



Inspire…Educate…Transform.

AI and Decision Sciences

Effective Decision Making: Linear Programming

Dr. Anand Narasimhamurthy

Outline

- A high level overview of applications of Linear Programming with use cases
- General overview of optimization
  - Linear Program : A specific type of optimization problem
- Understanding Linear Programming formulation with examples
- Lab exercise
- Multiple use cases

Advanced topics not covered today

- Duality
- The Simplex Method

Linear Programming : A few sample applications

Linear Programming finds applications in a variety of problem domains and disciplines such as :

- Operations research : Eg. Transportation problem, assignment problem
- Inventory and marketing management
- Applications in Finance : Eg. Portfolio Selection, Financial Planning, Financial Mix Strategy
- A general framework for variants of problems such as max flow, bipartite matching

Use case 1 : Supply chain/logistics

A clothing brand has decided to use its **retail stores in addition to its warehouses** for fulfilling online orders since retail stores are generally situated in convenient locations.
A batch of online orders available at the beginning of the workday are considered.

The following constraints need to be considered:

- Retail stores have limited inventory as compared to warehouses and would also need to keep a minimum stock of various items.

- Due to limited staff, each retail outlet has specified the maximum number of orders per day it can service.

Use case 1 : Supply chain/logistics

- Assume that the cost of servicing an order by a retail store is available and represented in the form of a matrix.
  - The cost takes into account the distance of the customer location from the store and other logistical considerations.

- **Problem :** For a given (fixed) batch of orders, determine the allocation of retail stores to orders (i.e. which retail store services which order) **so as to minimize the overall cost.**

Less obvious applications

- Many real world problems can be formulated as **bipartite matching** and **max flow problems.**

  Examples : Airline scheduling sub tasks, image segmentation

- Although efficient algorithms exist for standard versions of the bipartite matching and max flow problems, these algorithms cannot directly be used on variants of above problems.
  - These variants (as well as standard versions) can be formulated as Linear Programming Problems.

Optimization : A general overview

Common broad classes of optimization problems : Unconstrained optimization

Objective function eg. $f(x_1,...x_n)$ only → **Unconstrained optimization**

Examples

Data fitting problems
- Linear/polynomial regression
- Lasso, Ridge regression

Classification
- Logistic regression
- Neural networks

Solution techniques

- **Analytical solution using multivariate calculus :** Directly evaluate first and second order conditions

- **Iterative methods**
  Start with one or more initial solutions and converge to a better solution. Examples:
  - **Gradient descent,** Newton's method
  - Evolutionary approaches. Eg. Simulated annealing, Genetic Algorithms etc.

Common broad classes of optimization problems : Constrained optimization

Objective function $f(x_1,...x_n)$

+

additional constraints

Constrained optimization

Examples

Linear Program

Quadratic Program: Support Vector Machine

A few general solution techniques

- **Primal methods**
- **Penalty and barrier methods**
- **Dual and cutting plane methods**
- **Lagrangian methods**

Constrained optimization problem : General framework

Objective function minimize/maximize

$$\text{Optimize } z = f(x_1, x_2, \ldots, x_n)$$

$$\text{Subject to } g_1(x_1, x_2, \ldots, x_n) \le or \ge or = b_1$$

$$g_2(x_1, x_2, \ldots, x_n) \le or \ge or = b_2$$

$$g_3(x_1, x_2, \ldots, x_n) \le or \ge or = b_3$$

Constraints (Inequality and equality)

The optimization problem is generally specified using matrices and vectors.

Linear program : General description

Linear Program (LP) is a constrained optimization problem where :

- The objective function is a linear combination of design variables
  i.e. it is of the form $f^T x$ (= $f_1 x_1 + f_2 x_2 \dots + f_n x_n$)
- The constraints (both equality and inequality) are also linear.
  - Equality constraints may be expressed as $A_{eq}\, x = b_{eq}$
  - Inequality constraints may be expressed as $A_{ineq}\, x \le b_{ineq}$

where $x = \begin{bmatrix} x_1 \\ x_2 \\ . \\ . \\ x_n \end{bmatrix}$

and A_{ineq}, A_{eq}, b_{ineq}, b_{eq} are matrices and vectors of compatible sizes.

Two special classes of constrained optimization problems

	Objective function	Constraints
Linear Program	Linear	Linear equality & inequality
	$f^T x$	$A_{eq}x = b_{eq}$ $Ax \leq b$
Quadratic Program	Quadratic	Linear equality & inequality
	$x^T Hx + f^T x$	$A_{eq}x = b_{eq}$ $Ax \leq b$

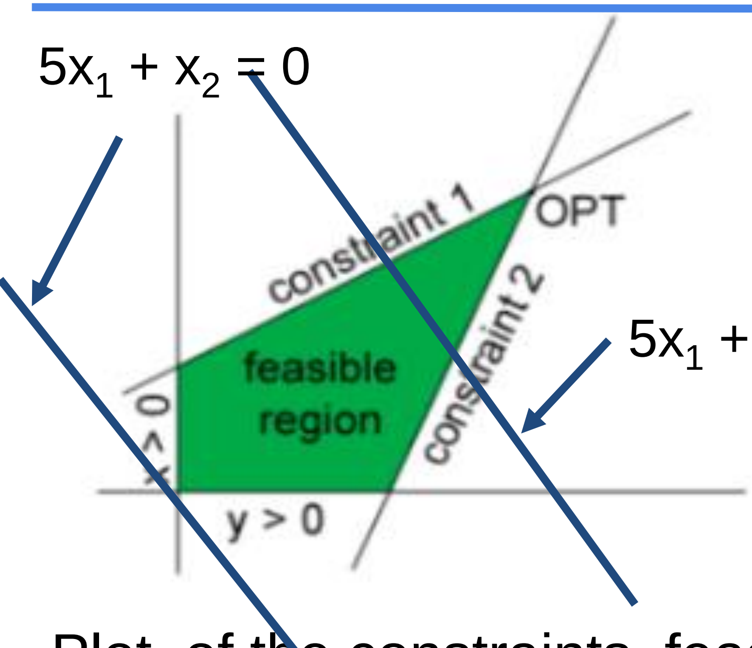
A simple example with two variables

Source : http://pages.cs.wisc.edu/~shuchi/courses/787-F09/scribe-notes/lec5.pdf

$$\begin{aligned} \text{maximize} \quad & 5x_1 + x_2 \\ \text{s.t.} \quad & -x_1 + 2x_2 \le 3 \\ & x_1 - x_2 \le 2 \\ & x_1, x_2 \ge 0 \end{aligned}$$

