IE 400: Principles of Engineering Management

Set 5: Integer Programming

Integer Programs

- Integer Program: a linear program plus the additional constraints that some or all of the variables must be integer valued.

- We also permit "$x_j \in \{0,1\}$", or equivalently, x_j is binary.

- This is a shortcut for writing the constraint:

  $0 \leq x_j \leq 1$ and x_j is integer

Why Integer Programs?

- Advantages of restricting variables to take on integer values:
  - More realistic
  - More flexibility

- Disadvantages
  - More difficult to model
  - Can be <u>much</u> more difficult to solve

A 2-Variable Integer Program

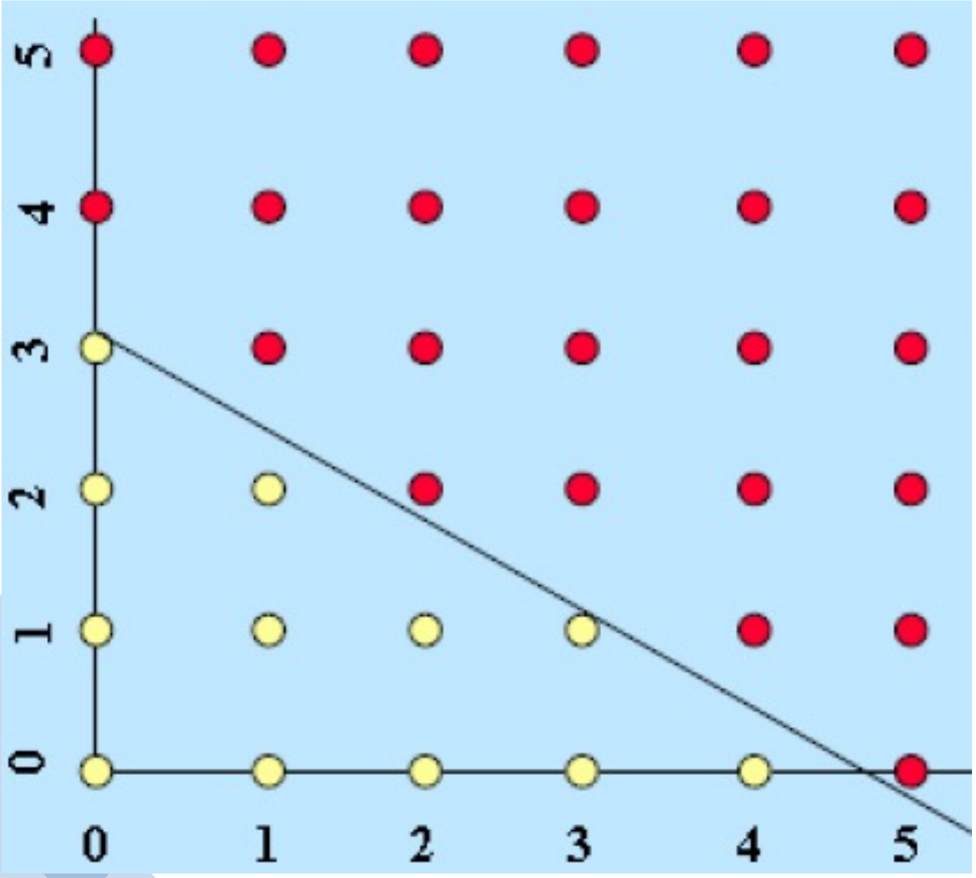
maximize $3x + 4y$

subject to $5x + 8y \leq 24$

$x, y \geq 0$ and integer

- What is the optimal solution?

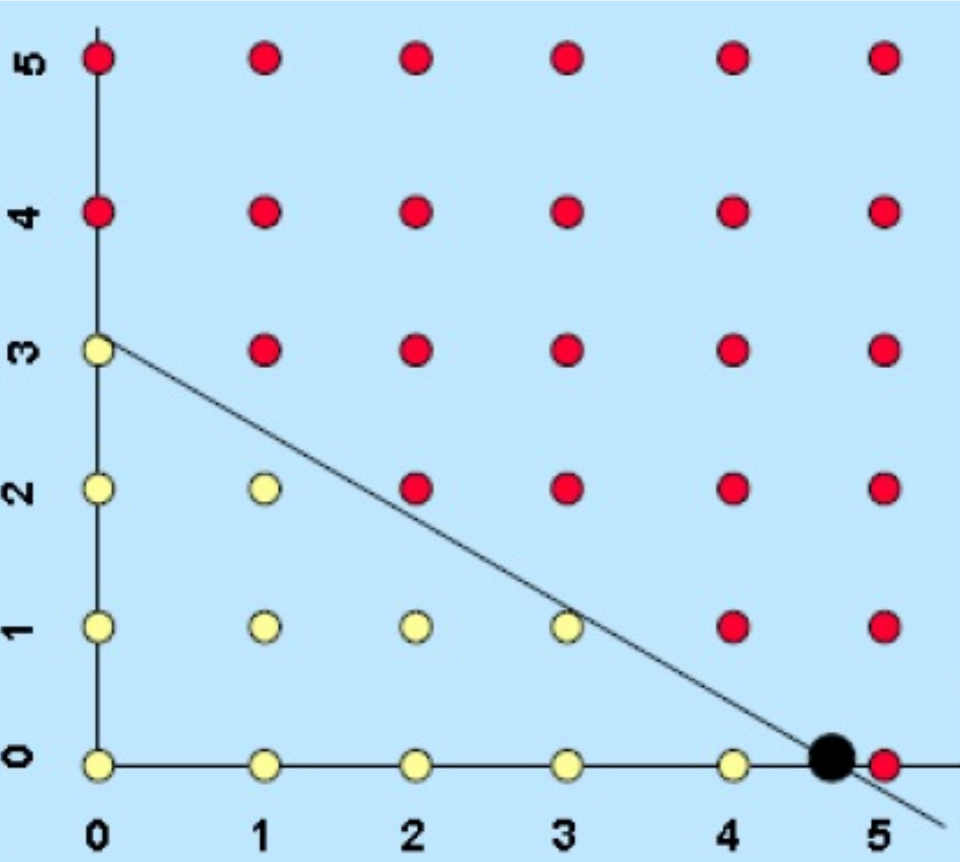
The Feasible Region



- **Question:** What is the optimal integer solution?
- What is the optimal linear solution?
- Can one use linear programming to solve the integer program?

The Rounding Technique

sometimes useful, not always



- Solve LP (ignore integrality). Get x=24/5, y=0, and z=14 2/5.
- Round. Get x=5, y=0. Infeasible.
- Truncate. Get x=4, y=0, z=12.
- Same objective value at x=0, y=3.
- Optimal is x=3, y=1, and z=13.

Integer Programming

- So far, we have considered problems under the following assumptions:
  i. Proportionality & Additivity
  ii. Divisibility
  iii. Certainty

- While many problems satisfy these assumptions, there are other problems inn which we will need to relax these assumptions. For example, consider the following bus scheduling problem:

Ex: Bus Scheduling

Periods	**1**	**2**	**3**	**4**	**5**	**6**
Hours	0-4 AM	4-8 AM	8 AM – 12 PM	12 - 4 PM	4 – 8 PM	8 PM-0AM
Min. # of buses required	4	8	10	7	12	4

- Each bus starts to operate at the beginning of a period and operates for **8 consecutive hours and then receives 16 hours off.**
- For example, a bus operating from 4 AM to 12 PM must be off between 12 PM and 4 AM.
- Find the minimum # of busses to provide the required service.

