IE 400 Principles of Engineering Management

Integer Programming: Modeling Examples

Lecture Notes 6

Integer Programs

Integer programs: 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 constraints:

$$0 \le x_j \le 1 \text{ and } x_j \text{ integer.}$$

A 2-Variable Integer program

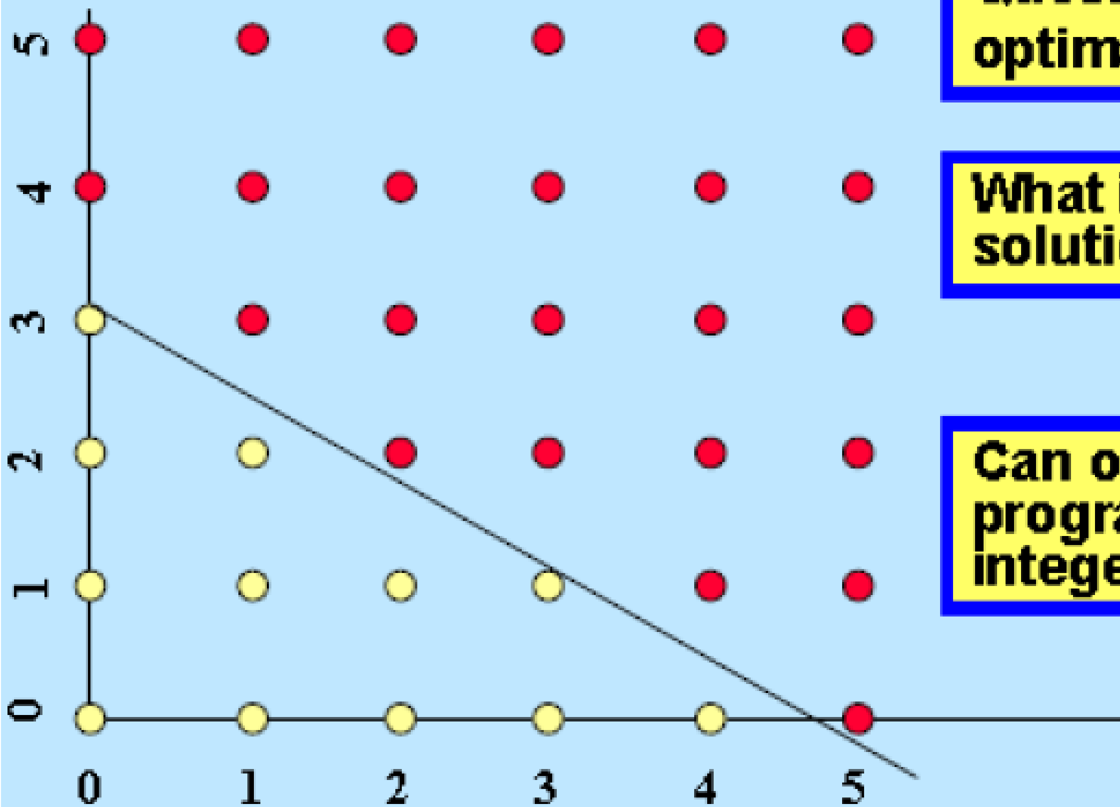
maximize $3x + 4y$

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

$x, y \ge 0$ and integer

- What is the optimal solution?

The Feasible Region

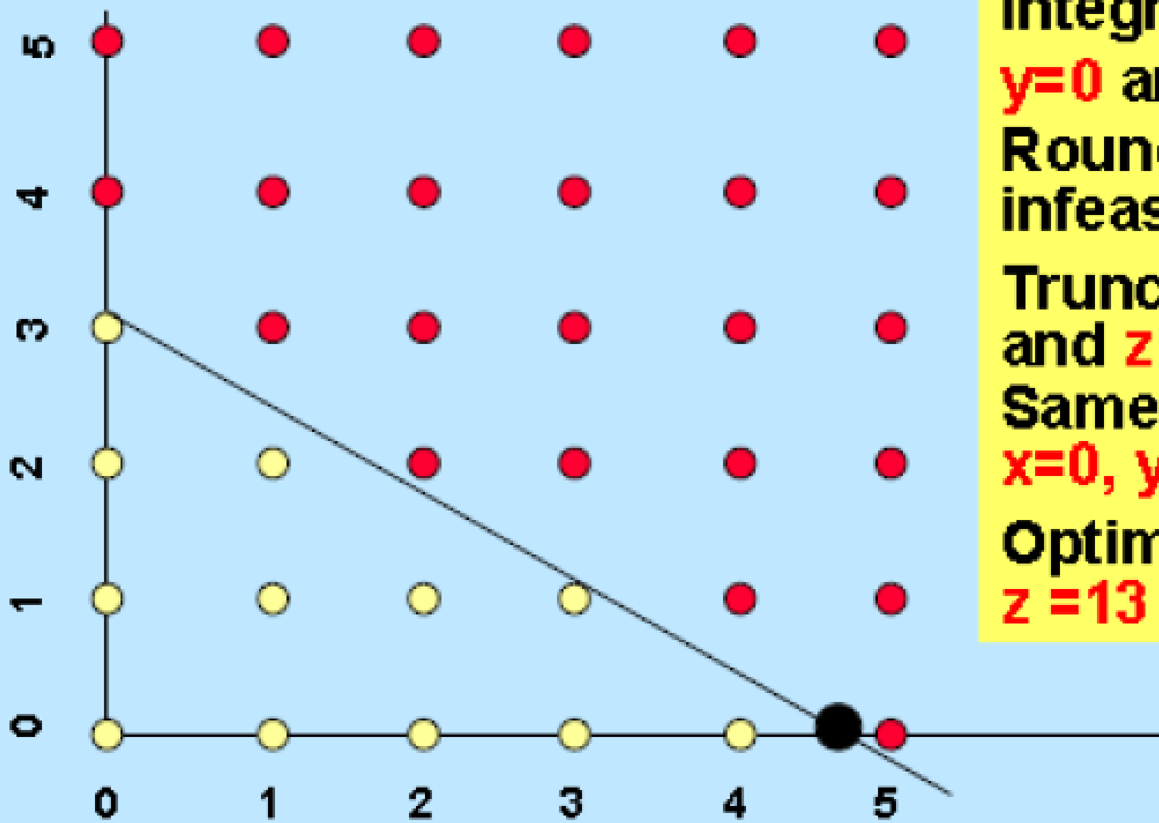


Question: What is the optimal integer solution?

What is the optimal linear solution?

Can one use linear programming to solve the integer program?

A rounding technique that sometimes is useful, and sometimes not.



Solve LP (ignore integrality) get $x=24/5$, $y=0$ and $z=14\ 2/5$.

Round, get $x=5$, $y=0$, infeasible!

Truncate, get $x=4$, $y=0$, and $z=12$

Same solution value at $x=0$, $y=3$.

Optimal is $x=3$, $y=1$, and $z=13$

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

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 in which we will need to either relax or strengthen these assumptions.