Formulation of the LP using matrix vector notation

$$\begin{aligned} \text{minimize} \quad & f^T x \quad \Leftrightarrow \text{maximize } (-f^T x) \\ \text{s.t.} \quad & Ax \le b \\ & lb \le x \end{aligned}$$



Plot of the constraints, feasible region + objective function

where

Objective function

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad f = \begin{bmatrix} -5 \\ -1 \end{bmatrix}$$

Constraints

$$A = \begin{bmatrix} -1 & 2 \\ 1 & -1 \end{bmatrix} \quad b = \begin{bmatrix} 3 \\ 2 \end{bmatrix} \quad lb = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Python code for the simple example

```
import numpy as np
from scipy.optimize import linprog
from numpy.linalg import solve

f = -1*np.array([5,1]); #Objective function
                           #maximize 5*x1 + x2 equivalent to minimize -5*x1 - x2

#Inequality constraints
A_ineq = np.array([[-1,2],[1,-1]]);
b_ineq = np.array([3,2]);

lb_ub = (0,None); #Bounds on design variables

res = linprog(f, A_ub=A_ineq, b_ub=b_ineq,bounds=lb_ub);
print('Value of objective function at optimal solution = ', -res.fun);
print(' Solution x = ', res.x);
```

How to search for the optimal solution?

The Fundamental Theorem of Linear Programming : If a Linear Program has a (bounded) optimal solution, then there exists an extremal point of the feasible region which is optimal.

Identify all the corner points of the feasible region and calculate the objective function at each of them.

The **corner point with the best objective function value** yields the optimal solution to the problem.

Solving Linear Programs

Two commonly used classes of techniques for solving Linear Programs

- **Simplex :**
  Algorithms using the Simplex method essentially start from some initial extreme point (an initial basic feasible solution), and follow a path along the edges of the feasible region towards an optimal extreme point, such that all the intermediate extreme points visited are improving (more accurately, not worsening) the objective function.

- **Interior point methods**
  - Interior point methods approach a solution from the interior or exterior of the feasible region
  - two important interior point algorithms: the barrier method and primal-dual IP method.

Solving Linear Programs

A few examples of software/tools for Linear Programming (and other optimization) problems

- Free : **R** (library lpsolve), **python** (scipy and numpy libraries)

- Proprietary :
  - **Matlab** : Optimization and other specialized toolboxes.
  - **AMPL** : Modelling language for large-scale linear, mixed integer and nonlinear optimization
  - **CPLEX** : Solves integer programming, very large linear programming problems and others.

Linear Program : Goals

- Recognize the optimization problem:
  - Establish the objective function
- Problem formulation
  - Identify parameters, decision variables, equality and inequality constraints
- Solve and analyze the resulting Linear Program
- Bipartite matching and max flow : A short review
- Problems covering use cases in multiple domains

Process of optimizing

- Identify and name the decision variables consistently
- Mathematically define the objective/fitness function
- Identify all requirements, restrictions and limitations
    - Express any hidden constraints (Non-negative or integer only constraints, e.g.: Price cannot be –ve, production values must be +ve integers)
- Solve the resulting Linear Program

Illustrative example 1

Blue Ridge Hot Tubs manufactures and sells two models of hot tubs: the ***Aqua-Spa*** *and the* ***Hydro-Lux****.*
Blue Ridge buys prefabricated fiberglass hot tub shells from a local supplier and adds the pump and tubing to the shells to create his hot tubs. (This supplier has the capacity to deliver as many hot tub shells as needed)

Blue Ridge installs the same type of pump into both hot tubs. They will have ***only 200 pumps available during their next production cycle****.*

Illustrative example 1

The main difference between the two models of hot tubs is the amount of tubing and labor required.

- *Each Aqua-Spa requires 9 hours of **labor** and 12 feet of **tubing**.*

- *Each Hydro-Lux requires 6 hours of **labor** and 16 feet of **tubing**.*

- *The firm expects to have 1,566 production labor hours and 2,880 feet of tubing available during the next production cycle.*

- *Each Aqua-Spa unit sold earns a **profit** of $350 and each Hydro-Lux sold earns a **profit** of $300.*

Historically, all the hot tubs are sold and no inventory needs to be carried forward.

Identify the decision variables

- How many Aqua-Spas and Hydro-Luxes should be produced?
  - Let X_1 and X_2 represent the number of Aqua-Spas and Hydro-Luxes respectively, to produce.

State the objective function as a linear combination of the decision variables

- The company earns a profit of \$350 on each Aqua-Spa (X_1) sold and \$300 on each Hydro-Lux (X_2) sold.
- The objective of maximizing the profit earned is stated mathematically as:

$$\text{MAX} \quad (350X_1 + 300X_2)$$

State the constraints as linear combinations of the decision variables.

- Only 200 pumps are available and each hot tub requires one pump

Constraint : $1X_1 + 1X_2 \leq 200$

- Only 1,566 labor hours available during the next production cycle.