Ex: Bus Scheduling

Periods	**1**	**2**	**3**	**4**	**5**	**6**
Hours	0-4 AM	4-8 AM	8 AM – 12 PM	12 - 4 PM	4 – 8 PM	8 PM-0AM
Min. # of buses required	4	8	10	7	12	4

- How do you define your decision variables here?

a) x_i: # of busses operating in period i, i=1,2,…,6

b) x_i: # of busses starting to operate in period i, i=1,2,…,6

Ex: Bus Scheduling

- The alternative “b” makes modeling of this problem much easier.
- It is not possible to keep track of the total number of busses under alternative “a”. (WHY?)
- The model is as follows:

Ex: Bus Scheduling

- The alternative “b” makes modeling of this problem much easier.
- It is not possible to keep track of the total number of busses under alternative “a”. (WHY?)
- The model is as follows:

$$\min x_1 + x_2 + x_3 + x_4 + x_5 + x_6$$

$$\text{s.t.}$$

$$\begin{aligned} x_1 + x_6 &\geq 4 \\ x_1 + x_2 &\geq 8 \\ x_2 + x_3 &\geq 10 \\ x_3 + x_4 &\geq 7 \\ x_4 + x_5 &\geq 12 \\ x_5 + x_6 &\geq 4 \end{aligned}$$

$$x_i \geq 0,\ x_i \text{ integer},\ i = 1, \ldots 6$$

Ex: Bus Scheduling

- Note that each decision variable represents the number of busses.
- Clearly, it does not make sense to talk about 3.5 busses because busses are not divisible.
- Hence, we should impose the additional condition that each of $x_1, \ldots,$ x_6 can only take integer values.
- If some or all of the variables are restricted to be integer valued, we call it an **integer program (IP)**.
- This last example is hence an IP problem.

Work Scheduling

Burger Queen would like to determine the min # of kitchen employees. Each employee works six days a week and takes the seventh day off.

Days	Mon	Tue	Wed	Thu	Fri	Sat	Sun
Min. # of workers	8	6	7	6	9	11	9

- How do you formulate this problem?
- The objective function?
- The constraints?

Work Scheduling

Burger Queen would like to determine the min # of kitchen employees. Each employee works six days a week and takes the seventh day off.

Days	**Mon**	**Tue**	**Wed**	**Thu**	**Fri**	**Sat**	**Sun**
Min. # of workers	8	6	7	6	9	11	9

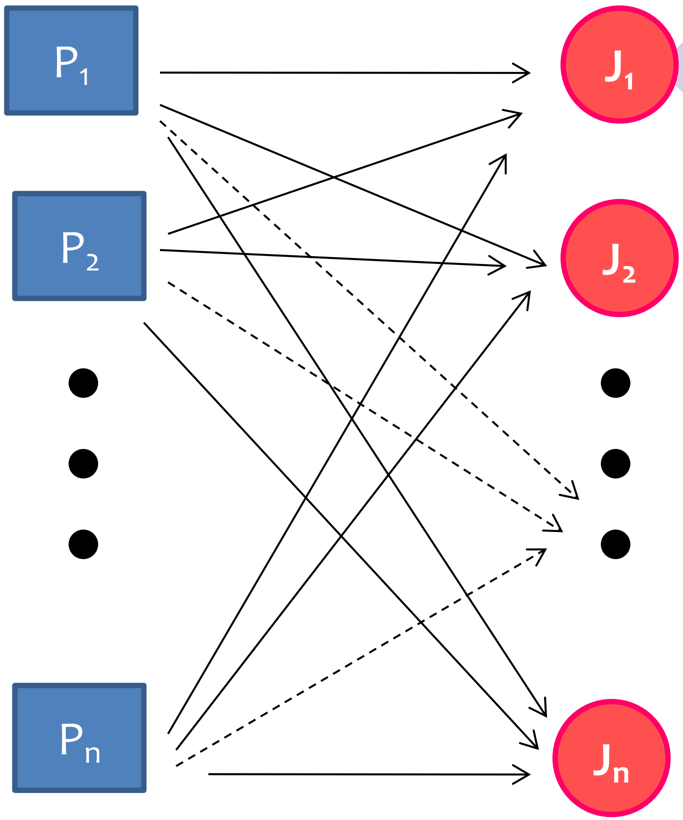
- x_i : # of workers who are <u>not working</u> on day i, $i = 1, \ldots, 7$ (1: Mon, 7: Sun)

Work Scheduling

- Decision Variables:
- IP Model:

The Assignment Problem

- There are n people and n jobs to carry out.
- Each person is assigned to carry out exactly one job.
- If person i is assigned to job j, then the cost is c_{ij} .



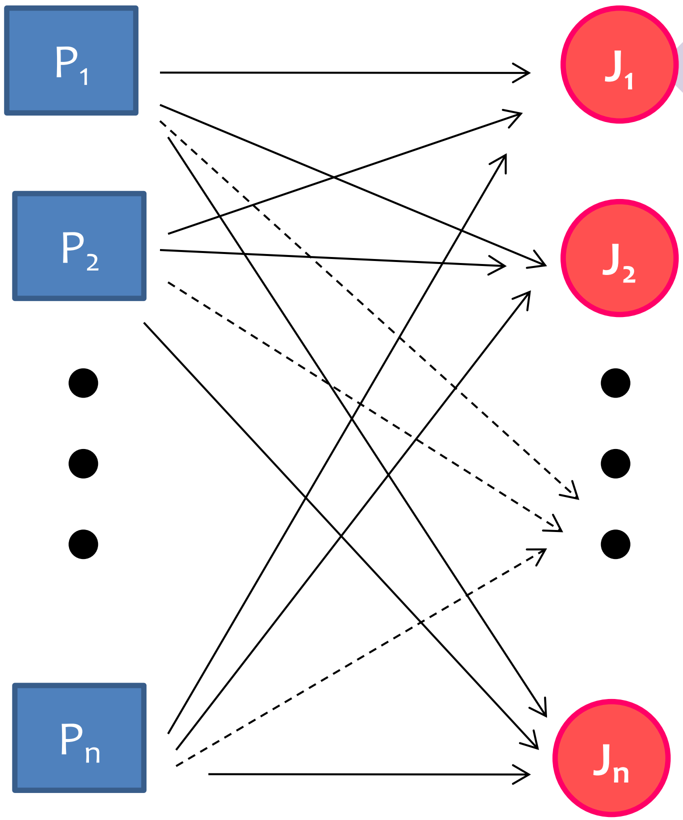
Find an assignment of n people to n jobs with minimum total cost.

The Assignment Problem

$$x_{ij} = \begin{cases} 1, & \text{person } i \text{ is assigned to job } j \\ 0, & \text{otherwise} \end{cases}$$

$$i = 1,2,\ldots,n\,; \quad j = 1,2,\ldots,n$$

- Such variables are called BINARY VARIABLES and offer great flexibility in modeling. They are ideal for YES/NO decisions.



- So how is the problem formulated?

The Assignment Problem

- The problem is formulated as follows:

$$\min \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij}$$

$$s.t.$$

$$\sum_{j=1}^{n} x_{ij} = 1, \qquad i = 1,\dots,n \qquad \text{(person } i \text{ is assigned to exactly one job)}$$

$$\sum_{i=1}^{n} x_{ij} = 1, \qquad j = 1,\dots,n \qquad \text{(job } j \text{ is assigned to exactly one person)}$$