For example, consider the following bus scheduling problem:

Integer Programming :

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 receive 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.

Integer Programming :

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

or

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

Integer Programming :

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:

Integer Programming :

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:

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

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

$$x_1 + x_6 \geq 4$$

$$x_1 + x_2 \geq 8$$

$$x_2 + x_3 \geq 10$$

$$x_3 + x_4 \geq 7$$

$$x_4 + x_5 \geq 12$$

$$x_5 + x_6 \geq 4$$

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

Integer Programming :

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.

We solve IP problems utilizing methods that build on the simplex method.

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, . . . , 7 (1: Mon, 7: Sun)

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 not working on day i,
$i = 1, \dots, 7$ (1: Mon, 7: Sun)

i.e) How do you write the Monday work force requirement constraint?

Work Scheduling:

The problem is formulated as follows:

$$\min\ x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7$$

$$s.t.$$

$$\begin{array}{llll}
\quad\; x_2 + x_3 + x_4 + x_5 + x_6 + x_7 & \geq & 8 & \text{(Monday constraint)} \\
x_1 \quad\;\; + x_3 + x_4 + x_5 + x_6 + x_7 & \geq & 6 & \text{(Tuesday constraint)} \\
x_1 + x_2 \quad\;\; + x_4 + x_5 + x_6 + x_7 & \geq & 7 & \text{(Wednesday constraint)} \\
x_1 + x_2 + x_3 \quad\;\; + x_5 + x_6 + x_7 & \geq & 6 & \text{(Thursday constraint)} \\
x_1 + x_2 + x_3 + x_4 \quad\;\; + x_6 + x_7 & \geq & 9 & \text{(Friday constraint)} \\
x_1 + x_2 + x_3 + x_4 + x_5 \quad\;\; + x_7 & \geq & 11 & \text{(Saturday constraint)} \\
x_1 + x_2 + x_3 + x_4 + x_5 + x_6 & \geq & 9 & \text{(Sunday constraint)}
\end{array}$$

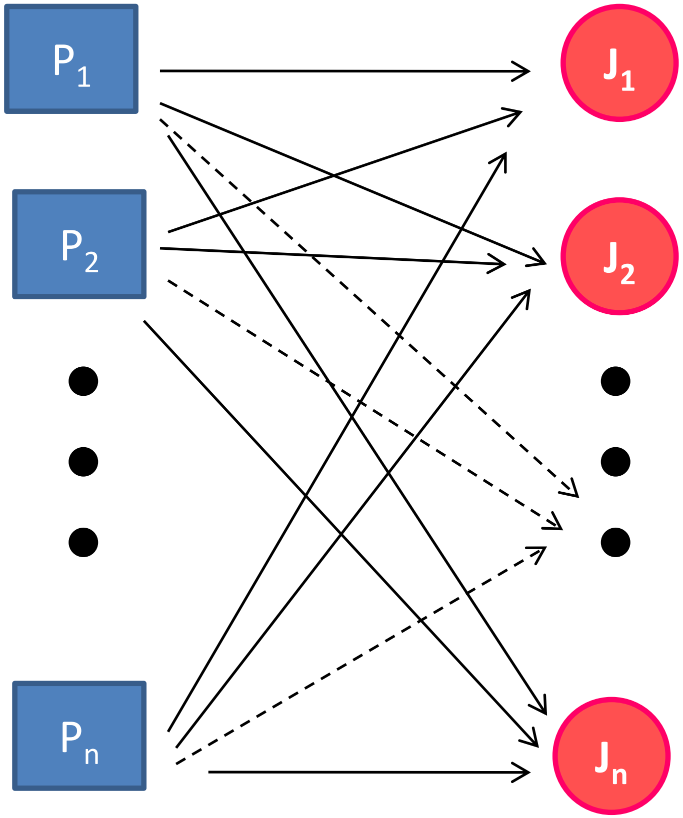
$$x_i \geq 0,\ i = 1, \ldots 7$$

$$x_i \text{ integer, i} = 1, \ldots 7.$$

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} .

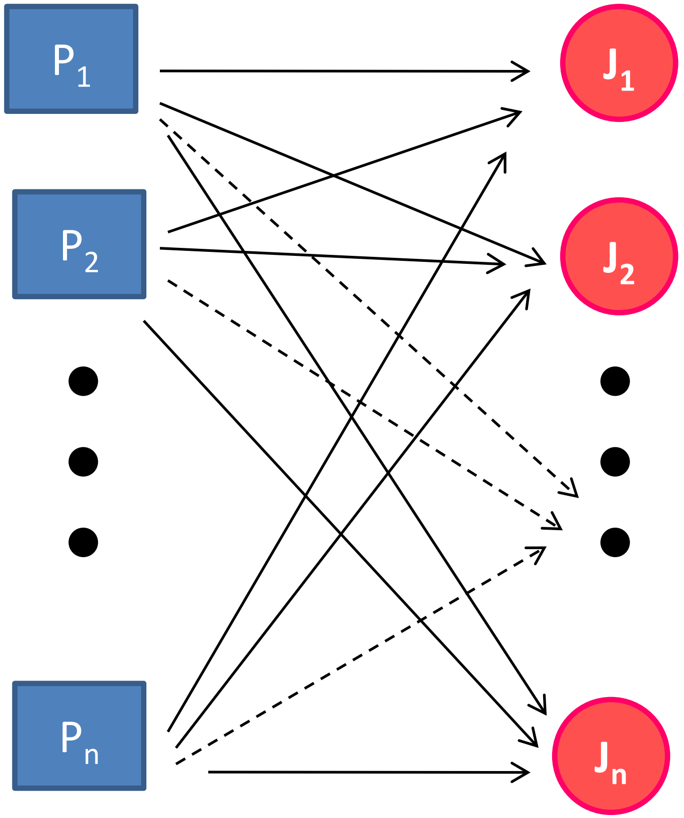
Find an assignment of n people to n jobs with minimum total cost.