- Each Aqua-Spa built (each unit of X_1) requires 9 labor hours and each Hydro-Lux (each unit of X_2) requires 6 labor hours

Constraint : $9 X_1 + 6 X_2 \leq 1{,}566$

- Each Aqua-Spa requires 12 feet of tubing, and each Hydro-Lux produced requires 16 feet of tubing

Constraint : $12X_1 + 16X_2 \leq 2{,}880$

Hidden constraints

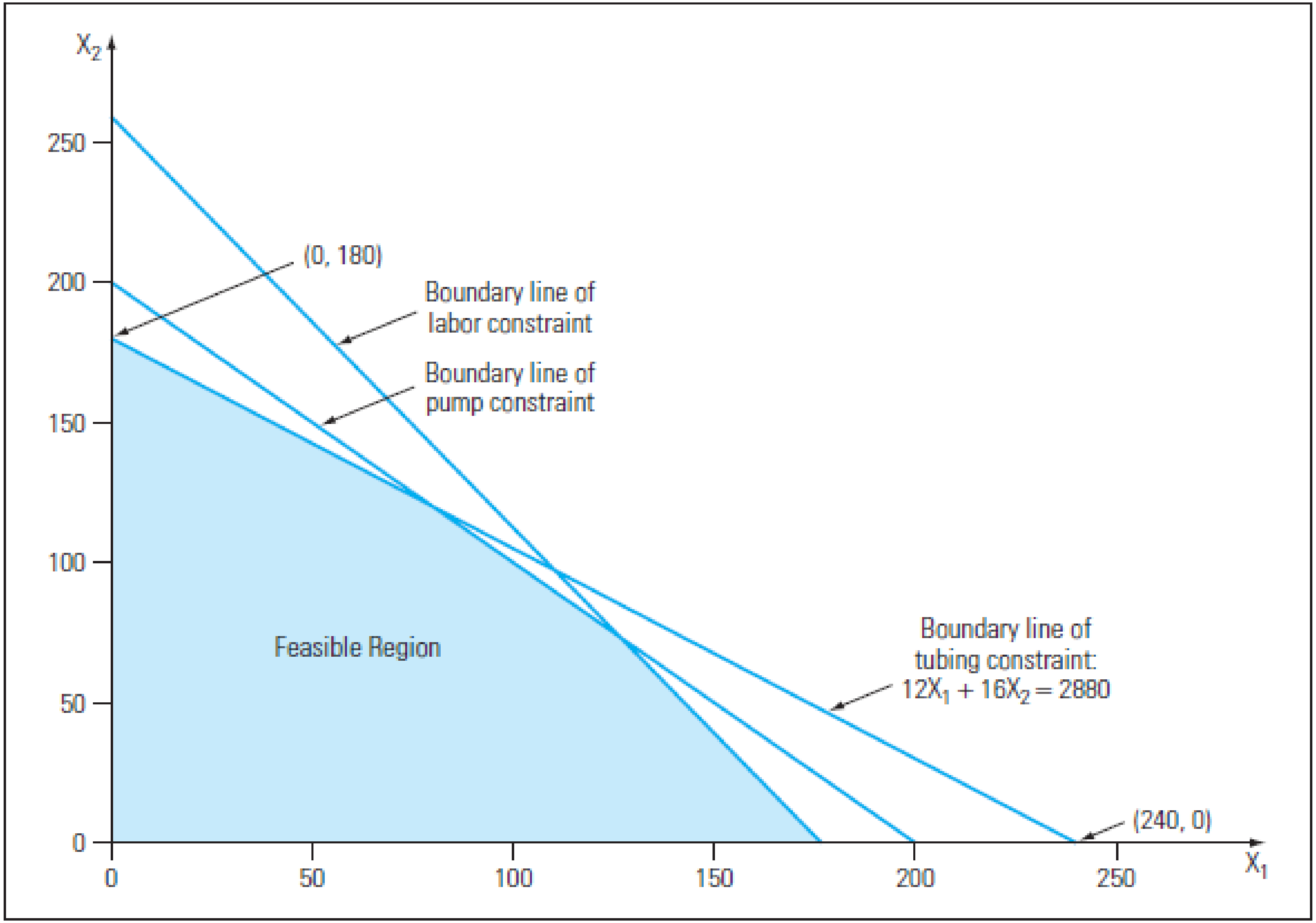
- There are simple lower bounds of zero on the variables X1 and X2 because it is impossible to produce a negative number of hot tubs.