$$x_{ij} \in \{0,1\}, i = 1, \dots n;\ j = 1, \dots n$$

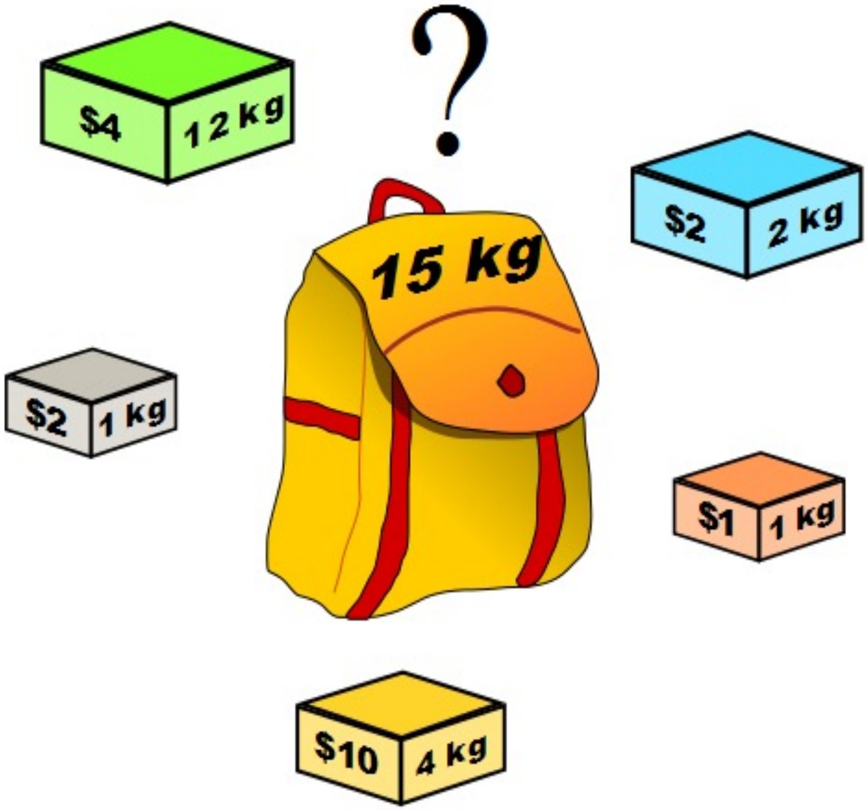
- This is still an example of an IP problem since x_{ij} can only take one of the two integer values 0 and 1.

The 0-1 Knapsack Problem

You are going on a trip. Your **Suitcase has a capacity of b kg's.**

- You have **n different items** that you can pack into your suitcase 1, . . . , n.
- **Item i** has a weight of $\mathbf{a_i}$ and gives you a utility of $\mathbf{c_i}$.

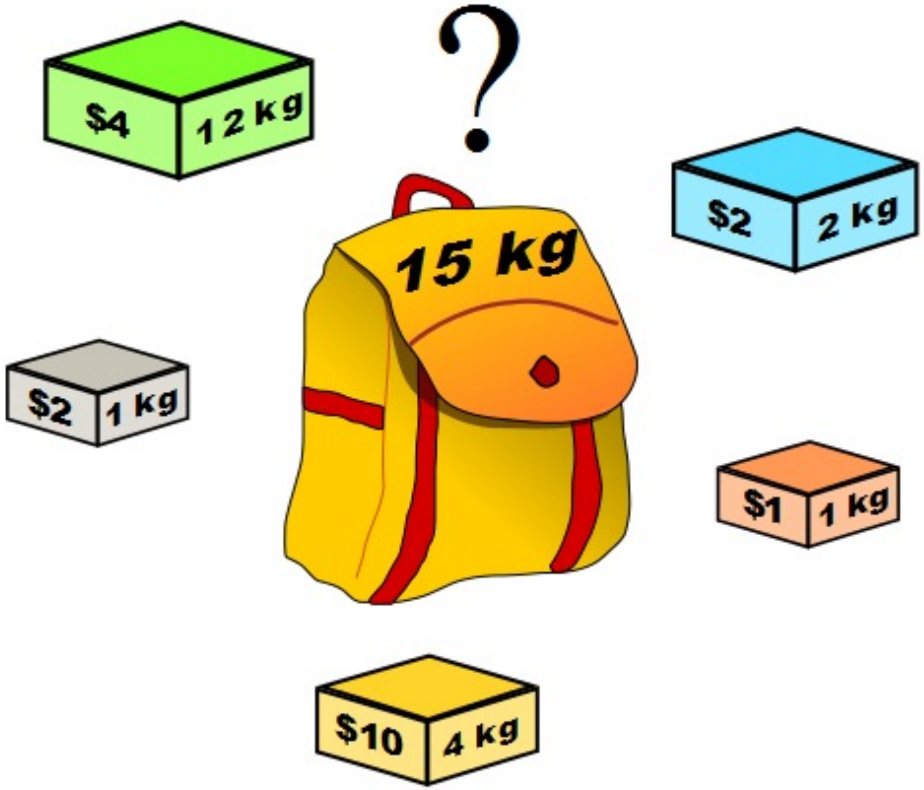
How should you pack your suitcase to maximize total utility?



The 0-1 Knapsack Problem

Decision Variables:

IP Model:



Workforce Scheduling

- Monthly demand for a shoe company is:

	Month			
	1	**2**	**3**	**4**
Demand	300	500	100	100

- A pair of shoes requires 4 hours of labor and $5 worth of raw materials
- 50 pairs of shoes in stock
- Monthly inventory cost is charged for shoes in stock at the end of each month at $30/pair
- Currently 3 workers are available
- Salary $1500/month
- Each worker works 160 hours/month
- Hiring a worker costs $1600
- Firing a worker costs $2000

Workforce Scheduling

- Monthly demand for a shoe company is:

	Month			
	1	**2**	**3**	**4**
Demand	300	500	100	100

- A pair of shoes requires 4 hours of labor and $5 worth of raw materials
- 50 pairs of shoes in stock
- Monthly inventory cost is charged for shoes in stock at the end of each month at $30/pair
- Currently 3 workers are available
- Salary $1500/month
- Each worker works 160 hours/month
- Hiring a worker costs $1600
- Firing a worker costs $2000

How to meet the demand
With minimum total cost?

IP Model for Workforce Scheduling

- **Decision Variables:**

- **IP Model:**

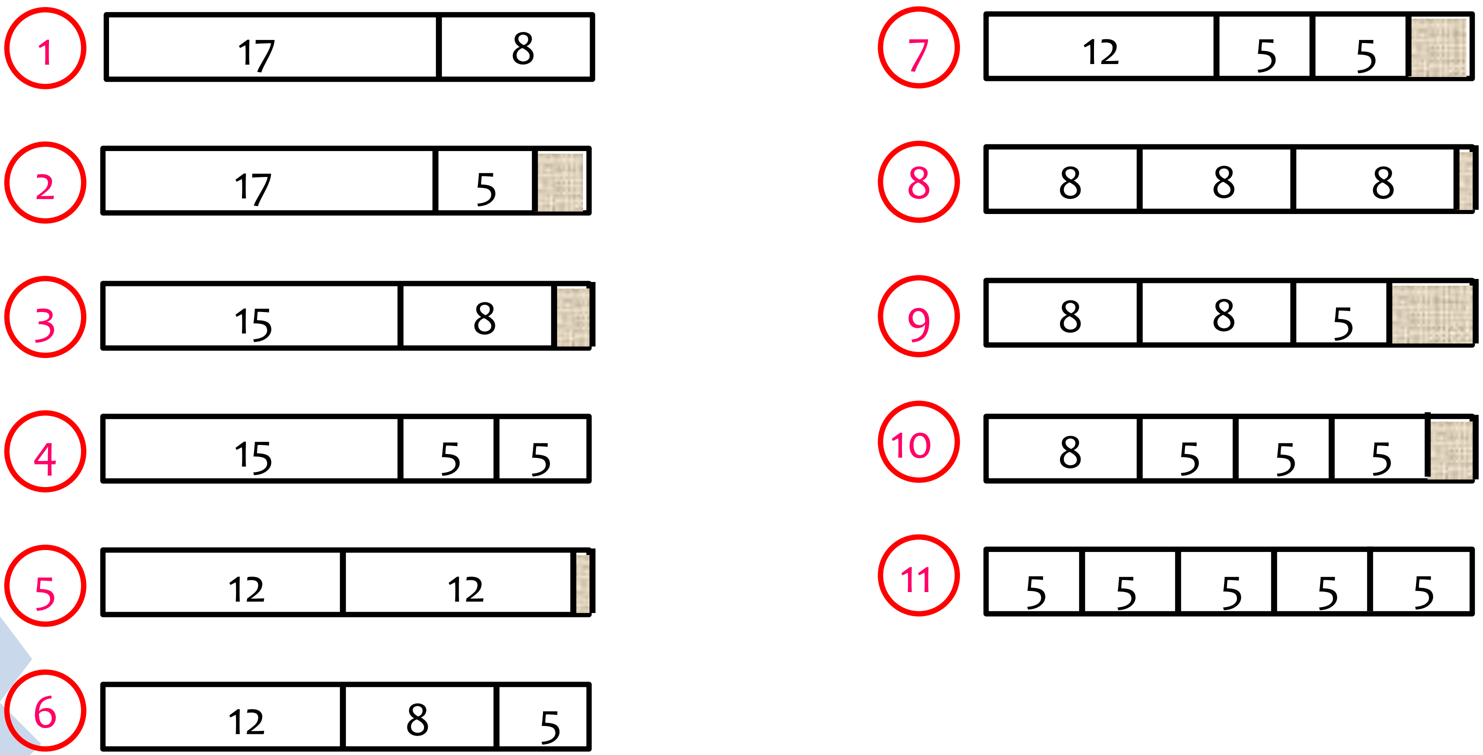
The Cutting Stock Problem

- A paper company manufactures and sells rolls of paper of fixed width in 5 standard lengths: 5, 8, 12, 15 and 17 m. Demand for each size is given below.
- It produces 25 m length rolls and cuts orders from its stock of this size.
- What is the min # of rolls to be used to meet the total demand?