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:

The Assignment Problem :

The problem is formulated as follows:

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

$$x_i \in \{0,1\}, \quad i = 1,\ldots,n; \quad j = 1,\ldots,n$$

The Assignment Problem :

The problem is formulated as follows:

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

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

$$x_i \in \{0,1\}, \quad i = 1,\ldots,n; \quad j = 1,\ldots,n$$

The Assignment Problem :

The problem is formulated as follows:

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

$s.t.$

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

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

$$x_i \in \{0,1\}, \quad i = 1,\ldots,n; \quad j = 1,\ldots,n$$

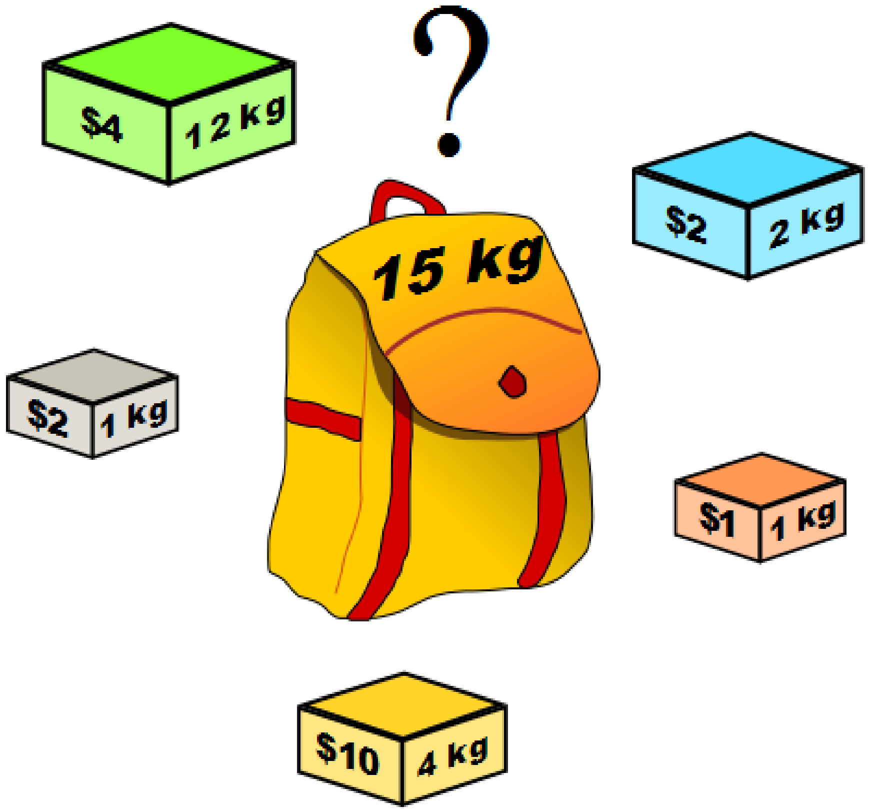
This is still an example of an IP problem since x_{ij} can only take one of the two integer values 0 and 1.

The 0-1 Knapsack Problem

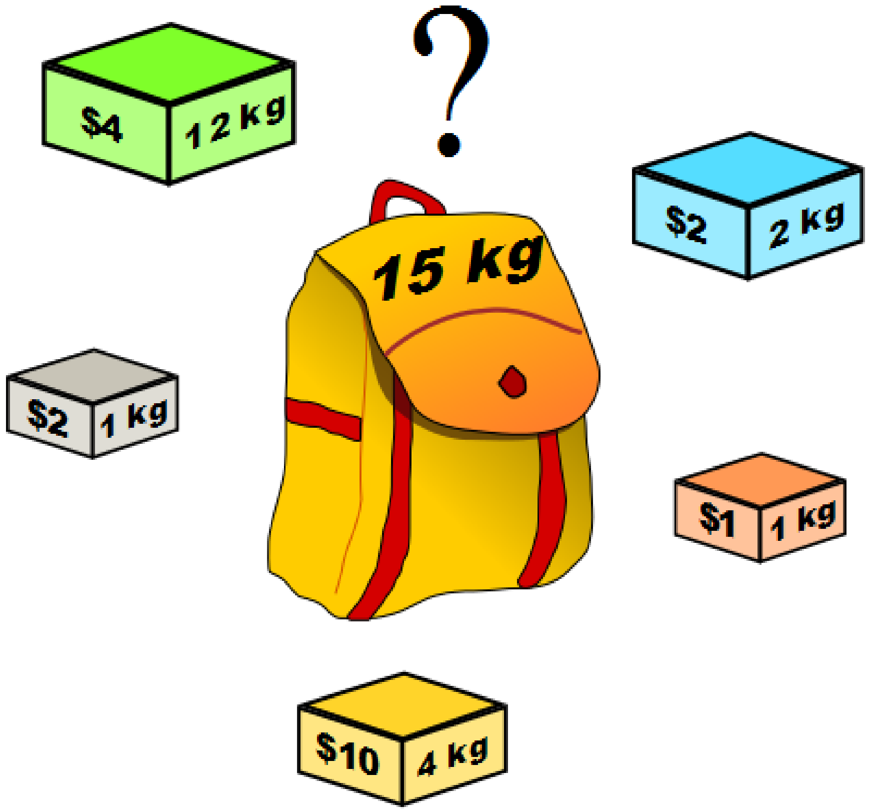
You are going on a trip. Your **Suitcase has a capacity of b kg's**.

- 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?



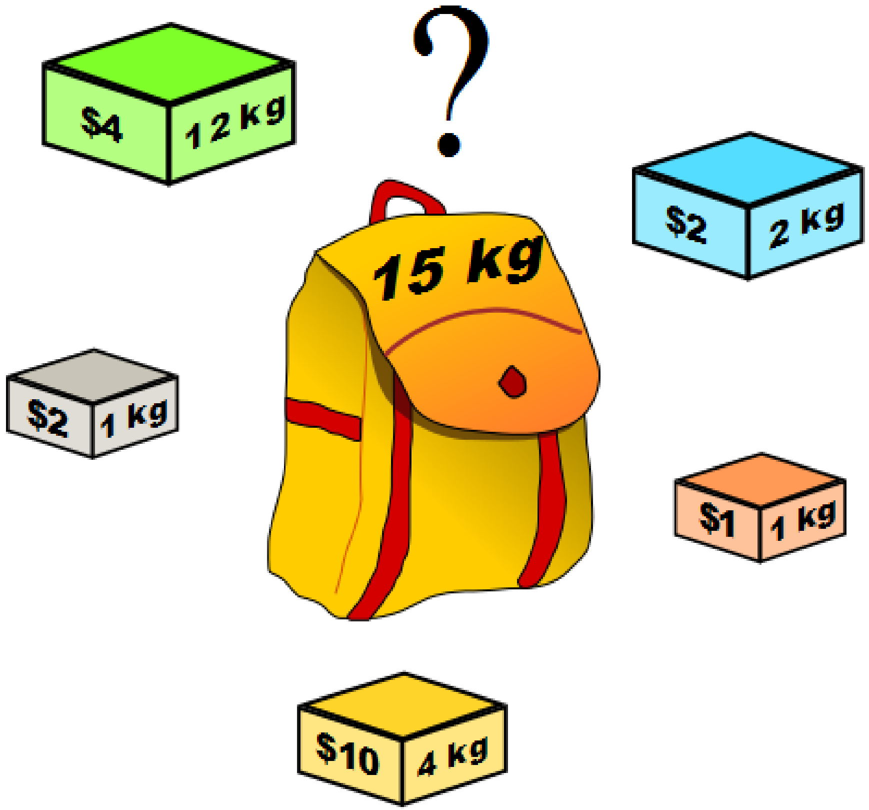
The 0-1 Knapsack Problem



The 0-1 Knapsack Problem

$$x_i = \begin{cases} 1, & \text{if item } i \text{ is packed in the suitcase} \\ 0, & \text{otherwise} \end{cases}$$