$X1 \geq 0; X2 \geq 0$

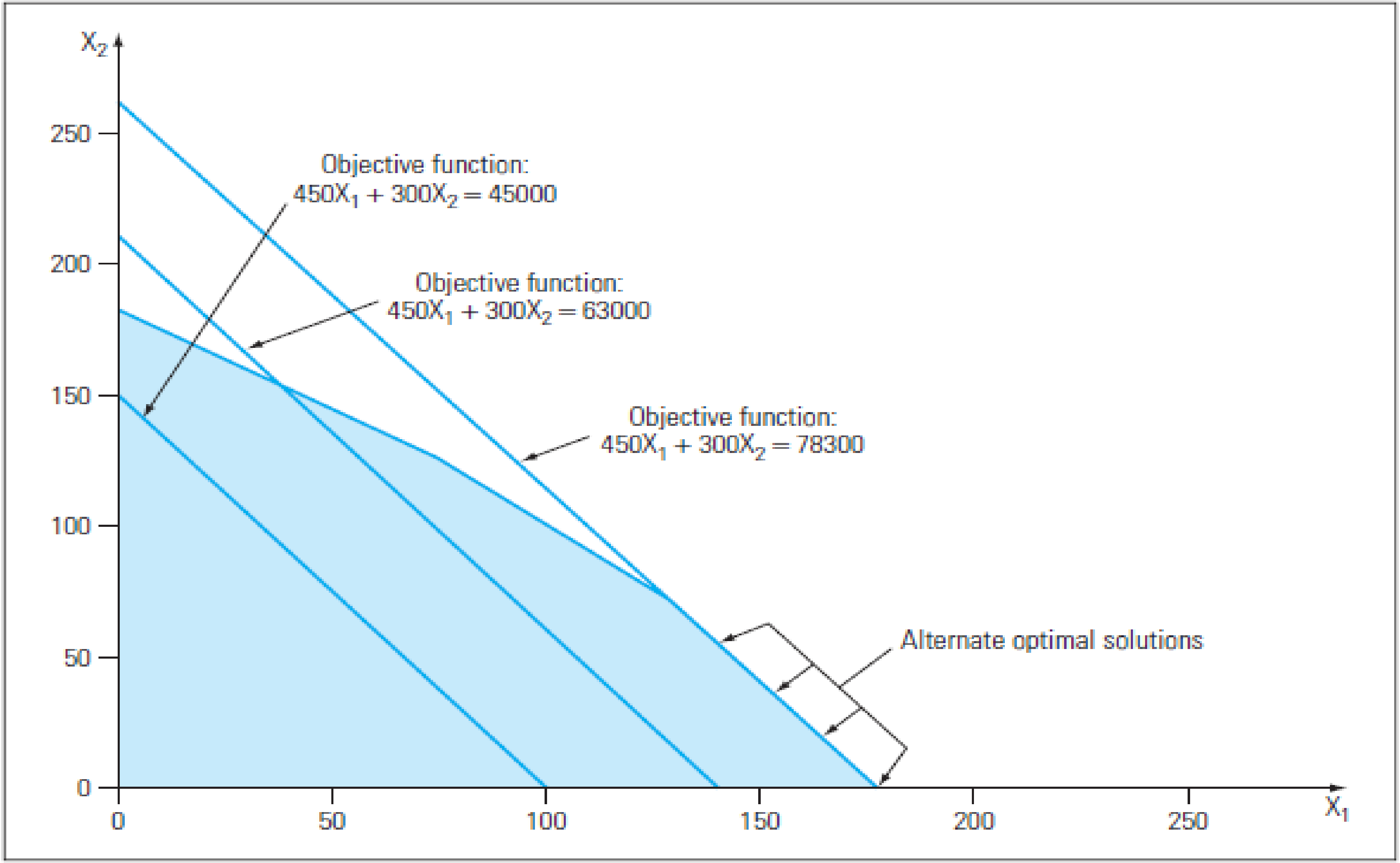
$$
\begin{aligned}
\text{MAX:} \quad & 350X_1 + 300X_2 \\
\text{Subject to:} \quad & 1X_1 + 1X_2 \le 200 \\
& 9X_1 + 6X_2 \le 1{,}566 \\
& 12X_1 + 16X_2 \le 2{,}880 \\
& 1X_1 \ge 0 \\
& 1X_2 \ge 0
\end{aligned}
$$

Run the above Linear Program

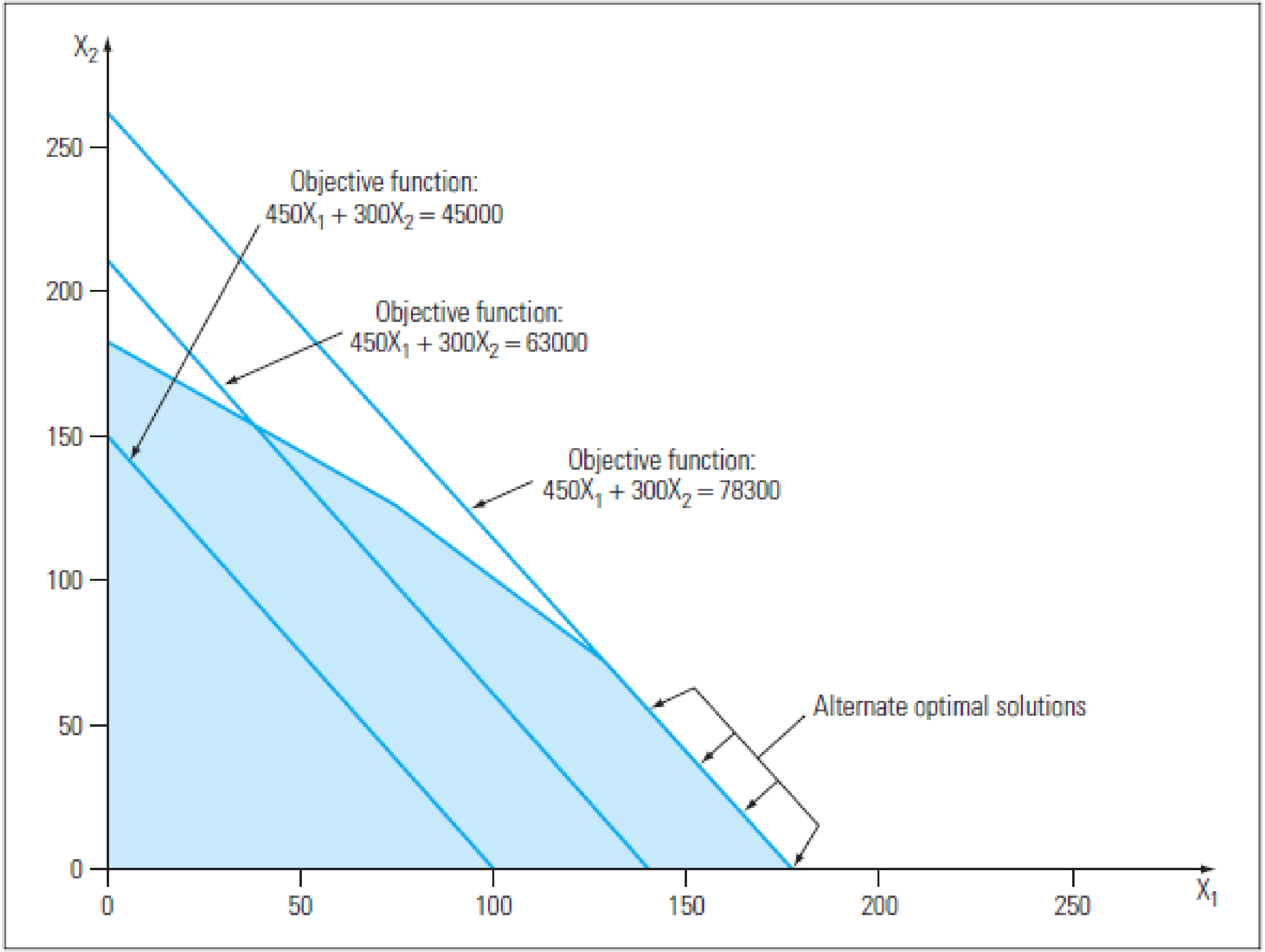
Geometric illustration of feasible region and constraints for LP in previous slide