Length (meters)	5	8	12	15	17
Demand	40	35	30	25	20

The Cutting Stock Problem

- **First, identify the cutting patterns**



The Cutting Stock Problem

- **Then how do you define the decision variable?**

x_j : # of rolls cut according to pattern *j*.

The Cutting Stock Problem

- **Then how do you define the decision variable?**

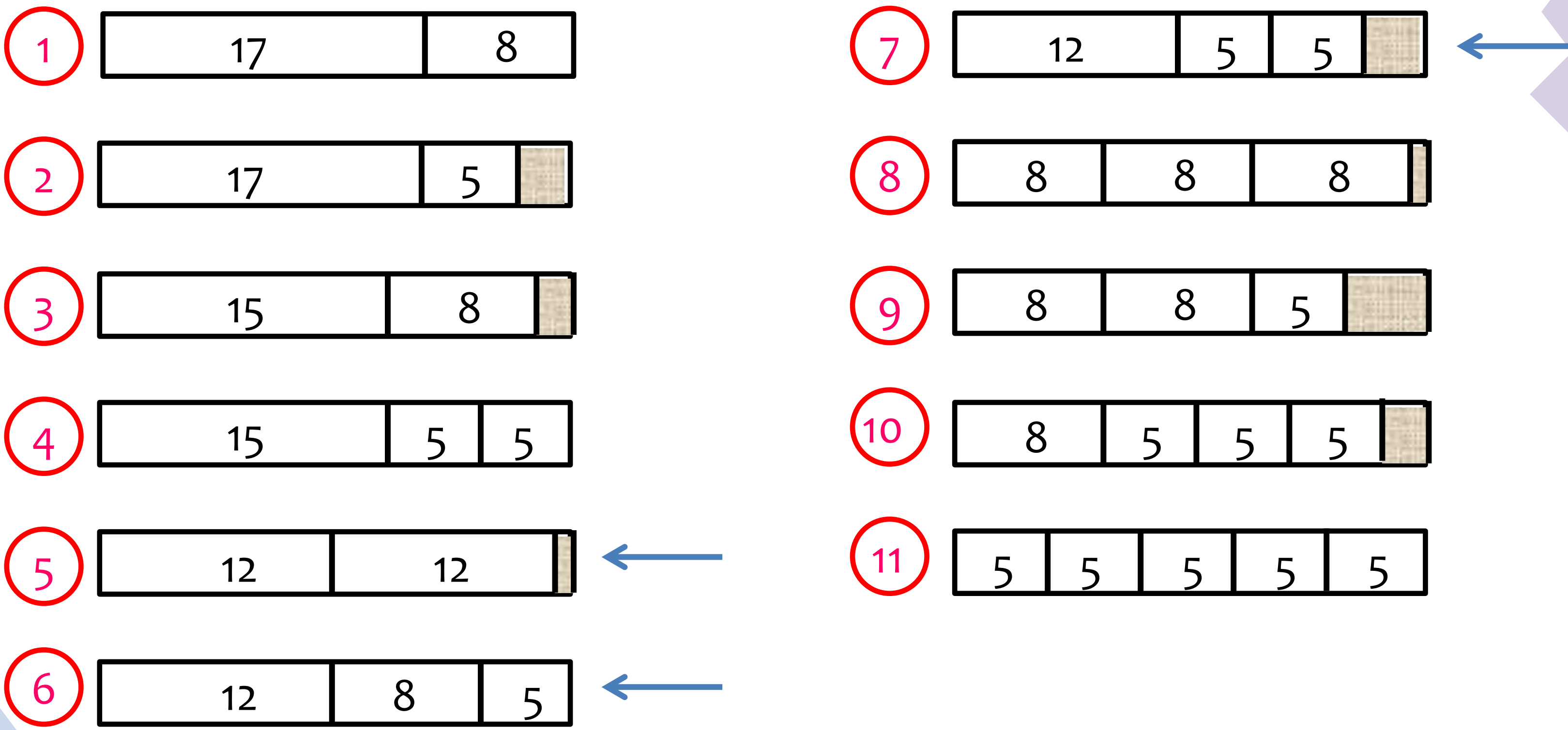
  x_j : # of rolls cut according to pattern *j*.

- **What about the constraints?**
  - e.g., How do you write the constraint for the demand of 12 m rolls of paper?
  - We look at the patterns that contain 12 m pieces, and then write the constraint based on x_j.