$$i = 1, \ldots, n$$



The 0-1 Knapsack Problem

$$x_i = \begin{cases} 1, & \text{if item } i \text{ is packed in the suitcase} \\ 0, & \text{otherwise} \end{cases}$$

$$i = 1, \ldots, n$$

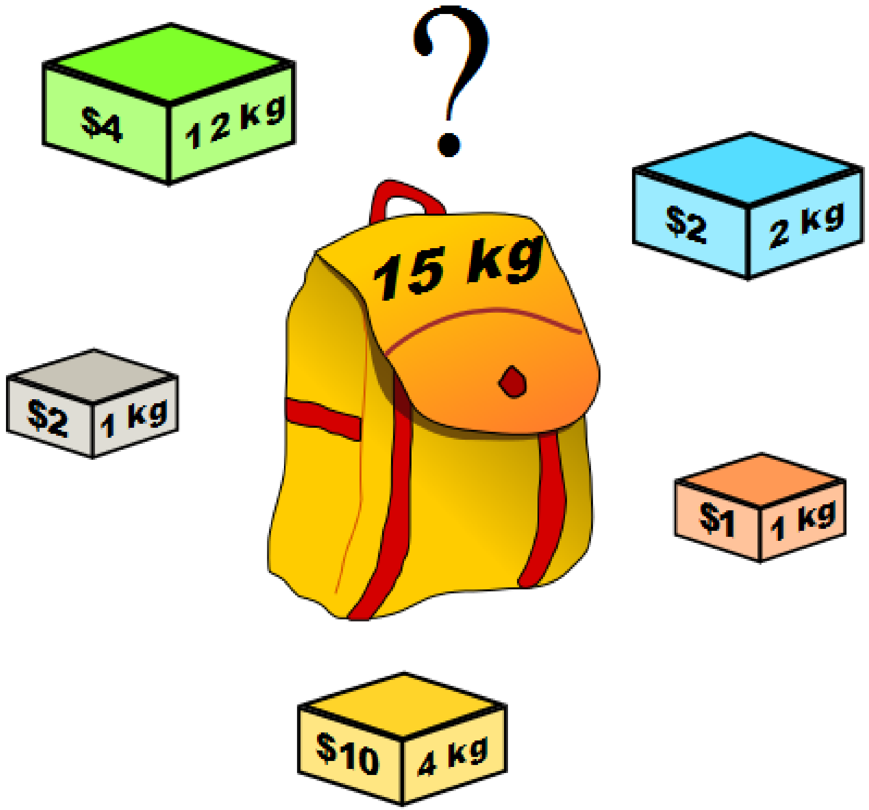
So the model formulation is:

$$\max \sum_{i=1}^{n} c_i x_i$$

$s.t.$

$$\sum_{i=1}^{n} a_i x_i \le b$$

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



This is also an example of an IP problem since x_i can only take one of the two integer values 0 and 1.

The Set Covering Problem

You wish to decide where to install hospitals in the 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

1. What should be an appropriate decision variable?

The Set Covering Problem

1. What should be an appropriate decision variable?

$$x_i = \begin{cases} 1, & \text{if location } i \text{ is selected} \\ 0, & \text{otherwise} \end{cases}$$

$$i = 1,2,3,4.$$

The Set Covering Problem

1. What should be an appropriate decision variable?

$$x_i = \begin{cases} 1, & \text{if location } i \text{ is selected} \\ 0, & \text{otherwise} \end{cases}$$

$$i = 1,2,3,4.$$

2. What should be the objective function?

The Set Covering Problem

1. What should be an appropriate decision variable?

$$x_i = \begin{cases} 1, & \text{if location } i \text{ is selected} \\ 0, & \text{otherwise} \end{cases}$$

$$i = 1,2,3,4.$$

2. What should be the objective function?

$$\text{Min} \;\; x_1 + x_2 + x_3 + x_4$$

The Set Covering Problem

3. What about the constraints?

   We need to make sure there is at least one accessible hospital to each town, which is defined by the statement that the nearest hospital <u>should not be more than 10 minutes away</u>. Looking at the table in the previous slide, we should identify each potential location that could offer an accessible service to each town.

The Set Covering Problem

You wish to decide where to install hospitals in the 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.

The Set Covering Problem

3. What about the constraints?

   We need to make sure there is at least one accessible hospital to each town, which is defined by the statement that the nearest hospital should not be more than 10 minutes away. Looking at the table in the previous slide, we should identify each potential location that could offer an accessible service to each town.

For example, for Town 1, only locations 1 and 3 satisfy the requirement. And at least, one of them should be selected. How do we define this constraint?

$$x_1+x_3 \geq 1$$

The Set Covering Problem

The mathematical formulation:

$$\min \quad x_1 + x_2 + x_3 + x_4$$

$$s.t.$$

$$\begin{array}{lllll} x_1 & & +\, x_3 & & \geq 1 \quad \text{(Town 1)} \\ & x_2 & +\, x_3 & & \geq 1 \quad \text{(Town 2)} \\ & x_2 & & +\, x_4 & \geq 1 \quad \text{(Town 3)} \\ x_1 & & & +\, x_4 & \geq 1 \quad \text{(Town 4)} \\ & & x_3 & +\, x_4 & \geq 1 \quad \text{(Town 5)} \end{array}$$

$$x_i \in \{0,1\}, \;\; i = 1,2,3,4.$$

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,

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

- $\mathbf{a}_i$: capacity of warehouse i in terms of # of units, $i = 1,\dots,m$
- $\mathbf{d}_j$: # of units demanded by client j, $j = 1,\dots,n$

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?

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?

$$y_i = \begin{cases} 1, & \text{if warehouse } i \text{ is used} \\ 0, & \text{otherwise} \end{cases}$$

$$i = 1, \ldots, m$$

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?

$$y_i = \begin{cases} 1, & \text{if warehouse } i \text{ is used} \\ 0, & \text{otherwise} \end{cases}$$

$$i = 1, \ldots, m$$

$$x_{ij} : \#\text{ of units shipped from warehouse } i \text{ to client } j,$$

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

Modeling Fixed Costs

The problem formulation is as follows:

$$s.t.$$

$$\sum_{i=1}^{m} x_{ij} = d_j, \qquad j = 1,\dots,n \text{ (Demand)}$$

$$x_{ij} \geq 0 \quad \text{(integer)}, \; i = 1,\dots,m; \; j = 1,\dots,n$$

Modeling Fixed Costs

The problem formulation is as follows:

$$s.t.$$

$$\sum_{\mathrm{i}=1}^{\mathrm{m}} x_{ij} = \mathrm{d}_{\mathrm{j}}, \qquad j = 1,\ldots,n \text{ (Demand)}$$

$$\sum_{\mathrm{j}=1}^{\mathrm{n}} x_{ij} \leq \mathrm{y}_i \mathrm{a}_i, \qquad i = 1,\ldots,m \text{ (Capacity)}$$

$$\mathrm{y}_{\mathrm{i}} \in \{0,1\}, \quad i = 1,\ldots,m$$

$$x_{ij} \geq 0 \ \text{ (integer)}, \ i = 1,\ldots,m; \ j = 1,\ldots,n$$

Modeling Fixed Costs

The problem formulation is as follows:

$$\min \sum_{i=1}^{m} \mathrm{f}_i y_i + \sum_{i=1}^{m}\sum_{j=1}^{n} \mathrm{c}_{ij} x_{ij}$$