Geometric illustration of other possibilities

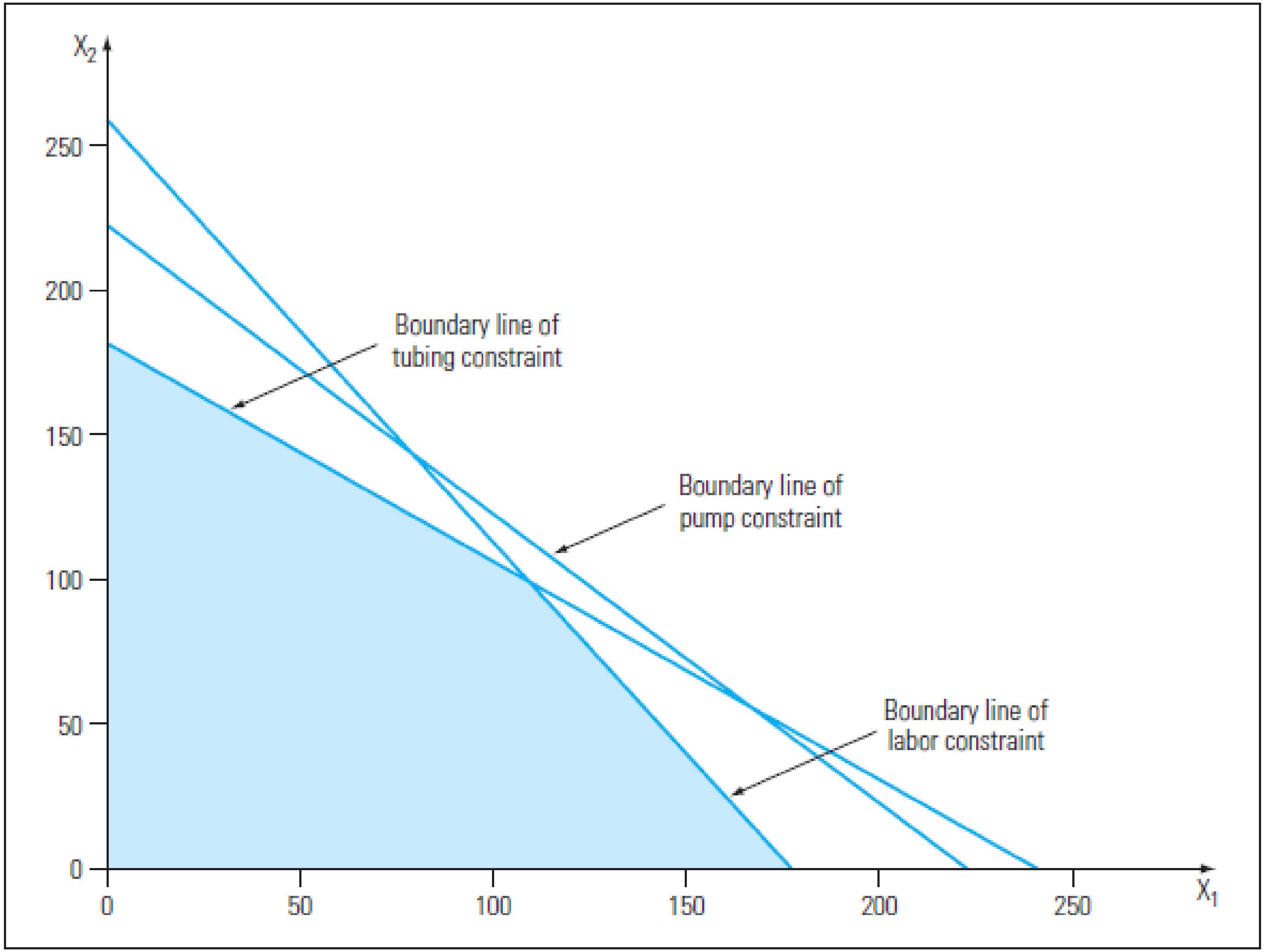


Special cases: Alternate solutions



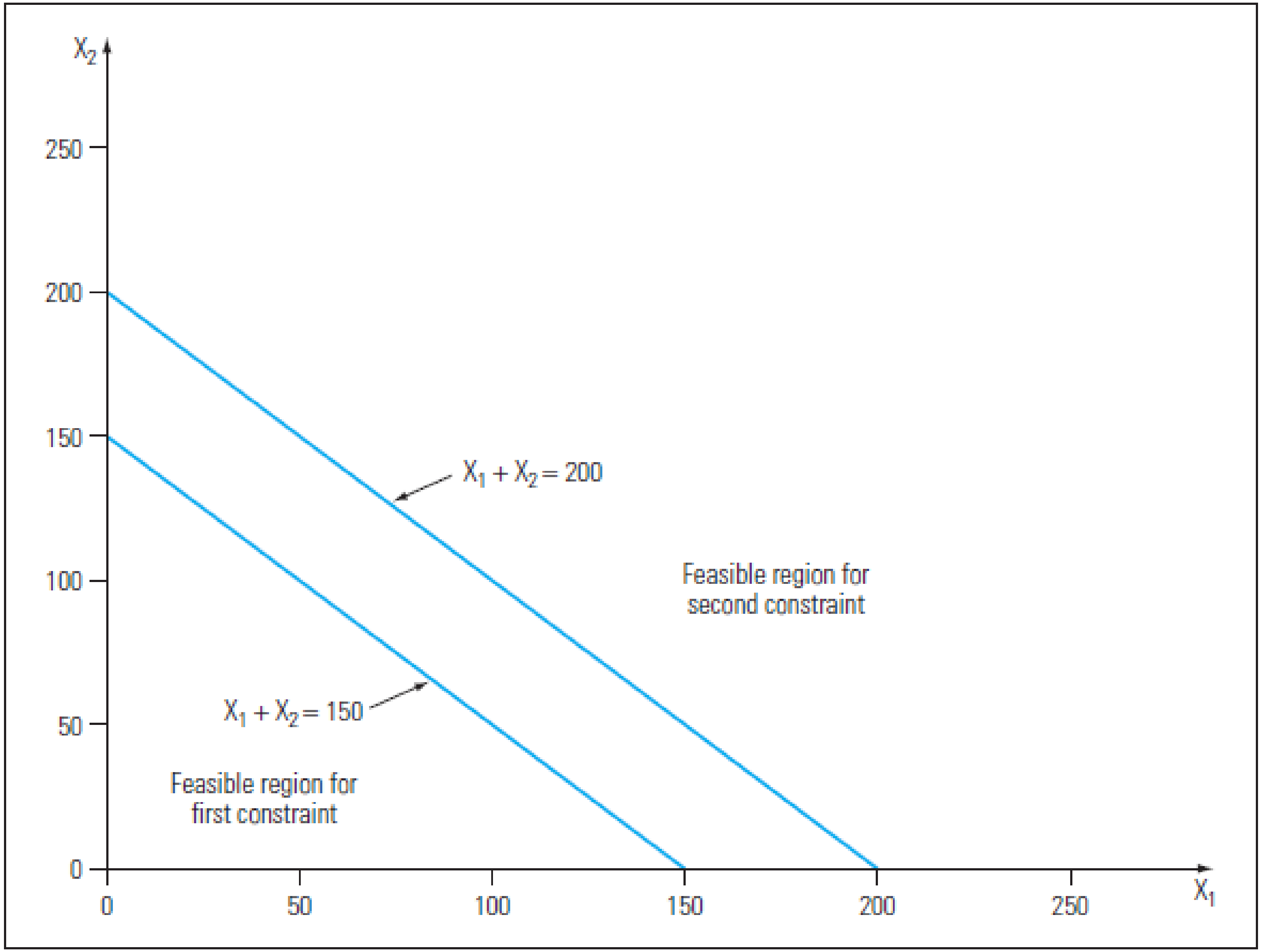
The objective function can move. Different values correspond to the solution. E.g.: Here the profit can