Length (meters)	5	8	12	15	17
Demand	40	35	30	25	20

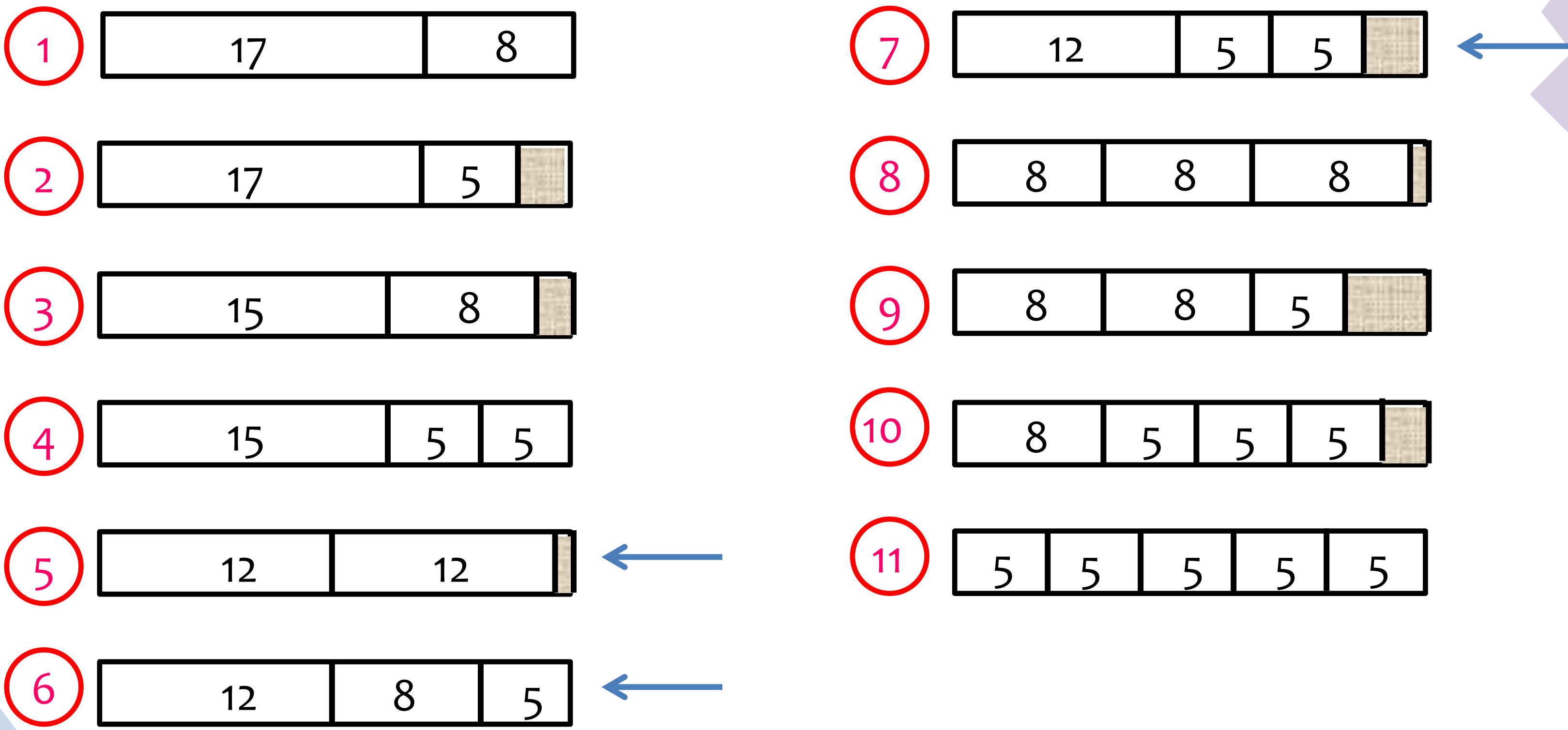
The Cutting Stock Problem

The cutting patterns



The Cutting Stock Problem

The cutting patterns



$$2x_5 + x_6 + x_7 \geq 30$$

Why do we have "≥" sign?

The Cutting Stock Problem

- **Then how do you define the decision variable?**

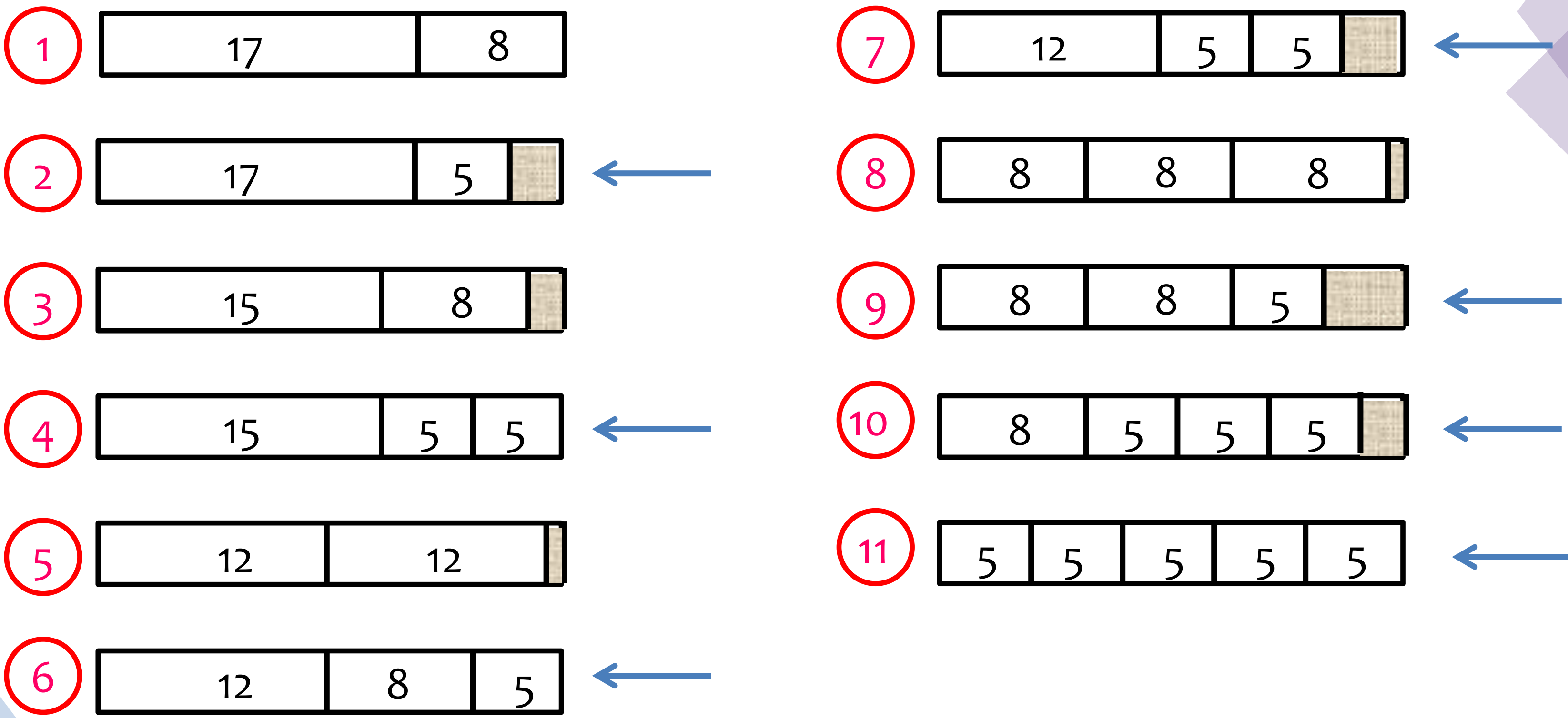
  x_j : # of rolls cut according to pattern *j*.

- **What about the constraints?**
  - e.g., What about the demand for 5m?

Length (meters)	5	8	12	15	17
Demand	40	35	30	25	20

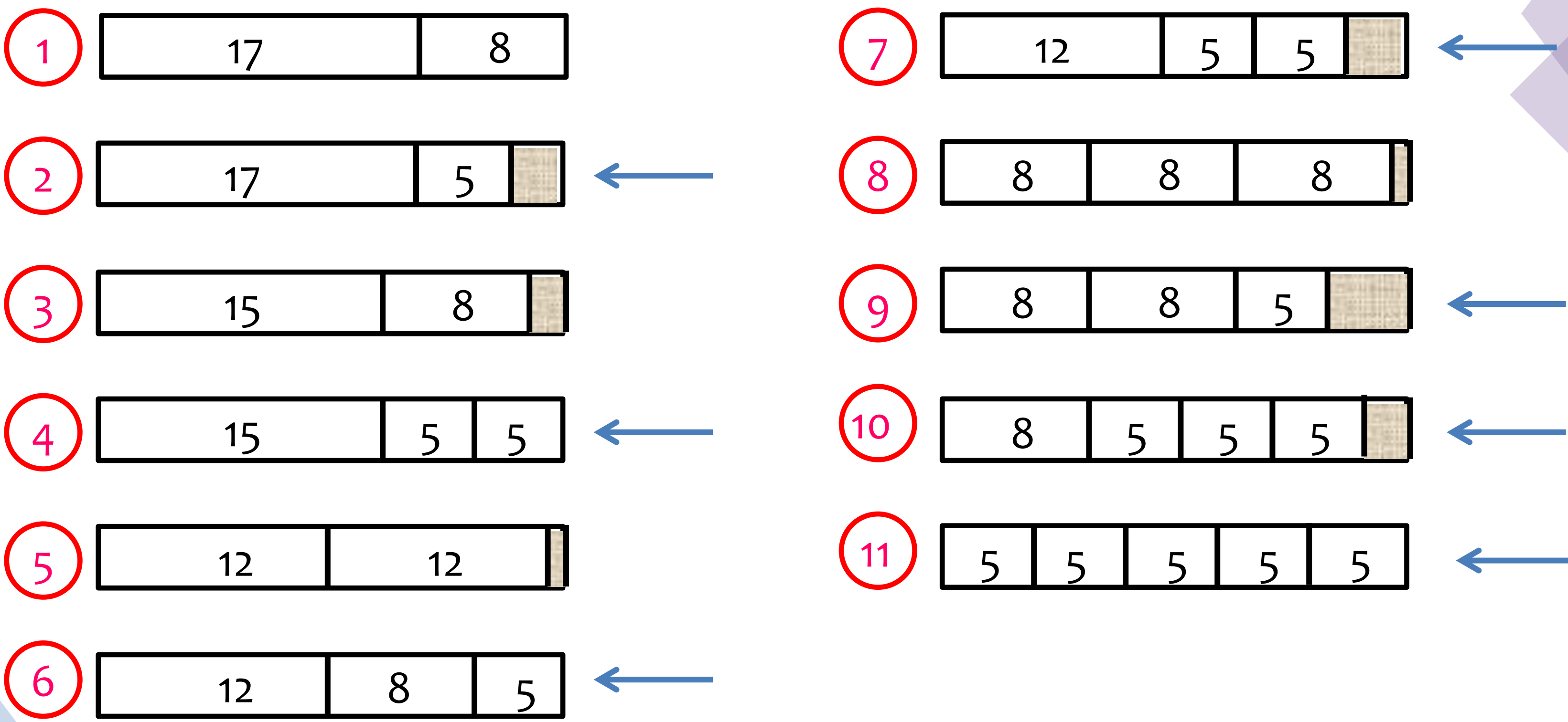
The Cutting Stock Problem

The cutting patterns



The Cutting Stock Problem

The cutting patterns



$$x_2 + 2x_4 + x_6 + 2x_7 + x_9 + 3x_{10} + 5x_{11} \geq 40$$

The Cutting Stock Problem

- The problem is formulated as follows:

$$\min \sum_{j=1}^{11} x_j$$

$$s.t.$$

$$x_2 + 2x_4 + x_6 + 2x_7 + x_9 + 3x_{10} + 5x_{11} \geq 40$$

$$x_1 + x_3 + x_6 + 3x_8 + 2x_9 + x_{10} \geq 35$$

$$2x_5 + x_6 + x_7 \geq 30$$

$$x_3 + x_4 \geq 25$$

$$x_1 + x_2 \geq 20$$

$$x_j \geq 0,\ j = 1, \ldots, 11,\ x_j \text{ integer.}$$

Project Selection

A manager should choose from among 10 different projects:

p_j : profit if we invest in project j , j = 1, . . . , 10

c_j : cost of project j , j = 1, . . . , 10

q: total budget available

There are also some additional requirements:

i. Projects 3 and 4 cannot be chosen together
ii. Exactly 3 projects are to be selected
iii. If project 2 is selected, then so is project 1
iv. If project 1 is selected, then project 3 should not be selected
v. Either project 1 or project 2 is chosen, but not both
vi. Either both projects 1 and 5 are chosen or neither are chosen
vii. At least two and at most four projects should be chosen from the set {1, 2, 7, 8, 9, 10}
viii. Neither project 5 nor 6 can be chosen unless either 3 or 4 is chosen
ix. At least two of the investments 1, 2, 3 are chosen, or at most three of the investments 4, 5, 6, 7, 8 are chosen.
x. If both 1 and 2 are chosen, then at least one of 9 and 10 should also be chosen.

Project Selection

- Under these requirements, the objective is to maximize the profit.

- Decision Variables:

$$x_j = \begin{cases} 1, & \text{if project } j \text{ is chosen} \\ 0, & \text{otherwise} \end{cases}$$

$$j = 1, \ldots, 10$$

Project Selection

- **The objective function is:**

$$\max \sum_{j=1}^{10} \mathrm{p}_j x_j$$

Project Selection

- **The objective function is:**

$$\max \sum_{j=1}^{10} p_j x_j$$

- **First, the budget constraint:**

$$\sum_{j=1}^{10} c_j x_j \leq q$$

Project Selection

- **The objective function is:**

$$\max \sum_{j=1}^{10} p_j x_j$$

- **First, the budget constraint:**

$$\sum_{j=1}^{10} c_j x_j \le q$$

i. **Projects 3 and 4 cannot be chosen together**

$$x_3 + x_4 \le 1$$

Project Selection

ii. **Exactly 3 projects are to be chosen:**

$$x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8 + x_9 + x_{10} = 3$$

iii. **If project 2 is chosen, then so is project 1:**

$$x_2 \le x_1$$

Project Selection

iv. If project 1 is chosen, then project 3 should not be chosen:

$$x_1 \leq 1 - x_3$$

v. Either project 1 or project 2 is chosen, but not both:

$$x_1 + x_2 = 1$$

Project Selection

vi. Either both projects 1 and 5 are chosen or neither are chosen:

$$x_1 = x_5 \quad \text{or} \quad x_1 - x_5 = 0$$

vii. At least two and at most four projects should be chosen from the set {1,2,7,8,9,10}:

$$x_1 + x_2 + x_7 + x_8 + x_9 + x_{10} \geq 2$$

$$x_1 + x_2 + x_7 + x_8 + x_9 + x_{10} \leq 4$$

Project Selection

viii. Neither project 5 nor 6 can be chosen unless either 3 or 4 is chosen:

- **Equivalently,** $\text{If } x_3 + x_4 = 0 \quad \text{then} \quad x_5 = x_6 = 0$

$$x_5 \le x_3 + x_4$$

$$x_6 \le x_3 + x_4$$

- **Another possibility,**

$$x_5 + x_6 \le 2(x_3 + x_4)$$

Project Selection

ix. At least two of the investments {1,2,3} are chosen, or at most three of the investments {4,5,6,7,8} are chosen:

- It is more difficult to express this requirement in mathematical terms due to the "or" operator!

$$\underset{(\mathrm{I})}{x_1 + x_2 + x_3 \geq 2} \qquad \text{or} \qquad \underset{(\mathrm{II})}{x_4 + x_5 + x_6 + x_7 + x_8 \leq 3}$$

- So, we want to ensure that at least one of (I) and (II) is satisfied.

Either-Or Constraints

- The following situation commonly occurs in mathematical programming problems. We are given two constraints of the form,

$$f(x_1, x_2, \ldots . x_n) \leq 0 \qquad (\mathrm{I})$$

$$g(x_1, x_2, \ldots . x_n) \leq 0 \qquad (\mathrm{II})$$

Either-Or Constraints

- The following situation commonly occurs in mathematical programming problems. We are given two constraints of the form,

$$f(x_1, x_2, \ldots. x_n) \le 0 \qquad (\mathrm{I})$$

$$g(x_1, x_2, \ldots. x_n) \le 0 \qquad (\mathrm{II})$$

- We want to make sure that at least one of (I) and (II) is satisfied. These are often called either-or constraints.

Either-Or Constraints

- The following situation commonly occurs in mathematical programming problems. We are given two constraints of the form,

$$f(x_1, x_2, \dots. x_n) \le 0 \qquad \text{(I)}$$

$$g(x_1, x_2, \dots. x_n) \le 0 \qquad \text{(II)}$$

- We want to make sure that at least one of (I) and (II) is satisfied. These are often called either-or constraints.
- Adding two new constraints as below to the formulation will ensure that <u>at least one of (I) or (II) is satisfied</u>,

$$f(x_1, x_2, \dots. x_n) \le My \qquad \text{(I}')$$

$$g(x_1, x_2, \dots. x_n) \le M(1-y) \qquad \text{(II}')$$

where $y \in \{0, 1\}$.

Either-Or Constraints

- ***M*** is a number chosen **large enough to ensure that** *$f(x_1, x_2, \ldots, x_n) \leq M$ and $g(x_1, x_2, \ldots, x_n) \leq M$ are satisfied for all values of $x_1, x_2, \ldots, x_n$ that satisfy the other constraints in the problem.*

Either-Or Constraints

$$f\left(x_1, x_2, \ldots . x_n\right) \leq My \qquad \left(\mathrm{I}'\right)$$

$$g\left(x_1, x_2, \ldots . x_n\right) \leq M\left(1-y\right) \qquad \left(\mathrm{II}'\right)$$

where $y \in \{0,1\}$.

Either-Or Constraints

$$f\left(x_1, x_2, \ldots . x_n\right) \leq My \qquad (\mathrm{I}')$$

$$g\left(x_1, x_2, \ldots . x_n\right) \leq M\left(1-y\right) \qquad (\mathrm{II}')$$

where $y \in \{0,1\}$.

a) If y=0, then (I') and (II') becomes:

$$f\,(x_1, x_2, \ldots, x_n) \leq 0,$$

and

$$g\,(x_1, x_2, \ldots, x_n) \leq M.$$

Hence, this means that (I) must be satisfied, and possibly (II) is satisfied.