$s.t.$

$$\sum_{\mathrm{i}=1}^{\mathrm{m}} x_{ij} = \mathrm{d}_{\mathrm{j}}, \qquad j = 1,\ldots,n \text{ (Demand)}$$

$$\sum_{\mathrm{j}=1}^{\mathrm{n}} x_{ij} \le \mathrm{y}_i \mathrm{a}_i, \qquad i = 1,\ldots,m \text{ (Capacity)}$$

$$\mathrm{y}_{\mathrm{i}} \in \{0,1\}, \quad i = 1,\ldots,m$$

$$x_{ij} \ge 0 \quad \text{(integer)}, \; i = 1,\ldots,m; \; j = 1,\ldots,n$$

Either-Or Constraints

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

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

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

Either-Or Constraints

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

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

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

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,

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

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

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>,

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

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

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

Either-Or Constraints

M is a number chosen **large enough to ensure that** *$f(x_1, x_2, \ldots, x_n) \le M$ and $g(x_1, x_2, \ldots, x_n) \le M$ 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

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

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

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

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

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

and

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

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

Either-Or Constraints

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

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

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

Either-Or Constraints

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

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

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

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.

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

.

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

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

while if a constraint $f(x_1,\ldots,x_n)>0$ *is not satisfied,*
then $g(x_1,\ldots,x_n)\geq 0$ *may or may not be satisfied.*

If-Then Constraints

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

To ensure this, we include the following set of constraints in the formulation:

$$
\begin{aligned}
&f\left(x_1, x_2, \ldots. x_n\right) \leq M\left(1-y\right) && \left(\mathrm{I}'\right)\\
&\text{-}g\left(x_1, x_2, \ldots. x_n\right) \leq My && \left(\mathrm{II}'\right)\\
&\text{where } y \in \{0,1\}.
\end{aligned}
$$

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

Project Selection

A manager should choose from among 10 different projects:

$\mathbf{p}_j$: profit if we invest in project j , j = 1, . . . , 10

$\mathbf{c}_j$: 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. At least two and at most four projects should be chosen from the set {1, 2, 7, 8, 9, 10}.
iii. Either both projects 1 and 5 are chosen or neither of them is chosen.
iv. Either project 1 or project 2 is chosen, but not both.

Project Selection

v. At least two of the investments 1, 2, 3 are chosen, or at least two of the investments 4, 5, 6, 7, 8 are chosen.

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

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

Project Selection

v. At least two of the investments 1, 2, 3 are chosen, or at least two of the investments 4, 5, 6, 7, 8 are chosen.

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

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

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

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

Project Selection

Objective function is:

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

Project Selection

Objective function is:

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

First the budget constraint:

Project Selection

Objective function is:

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

First the budget constraint:

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

Project Selection

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:

Project Selection

Objective function is:

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

First the budget constraint:

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

i. Projects 3 and 4 cannot be chosen together:

$$x_3 + x_4 \le 1$$

Project Selection

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

Project Selection

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

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

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

Project Selection

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

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

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

iii. Either both projects 1 and 5 are chosen or neither of them is chosen:

Project Selection

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

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

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

iii. Either both projects 1 and 5 are chosen or neither of them is chosen:

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

Project Selection

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

Project Selection

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

$$x_1 + x_2 = 1$$

Project Selection

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

$$x_1 + x_2 = 1$$

v. At least two of the investments 1, 2, 3 are chosen, or at least two of the investments 4, 5, 6, 7, 8 are chosen:

Project Selection

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

$$x_1 + x_2 = 1$$

v. At least two of the investments 1, 2, 3 are chosen, or at least two of the investments 4, 5, 6, 7, 8 are chosen:

It is more difficult to express this requirement in mathematical terms!

Project Selection

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

$$x_1 + x_2 = 1$$

v. At least two of the investments 1, 2, 3 are chosen, or at least two of the investments 4, 5, 6, 7, 8 are chosen:

It is more difficult to express this requirement in mathematical terms!

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

Project Selection

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

$$x_1 + x_2 = 1$$

v. At least two of the investments 1, 2, 3 are chosen, or at least two of the investments 4, 5, 6, 7, 8 are chosen:

It is more difficult to express this requirement in mathematical terms!

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

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

Project Selection

Now, let us return to our problem:

v. At least two of the investments 1, 2, 3 are chosen, or at least two of the investments 4, 5, 6, 7, 8 are chosen:

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

Project Selection

Now, let us return to our problem:

v. At least two of the investments 1, 2, 3 are chosen, or at least two of the investments 4, 5, 6, 7, 8 are chosen:

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

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

$$2\text{-}\ (x_1 + x_2 + x_3) \leq 0$$

$$2\text{-}(x_4 + x_5 + x_6 + x_7 + x_8) \leq 0$$

Project Selection

Now, let us return to our problem:

v. At least two of the investments 1, 2, 3 are chosen, or at least two of the investments 4, 5, 6, 7, 8 are chosen:

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

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

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

$$\underbrace{2\text{-}\left(x_4 + x_5 + x_6 + x_7 + x_8\right)}_{\text{g(.)}} \leq 0$$

Project Selection

Now, let us return to our problem:

v. At least two of the investments 1, 2, 3 are chosen, or at least two of the investments 4, 5, 6, 7, 8 are chosen:

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

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

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

$$\underbrace{2\text{-}\left(x_4 + x_5 + x_6 + x_7 + x_8\right)}_{g(.)} \leq 0$$

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:

v. At least two of the investments 1, 2, 3 are chosen, or at least two of the investments 4, 5, 6, 7, 8 are chosen:

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

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

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

$$\underbrace{2\text{-}\left(x_4 + x_5 + x_6 + x_7 + x_8\right)}_{g(.)} \leq 0$$

So, M=2!

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

Project Selection

Now, let us return to our problem:

v. At least two of the investments 1, 2, 3 are chosen, or at least two of the investments 4, 5, 6, 7, 8 are chosen:

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

Hence by defining the binary variable *y*, the formulation becomes:

Project Selection

Now, let us return to our problem:

v. At least two of the investments 1, 2, 3 are chosen, or at least two of the investments 4, 5, 6, 7, 8 are chosen:

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

Hence by defining the binary variable *y*, the formulation becomes:

$$2\text{-}\left(x_1 + x_2 + x_3\right) \leq My_1,$$

$$2\text{-}\left(x_4 + x_5 + x_6 + x_7 + x_8\right) \leq M\left(1 - y_1\right),$$

$$y_1 \in \{0,1\}.$$

Project Selection

Now, let us return to our problem:

v. At least two of the investments 1, 2, 3 are chosen, or at least two of the investments 4, 5, 6, 7, 8 are chosen:

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

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

$$x_1 + x_2 + x_3 \geq 2(1 - y_1),$$
$$x_4 + x_5 + x_6 + x_7 + x_8 \geq 2y_1,$$
$$y_1 \in \{0,1\}.$$

Project Selection

vi. 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 $x_9 + x_{10} \geq 1$

Project Selection

vi. 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?