be 45,000, 63,000 or 78,300. The max profit is optimal solution

Redundancy



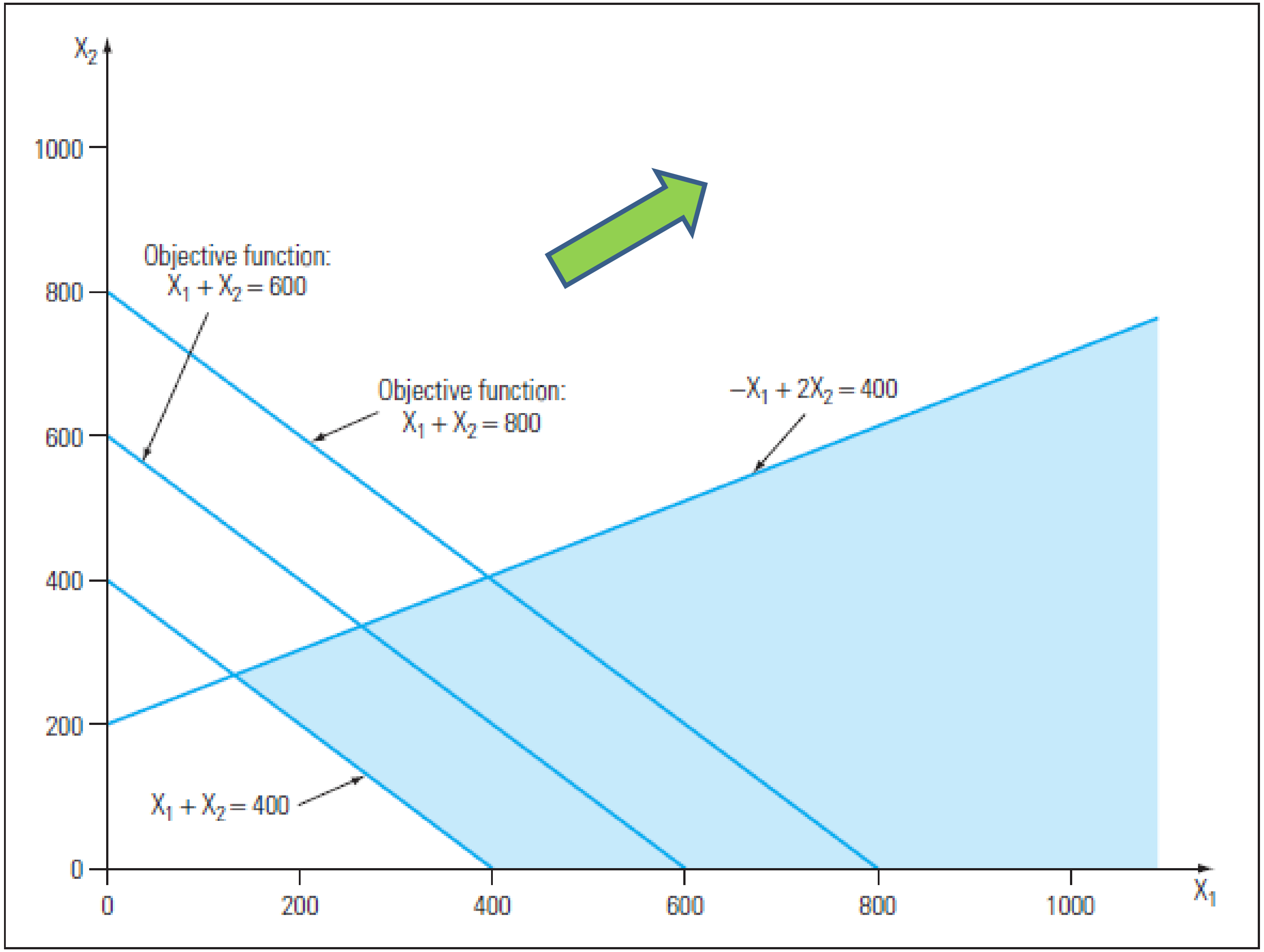
X_2
250
200
150
100
50
0
0
50
100
150
200
250
X_1
Boundary line of tubing constraint
Boundary line of pump constraint
Boundary line of labor constraint