Either-Or Constraints

$$f\left(x_1, x_2, \ldots . x_n\right) \leq My \qquad (\mathrm{I}')$$

$$g\left(x_1, x_2, \ldots . x_n\right) \leq M\left(1 - y\right) \qquad (\mathrm{II}')$$

where $y \in \{0,1\}$.

b) If y=1, then (I') and (II') becomes:

$$f\,(x_1, x_2, \ldots, x_n) \leq M,$$

and

$$g\,(x_1, x_2, \ldots, x_n) \leq 0.$$

Hence, this means that (II) must be satisfied, and possibly (I) is satisfied.

Either-Or Constraints

$$f\left(x_1, x_2, \ldots . x_n\right) \le My \qquad (\mathrm{I}')$$

$$g\left(x_1, x_2, \ldots . x_n\right) \le M\left(1-y\right) \qquad (\mathrm{II}')$$

where $y \in \{0,1\}$.

Therefore, whether y=0 or y=1, (I') and (II') ensure that at least one of (I) and (II) is satisfied.

$$f\left(x_1, x_2, \ldots . x_n\right) \le 0 \qquad (\mathrm{I})$$

$$g\left(x_1, x_2, \ldots . x_n\right) \le 0 \qquad (\mathrm{II})$$

Project Selection

Now, let us return to our problem:

ix. At least two of the investments {1,2,3} are chosen, or at most three of the investments {4,5,6,7,8} are chosen:

$$\underset{\text{(I)}}{x_1 + x_2 + x_3 \geq 2} \qquad \text{or} \qquad \underset{\text{(II)}}{x_4 + x_5 + x_6 + x_7 + x_8 \leq 3}$$

- This is an example of either-or constraint. Hence by defining the f(.) and g(.) functions, (I) and (II) becomes:

$$\underbrace{2\text{-}\left(x_1 + x_2 + x_3\right)}_{f(.)} \leq 0$$

$$\underbrace{-3 + x_4 + x_5 + x_6 + x_7 + x_8}_{g(.)} \leq 0$$

Project Selection

Now, let us return to our problem:

ix. At least two of the investments {1,2,3} are chosen, or at most three of the investments {4,5,6,7,8} are chosen:

$$\underset{\text{(I)}}{x_1 + x_2 + x_3 \geq 2} \quad \text{or} \quad \underset{\text{(II)}}{x_4 + x_5 + x_6 + x_7 + x_8 \leq 3}$$

- This is an example of either-or constraint. Hence by defining the f(.) and g(.) functions, (I) and (II) becomes:

$$\underbrace{2 - (x_1 + x_2 + x_3)}_{f(.)} \leq 0$$

$$\underbrace{-3 + x_4 + x_5 + x_6 + x_7 + x_8}_{g(.)} \leq 0$$

M=2 works!

We have to find M large enough to ensure that f(.)≤M and g(.)≤M for all values of $x_1,\dots,x_n$.

Project Selection

Now, let us return to our problem:

ix. At least two of the investments {1,2,3} are chosen, or at most three of the investments {4,5,6,7,8} are chosen:

$$x_1 + x_2 + x_3 \geq 2 \qquad \text{or} \qquad x_4 + x_5 + x_6 + x_7 + x_8 \leq 3$$

$$\text{(I)} \qquad\qquad\qquad\qquad \text{(II)}$$

- Putting M=2 and rearranging the terms, we have:

$$x_1 + x_2 + x_3 \geq 2(1 - \mathrm{y})$$

$$x_4 + x_5 + x_6 + x_7 + x_8 \leq 5 - 2y$$

$$\mathrm{y} \in \{0,1\}$$

Project Selection

x. If both 1 and 2 are chosen, then at least one of 9 and 10 should also be chosen.

- Both projects 1 and 2 are chosen if $x_1 + x_2 = 2$.
- At least one of projects 9 and 10 is chosen if $x_9 + x_{10} \geq 1$

Hence, we have $x_1 + x_2 = 2 \Rightarrow x_9 + x_{10} \geq 1$

- So, how to express this as a linear constraint?

If-Then Constraints

- The following situation commonly occurs in mathematical programming problems. We want to ensure that:

If a constraint $f(x_1,\ldots,x_n)>0$ *is satisfied,*

then $g(x_1,\ldots,x_n)\leq 0$ *must be satisfied;*

while if a constraint $f(x_1,\ldots,x_n)>0$ *is not satisfied,*

then $g(x_1,\ldots,x_n)\leq 0$ *may or may not be satisfied.*

If-Then Constraints

- In other words, we want to ensure that $f(x_1,\dots,x_n)>0$ implies $g(x_1,\dots,x_n)\leq 0$.

- Logically, this is equivalent to:

$$f(x_1, x_2, \dots, x_n) \leq 0 \qquad (\mathrm{I})$$

or

$$g(x_1, x_2, \dots, x_n) \leq 0 \qquad (\mathrm{II})$$

If-Then Constraints

$$f(x_1, x_2, \ldots, x_n) \leq 0 \qquad (\mathrm{I})$$

$$\text{or}$$

$$g(x_1, x_2, \ldots, x_n) \leq 0 \qquad (\mathrm{II})$$

Now, these are either-or type constraints, and we can handle them as before by defining a binary variable, say z:

$$f(x_1, x_2, \ldots, x_n) \leq Mz$$

$$g(x_1, x_2, \ldots, x_n) \leq M(1 - z)$$

$$\text{where } z \in \{0,1\}$$

where, as before, M is a large enough number such that $f(.)\leq M$ and $g(.)\leq M$ holds for all values $x_1, \ldots, x_n$.

Project Selection

Now, let us return to our problem:

x. If both 1 and 2 are chosen, then at least one of 9 and 10 should also be chosen.

$$x_1 + x_2 = 2 \Rightarrow x_9 + x_{10} \geq 1$$

- We need to define *f(.)>0* and g*(.)≤0* constraints:

$$x_1 + x_2 > 1 \quad \Rightarrow \quad x_9 + x_{10} \geq 1$$

$$\textit{equivalently}$$

$$\underbrace{x_1 + x_2 - 1}_{\text{f(.)}} > 0 \quad \Rightarrow \quad \underbrace{1 - (x_9 + x_{10})}_{\text{g(.)}} \leq 0$$

Project Selection

x. If both 1 and 2 are chosen, then at least one of 9 and 10 should also be chosen.

$$x_1 + x_2 = 2 \Rightarrow x_9 + x_{10} \geq 1$$

$$\underbrace{x_1 + x_2 - 1}_{f(.)} > 0 \quad \Rightarrow \quad \underbrace{1 - (x_9 + x_{10})}_{g(.)} \leq 0$$

Now converting to either-or, we have:

$$x_1 + x_2 - 1 \leq 0 \qquad \text{(I)}$$

$$or$$

$$1 - (x_9 + x_{10}) \leq 0 \qquad \text{(II)}$$

Project Selection

$$x_1 + x_2 - 1 \leq 0 \qquad \text{(I)}$$

$$or$$

$$1 - (x_9 + x_{10}) \leq 0 \qquad \text{(II)}$$

- To linearize the above constraints, define a binary variable, say z and a large enough value *M*:

$$x_1 + x_2 - 1 \leq Mz$$

$$1 - (x_9 + x_{10}) \leq \mathrm{M}(1 - z)$$

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

- M=1 works and after rearranging we have:

$$x_1 + x_2 \leq 1 + z$$

$$x_9 + x_{10} \geq z$$

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

Project Selection

x. If both 1 and 2 are chosen, then at least one of 9 and 10 should also be chosen.

$$x_1 + x_2 \le 1 + z$$

$$x_9 + x_{10} \ge z$$

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

Project Selection

Finally, our integer programming model with linear constraints becomes:

$$\max \sum_{j=1}^{10} \mathrm{p}_j x_j$$

$$\sum_{j=1}^{10} \mathrm{c}_j x_j \le \mathrm{q}$$

$$x_3 + x_4 \le 1 \qquad \text{(i)}$$

$$x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8 + x_9 + x_{10} = 3 \qquad \text{(ii)}$$

$$x_2 \le x_1 \qquad \text{(iii)}$$

$$x_1 \le 1 - x_3 \qquad \text{(iv)}$$

$$x_1 + x_2 = 1 \qquad \text{(v)}$$

$$x_1 - x_2 = 0 \qquad \text{(vi)}$$

Project Selection

Model cont'd.

