints work as intended only when all x_j's are binary. Further, we do not want to force more than what is required. We will first try to write the required constraints just using logic, but will later describe a more systematic approach.

3. Fixed Charge (continued...)

$$\min \; f(x_1) + c_2x_2 + \cdots + c_nx_n$$
$$A\bar{x} \le \bar{b}$$
$$0 \le x_1 \le M_1$$
$$f(x_1) = \begin{cases} 0, & x_1 = 0 \\ f_1, & x_1 > 0 \end{cases} \quad (f_1 > 0).$$

YES/NO question: Is $x_1 > 0$?

$\Rightarrow$ model using $y_1 \in \{0,1\}$.

We want
$$y_1 = \begin{cases} 1 & \text{if } x_1 > 0 \\ 0 & \text{o.w.} \end{cases}$$

$$\boxed{\begin{array}{ll} \min & f_1y_1 + c_2x_2 + \cdots + c_nx_n \\ \text{s.t.} & A\bar{x} \le \bar{b} \\ & 0 \le x_1 \le M_1y_1 \\ & y_1 \in \{0,1\} \end{array}}$$

If $x_1 > 0$, y_1 is forced to 1. Else, $x_1 \le My_1$ will not hold (with $y_1 = 0$).

If $x_1 = 0$, $x_1 \le M_1y_1$ can hold with $y_1 = 0$ or $y = 1$. But the term f_1y_1 in the min objective function forces $y_1 = 0$ in the optimal solution. Recall, $f_1 > 0$ here.

4. Interactive fixed charge $\longrightarrow$ Included in HW1!

Similar to Problem 3 but

$$f(x_1, x_2) = \begin{cases} 0, & \text{if } x_1 = x_2 = 0 \\ f_1, & \text{if } x_1 > 0,\ x_2 = 0 \\ f_2, & \text{if } x_1 = 0,\ x_2 > 0 \\ f_{12}, & \text{if } x_1 > 0\ \&\ x_2 > 0 \end{cases}$$

with $0 \le x_1 \le M_1$, $0 \le x_2 \le M_2$, $f_{12} \geq f_1 > 0$, $f_{12} \geq f_2 > 0$, $f_{12} \neq f_1 + f_2$

need not hold as equality; indeed $f_{12} \gtreqless f_1 + f_2$ are all ok.

Can we use $y_1, y_2 \in \{0,1\}$, and $y_1 \times y_2$? Yes, but $y_1 y_2$ is nonlinear, so you want to somehow linearize it — maybe define $y_{12} \in \{0,1\}$, and relate it to y_1 and y_2 using extra constraints.

Later on, we will talk about linearizing such nonlinear terms - products of binary variables, or x_i^2 when $x_i \in \{0,1\}$, etc.