Not feasible



Unbounded

In this case, the objective function can move along the green arrow. The solution can have infinite values



Lab exercise :
Scenario 1 : Original values

Type 1 more profitable, involves more labor but less material as compared to Type 2

$$
\begin{aligned}
\max \quad & 350x_1 + 300x_2 \\
\text{subject to :} \quad & x_1 + x_2 \leq 200 \\
& 9x_1 + 6x_2 \leq 1566 \\
& 12x_1 + 16x_2 \leq 2880 \\
& x_1 \geq 0, x_2 \geq 0
\end{aligned}
$$

LP Formulation

x_1 = , x_2 =

Total profit =

Solution

What if : Scenario 2

Tubing values interchanged :
Type 1 more profitable, but involves more labor and material, all other values remain unchanged.

$$\begin{aligned} \max \quad & 350x_1 + 300x_2 \\ \text{subject to :} \quad & x_1 + x_2 \leq 200 \\ & 9x_1 + 6x_2 \leq 1566 \\ & 16x_1 + 12x_2 \leq 2880 \\ & x_1 \geq 0, x_2 \geq 0 \end{aligned}$$

LP Formulation

x_1 =, x_2 =

Total profit =

Solution

What if : Scenario 3

Both labor hours and tubing interchanged
i.e. Type 1 is more profitable, involves less material and labor

$$\begin{array}{ll} \max & 350x_1 + 300x_2 \\ \text{subject to :} & x_1 + x_2 \leq 200 \\ & 6x_1 + 9x_2 \leq 1566 \\ & 12x_1 + 16x_2 \leq 2880 \\ & x_1 \geq 0, x_2 \geq 0 \end{array}$$

LP Formulation

x_1 =, x_2 =
Total profit =

Solution

What if : Scenario 4

Type 1 more profitable but consumes more labor and significantly more material, also available labor hours = 1700

$$\begin{array}{ll} \max & 350x_1 + 300x_2 \\ \text{subject to :} & x_1 + x_2 \leq 200 \\ & 9x_1 + 6x_2 \leq 1700 \\ & 36x_1 + 16x_2 \leq 2880 \\ & x_1 \geq 0, x_2 \geq 0 \end{array}$$

LP Formulation

x_1 = , x_2 =

Total profit =

Solution

What if : Scenario 5

Labor and tubing interchanged, total labor hours = 1700
Type 1 more profitable, consumes less labor but more material than Type 2.

$$
\begin{aligned}
\max \quad & 350x_1 + 300x_2 \\
\text{subject to :} \quad & x_1 + x_2 \le 200 \\
& 6x_1 + 9x_2 \le 1700 \\
& 16x_1 + 12x_2 \le 2880 \\
& x_1 \ge 0, x_2 \ge 0
\end{aligned}
$$

LP Formulation

x_1 =, x_2 =

Total profit =

Solution

What if : Scenario 6

Tubing interchanged :

Type 1 more profitable, consumes more labor and more material than Type 2, also available labor hours = 1700

$$
\begin{array}{rc}
\max & 350x_1 + 300x_2 \\
\text{subject to :} & x_1 + x_2 \leq 200 \\
 & 9x_1 + 6x_2 \leq 1700 \\
 & 16x_1 + 12x_2 \leq 2880 \\
 & x_1 \geq 0, x_2 \geq 0
\end{array}
$$

LP Formulation

x_1 =, x_2 =

Total profit =

Solution

What if : Scenario 7

All other values unchanged, only labor hours changed to 1700.

$$\begin{aligned} \text{max} \quad & 350x_1 + 300x_2 \\ \text{subject to :} \quad & x_1 + x_2 \leq 200 \\ & 9x_1 + 6x_2 \leq 1700 \\ & 12x_1 + 16x_2 \leq 2880 \\ & x_1 \geq 0, x_2 \geq 0 \end{aligned}$$

LP Formulation

x_1 =, x_2 =

Total profit =

Solution

INTEGER AND BINARY PROGRAMMING

- Suppose, for example, that Blue Ridge Hot Tubs has only 1,520 hours of labor and 2,650 feet of tubing available during its next production cycle. The company might be interested in solving the following ILP problem

$$
\begin{array}{llll}
\text{MAX:} & 350X_1 + 300X_2 & & \text{\} profit} \\
\text{Subject to:} & 1X_1 + 1X_2 \le 200 & & \text{\} pump constraint} \\
& 9X_1 + 6X_2 \le 1{,}520 & & \text{\} labor constraint} \\
& 12X_1 + 16X_2 \le 2{,}650 & & \text{\} tubing constraint} \\
& X_1, X_2 \ge 0 & & \text{\} nonnegativity conditions} \\
& X_1, X_2 \text{ must be integers} & & \text{\} integrality conditions}
\end{array}
$$

Solve the Linear Program

An Assignment problem

- A 400-meter medley relay involves four different swimmers, who successively swim 100 meters of the backstroke, breaststroke, butterfly and freestyle. A coach has six very fast swimmers whose expected times (in seconds) in the individual events are given in following table.

Which swimmers must the coach assign for which style so that the overall team swimming time is reduced?