Project Selection

vi. 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:

Project Selection

vi. 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:

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

Project Selection

vi. 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:

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

To ensure thus, we need the constraints *f(.)≤M(1-y) and -g(.)≤My, where y=0 or 1.*

Project Selection

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

$$\underbrace{x_1 + x_2 - 1}_{f(.)} \le M\left(1 - y_2\right),$$

$$-\left(\underbrace{x_9 + x_{10} - 1}_{g(.)}\right) \le My_2,$$

$$\text{and } y_2 \in \{0,1\}.$$

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

M=1 satisfies this!

Project Selection

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

Putting the value of M=1 and rearranging the terms:

$$x_1 + x_2 \leq 2 - y_2,$$

$$x_9 + x_{10} \geq 1 - y_2,$$

$$\text{and } y_2 \in \{0,1\}.$$

The Cutting Stock Problem:

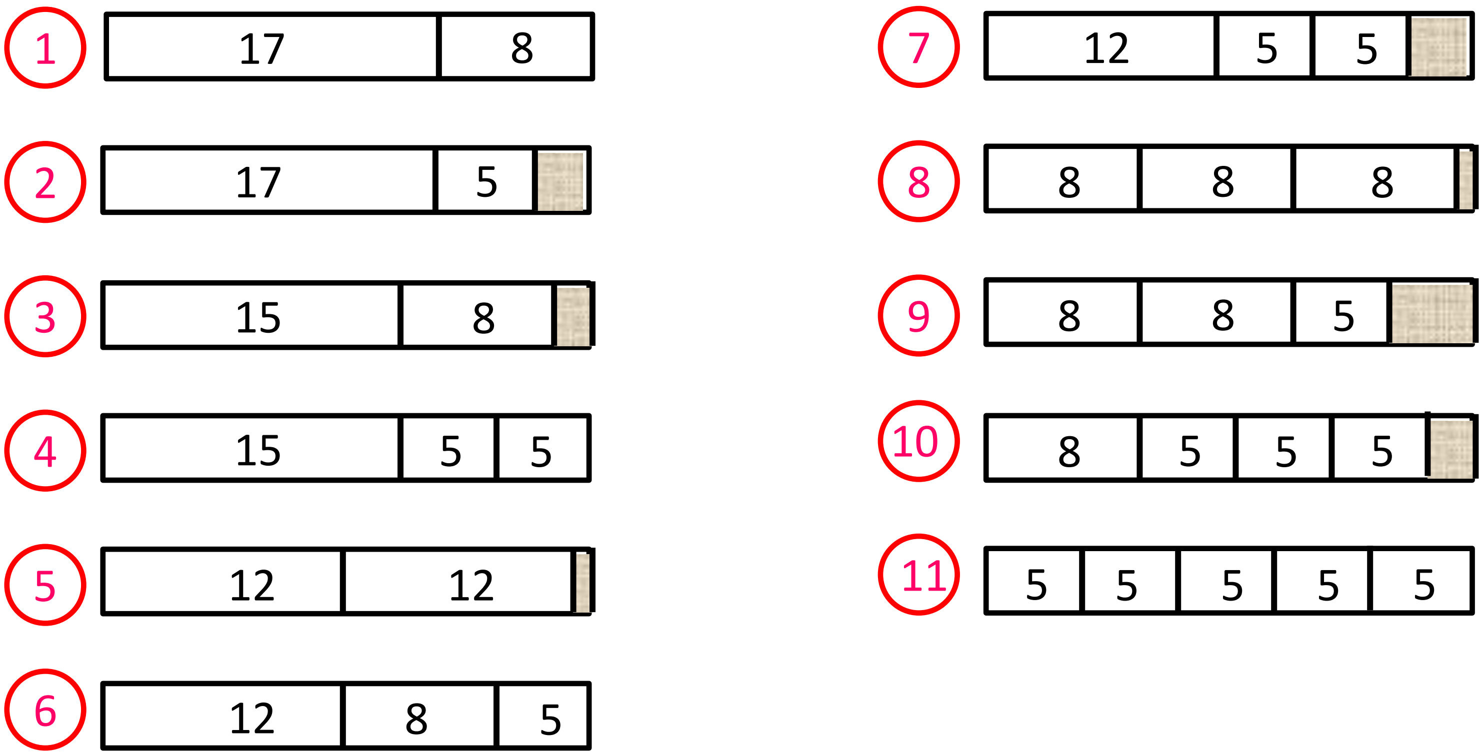
A paper company manufactures and sells rolls of paper of fixed width in 5 standard lengths: 5, 8, 12, 15 and 17 m. Demand for each size is given below.

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?

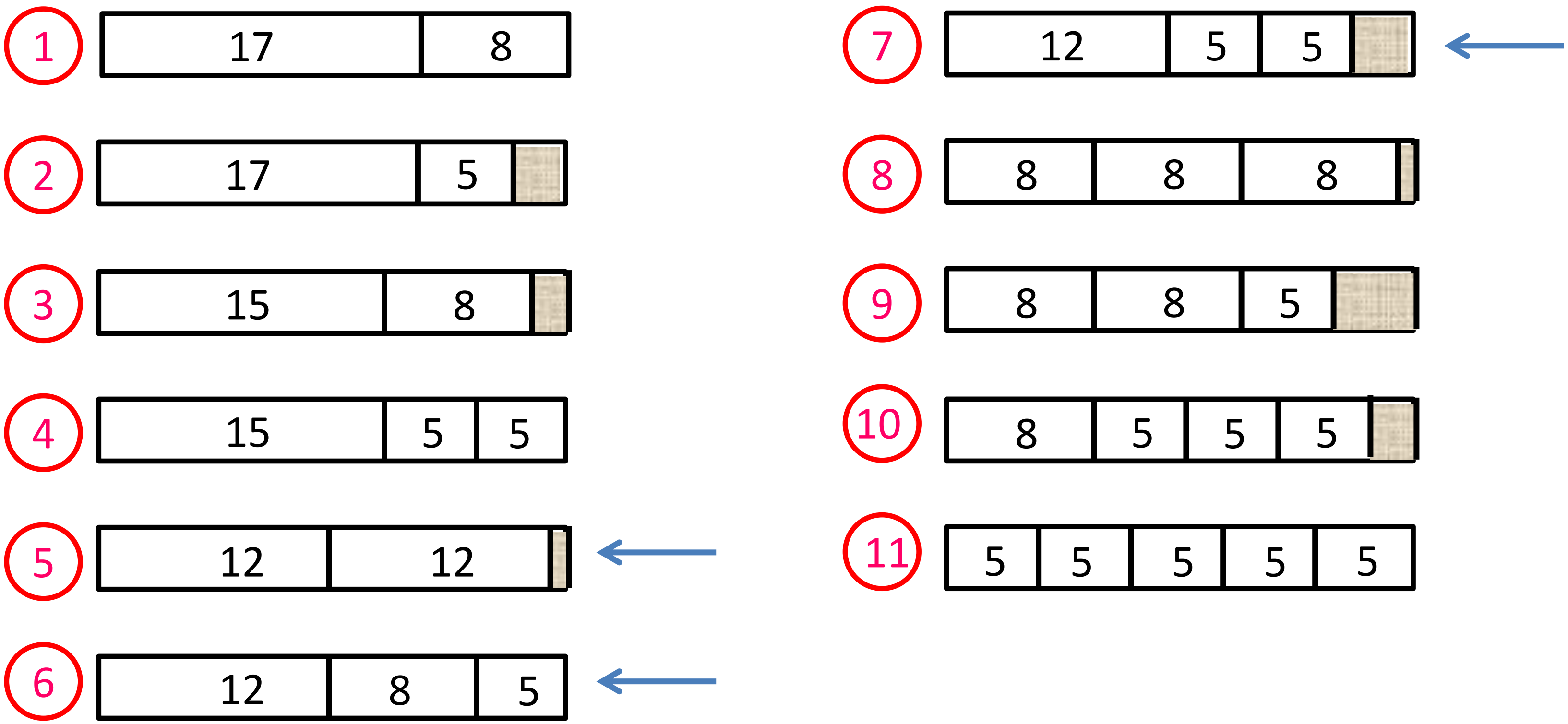
i.e. How do you write the constraint for the demand of 12 m rolls of paper?