$$\left.\begin{aligned} x_1 + x_2 + x_7 + x_8 + x_9 + x_{10} &\geq 2 \\ x_1 + x_2 + x_7 + x_8 + x_9 + x_{10} &\leq 4 \end{aligned}\right\} \quad \text{(vii)}$$

$$\left.\begin{aligned} x_5 &\leq x_3 + x_4 \\ x_6 &\leq x_3 + x_4 \end{aligned}\right\} \quad \text{(viii)}$$

$$\left.\begin{aligned} x_1 + x_2 + x_3 &\geq 2(1 - \mathrm{y}) \\ x_4 + x_5 + x_6 + x_7 + x_8 &\leq 5 - 2y \end{aligned}\right\} \quad \text{(ix)}$$

$$\left.\begin{aligned} x_1 + x_2 &\leq 1 + z \\ x_9 + x_{10} &\geq z \end{aligned}\right\} \quad \text{(x)}$$

$$x_1, x_2, \ldots, x_{10} \in \{0,1\}$$

$$\mathrm{y}, \mathrm{z} \in \{0,1\}$$

Modeling Fixed Costs

- A set of potential warehouses: 1, . . . ,m
- A set of clients : 1, . . . , n

- If warehouse *i* is used, then there is a fixed cost of $\mathbf{f_i}$, i =1,… ,m
- $\mathbf{c_{ij}}$: unit cost of transportation from warehouse *i* to client *j*,

  $i = 1,\ldots,m$; $j = 1,\ldots,n$

- $\mathbf{a_i}$: capacity of warehouse *i* in terms of # of units, $i = 1,\ldots,m$
- $\mathbf{d_j}$: # of units demanded by client *j*, j = 1,… ,n

Note: More than one warehouse can supply the client.

Modeling Fixed Costs

How to formulate the mathematical model in order to minimize total fixed costs and transportation costs while meeting the demand and capacity constraints?

Decision Variables:

IP Model:

The Set Covering Problem

You wish to decide where to locate hospitals in a city. Suppose that there are 5 different towns in the city.

Travel Time

Potential Location \ Town	1	2	3	4	5
1	10	15	20	10	16
2	15	10	8	14	12
3	8	6	12	15	10
4	12	17	7	9	10

- Travel time between each town and the nearest hospital should not be more than 10 minutes. What is the minimum number of hospitals?

The Set Covering Problem

Decision Variables:

IP Model:

Location Problems (Cover, Center, Median)

Ankara municipality is providing service to 6 regions. The travel times (in minutes) between these regions, say t_{ij} values are:

	R1	R2	R3	R4	R5	R6
R1	0	10	20	30	30	20
R2	10	0	25	35	20	10
R3	20	25	0	15	30	20
R4	30	35	15	0	15	25
R5	30	20	30	15	0	14
R6	20	10	20	25	14	0

Location Problems

(a) Cover

- The municipality wants to build fire stations in some of these regions. Find the minimum number of fire stations required if the municipality wants to ensure that there is a fire station within 15-minute drive time from each region. (Similar to ??)
- **Decision Variables:**

- **IP Model:**

Location Problems

(b) Median

The municipality now wants to build 2 post offices. There is going to be a back and forth trip each day from each post office to each one of the regions assigned to this post office. Formulate a model which will minimize the total trip time.

Decision Varriables:

IP Model:

Location Problems

c) What if additional to part (b), you have the restriction that a post office cannot provide service to more than 4 regions including itself?

Add the following constraint to the above model:

Location Problem

d) Center:

The municipality wants to build 2 fire stations so that the time to reach the farthest region is kept at minimum.

Decision Variables:

IP Model:

Location Problem

e) Consider the median problem in part (b). Assume that in addition to the back-and-forth trips between the post offices and the assigned regions, there will be a trip between the two post offices. The municipality wants to locate the offices in such a way that the total trip time is minimized.

Decision Variables:

IP Model:

Traveling Salesman Problem (TSP)

- A salesperson lives in city 1. He has to visit each of the cities $2, \ldots, n$ exactly once and return home.
- Let $\mathbf{c}_{ij}$ be the travel time from city i to city j, $i = 1, \ldots, n$;

$j = 1, \ldots, n$.

- What is the order in which she should make her tour to

finish as quickly as possible?

What should be the decision variables here?

Traveling Salesman Problem (TSP)

- A salesperson lives in city 1. He has to visit each of the cities 2,…,n exactly once and return home.
- Let $\mathbf{c_{ij}}$ be the travel time from city i to city j , $i = 1, . . . , n;$

$j = 1, . . . , n.$

- What is the order in which she should make her tour to

finish as quickly as possible?

What should be the decision variables here?

$$x_{ij} = \begin{cases} 1, & \text{if he goes directly from city } i \text{ to city } j \\ 0, & \text{otherwise} \end{cases}$$

$$i = 1, 2, \ldots, n \; ; \; j = 1, 2, \ldots, n$$

Traveling Salesman Problem (TSP)

Since each city has to be visited only once, the formulation is:

$$\min \quad \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}x_{ij}$$

$$s.t.$$

$$\sum_{j=1}^{n} x_{ij} = 1, \qquad i = 1,\ldots,n$$

$$\sum_{i=1}^{n} x_{ij} = 1, \qquad j = 1,\ldots,n$$

$$x_{ii} = 0, \qquad i = 1,\ldots,n$$

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

Traveling Salesman Problem (TSP)

- With this formulation, a feasible solution is:

$$x_{12} = x_{23} = x_{31} = 1,$$
$$x_{45} = x_{54} = 1.$$

- But this is a "subtour"! How can we eliminate "subtours"?

- We need to impose the constraints:

$$\sum_{i \in S} \sum_{j \notin S} x_{ij} \geq 1 \qquad \forall \mathrm{S} \subseteq \mathrm{N},\ \mathrm{S} \neq \varnothing$$

Traveling Salesman Problem (TSP)

- For example, for n=5 and S={1,2,3} this constraint implies:

$$x_{14} + x_{15} + x_{24} + x_{25} + x_{34} + x_{35} \geq 1$$

- The difficulty is that there are a total of $2^n - 1$ such constraints. And that is a lot!
  - For n = 100, $2^n - 1 \approx 1.27 \times 10^{30}$.
- How about the total number of valid tours? That is $(n - 1)!$
  - For n = 100, $(n - 1)! \approx 9.33 \times 10^{155}$.
- TSP is one of the hardest problems. But several efficient algorithms exist that can find "good" solutions.

Class Exercises (Blending Problem)

A company produces animal feed mixture. The mix contains 2 active ingredients and a filler material. 1 kg of feed mix must contain a minimum quantity of each of the following 4 nutrients:

Nutrients	A	B	C	D
Required amount (in grams) in 1 kg of mix	90	50	20	2

The ingredients have the following nutrient values and costs:

	A	B	C	D	Cost/kg
Ingredient1 (gram/kg)	100	80	40	10	$40
Ingredient2 (gram/kg)	200	150	20	N/A	$60

Class Exercises (Blending Problem)

a) What should be the amount of active ingredients and filler material in 1 kg of feed mix in order to satisfy the nutrient requirements at minimum total cost?

Decision Variables:

IP Model:

Class Exercises (Blending Problem)

b) Suppose now that we have the following additional limitations

i. If we use any of ingredient 2, we incur a fixed cost of \$15 (in addition to the \$60 unit cost).

ii. It is enough to only satisfy 3 of the nutrition requirements rather than 4.

Decision Variables:

IP Model:

Some Comments on Integer Programming :

- There are often multiple ways of modeling the same integer program.
- Solvers for integer programs are extremely sensitive to the formulation (not true for LPs).
- Not as easy to model.
- Not as easy to solve.

Next Week

- Branch-and-Bound