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

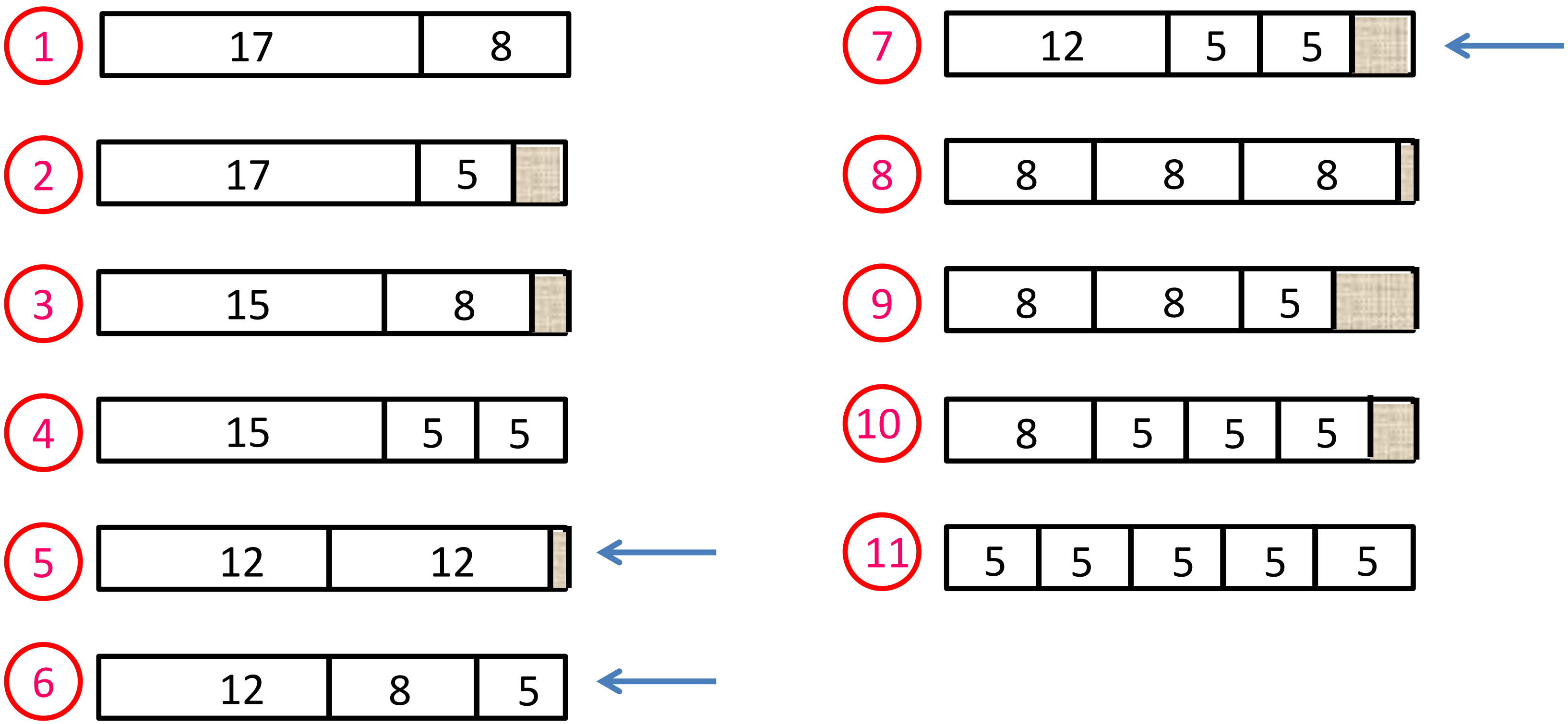
The Cutting Stock Problem:

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*.

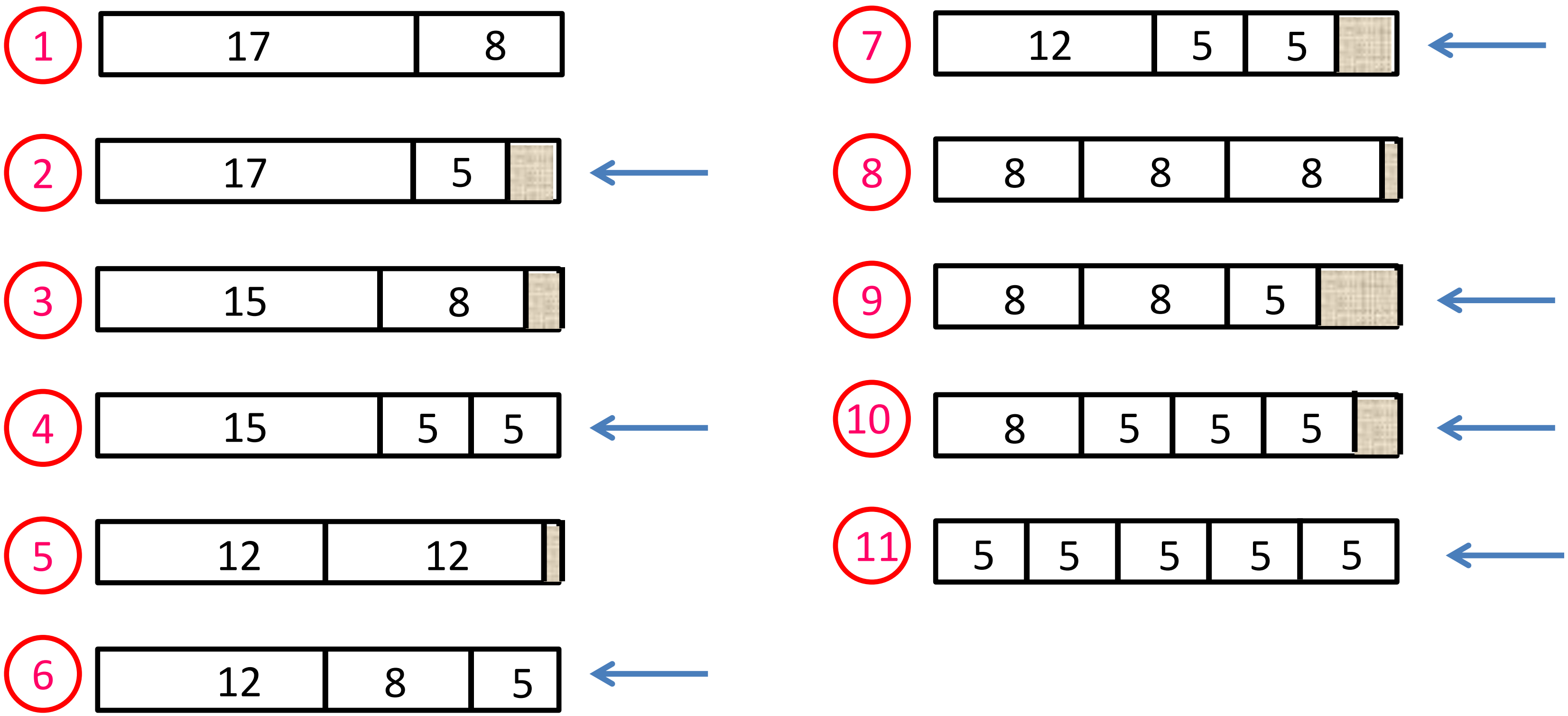
What about the constraints?

i.e. What about demand for 5 m?

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

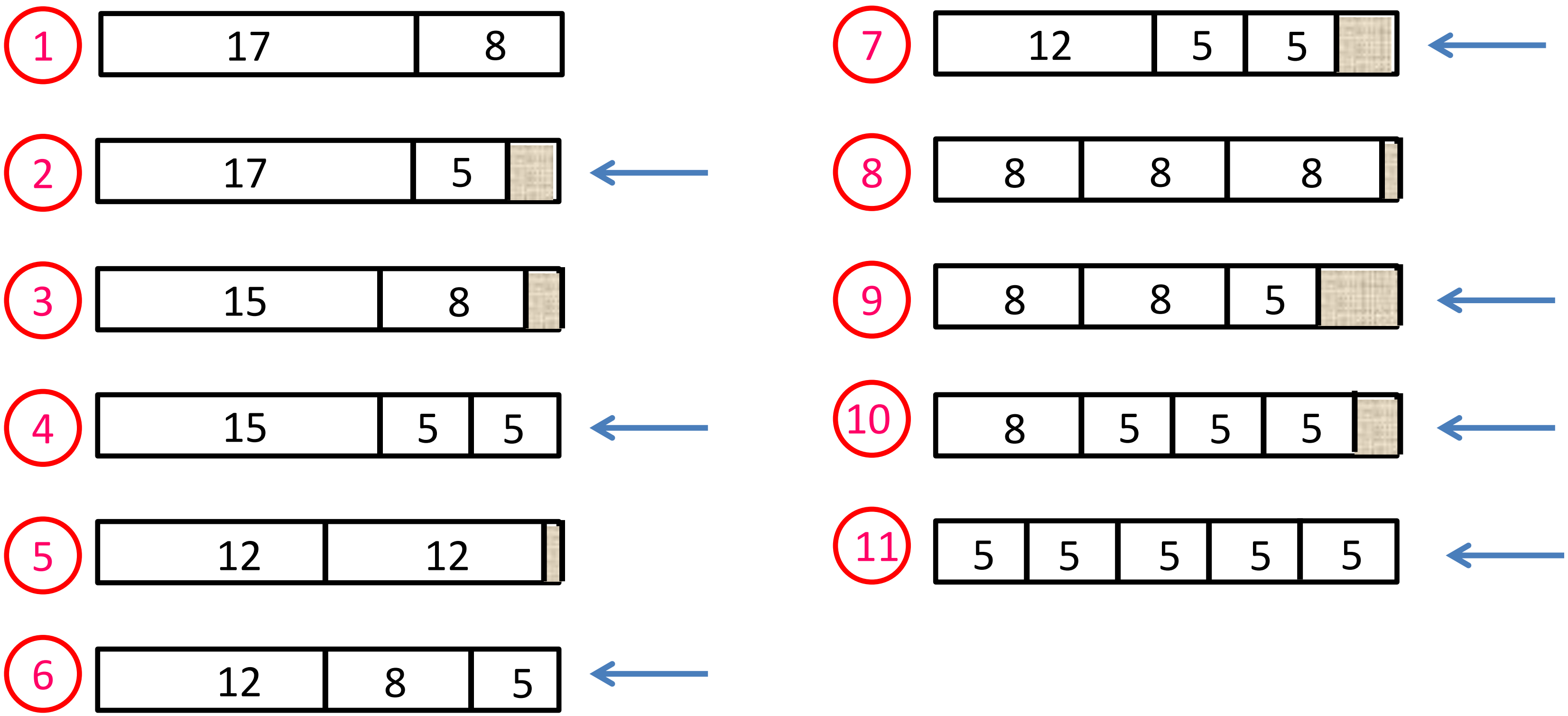
The Cutting Stock Problem:

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:

$$
\begin{aligned}
&s.t.\\
&x_2 + 2x_4 + x_6 + 2x_7 + x_9 + 3x_{10} + 5x_{11} \geq 40\\
&x_1 + x_3 + x_6 + 3x_8 + 2x_9 + x_{10} \geq 35\\
&2x_5 + x_6 + x_7 \geq 30\\
&x_3 + x_4 \geq 25\\
&x_1 + x_2 \geq 20\\
&x_j \geq 0,\ j = 1,\ldots,11,\ x_j \text{ integer.}
\end{aligned}
$$

The Cutting Stock Problem:

The problem is formulated as follows:

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

$$s.t.$$

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

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

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

$$x_3 + x_4 \ge 25$$

$$x_1 + x_2 \ge 20$$

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

Traveling Salesman Problem (TSP):

A sales person 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 variable here?

Traveling Salesman Problem (TSP):

A sales person lives in city 1. He has to visit each of the cities *2,…,n* exactly once and return home.

Let $\mathbf{c}_{\mathbf{ij}}$ 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 variable 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.

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.