	Event 1 (backstroke)	Event 2 (breaststroke)	Event 3 (butterfly)	Event 4 (freestyle)
Swimmer 1	68	73	63	57
Swimmer 2	67	70	64	58
Swimmer 3	68	72	69	55
Swimmer 4	67	75	70	59
Swimmer 5	71	69	75	57
Swimmer 6	69	71	66	59

A “Cost matrix” C where C(i,j) denotes the cost of assigning swimmer i to event j

Solving the assignment problem

Seeking a solution in the form of a matrix (say X) which is the same size as cost matrix where :

$$X(i,j) = \begin{cases} 1 & \text{if swimmer } i \text{ is assigned to event } j \\ 0 & \text{otherwise} \end{cases}$$

	Event 1	Event 2	Event 3	Event 4
Swimmer 1	X(1,1)	X(1,2)		
Swimmer 2				
Swimmer 3		…	…	
Swimmer 4		…	…	
Swimmer 5				
Swimmer 6	X(6,1)			X(6,4)

Total number of variables being solved for = size of cost matrix = 24

Efficiently algorithms exist for solving the Assignment Problem.
Eg. linear_sum_assignment in python scipy package is one implementation

It can also be formulated as a Linear Program as follows :
(Linear Program formulation especially useful if we wish to solve not the standard version but a variant)

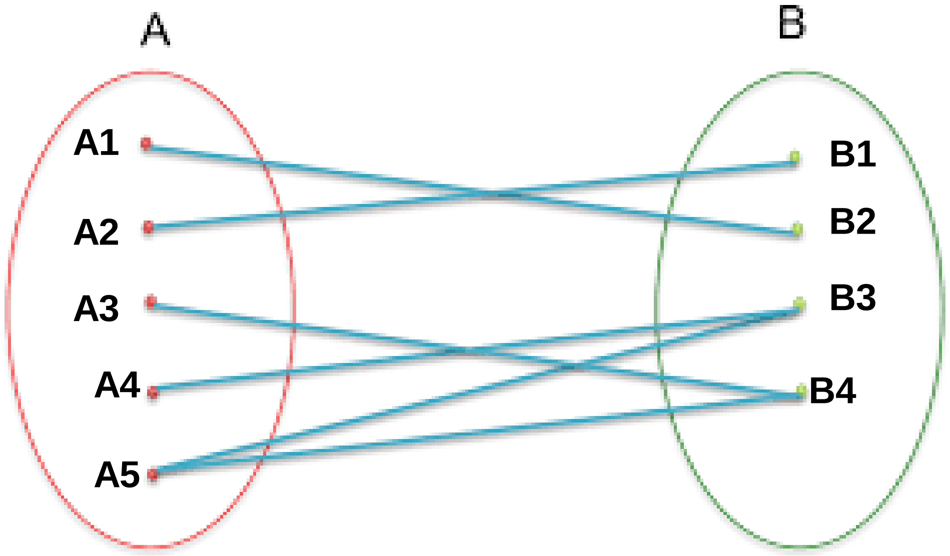
Total cost = sum of individual costs. Hence objective function is $\sum_{i=1}^{6}\sum_{j=1}^{4} C(i,j)X(i,j)$

Constraints

$\sum_{j=1}^{4} X(i,j) \leq 1$	$1 \leq i \leq 6$	row sums must not exceed 1 since any swimmer can be assigned to at most one event
$\sum_{i=1}^{6} X(i,j) = 1$	$1 \leq j \leq 4$	column sums must be exactly one since each event is assigned to exactly one swimmer
$\sum_{j} X(i,j) \leq 1$	$1 \leq i \leq 6$	row sums must be ≤ 1 since one swimmer can be assigned to at most one event
$X(i,j) \in \{0,1\}$	$1 \leq i \leq 6$ $1 \leq j \leq 4$	condition that $X(i,j) \in \{0,1\}$ is sometimes relaxed to $0 \leq X(i,j) \leq 1$

Another Assignment problem

	B1	B2	B3	B4
A1		1		
A2	1			
A3				1
A4			1	
A5			1	1

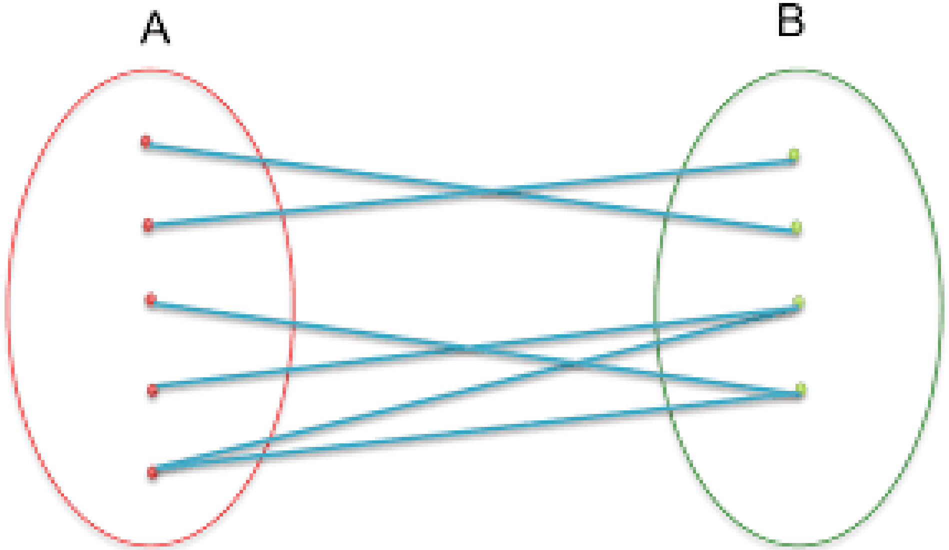


Employees with required skill sets for projects shown as a matrix and as a graph

Suppose A1,...,A5 are employees and B1,...,B4 are projects.
An entry of 1 in the table (correspongly presence of edge in the graph) iindicates that the employee is skilled to do that project.

Assuming only one employee can be matched to one project, find an assignment that maximizes utilization by matching as many employees to projects as possible.

A small detour ... Bipartite matching



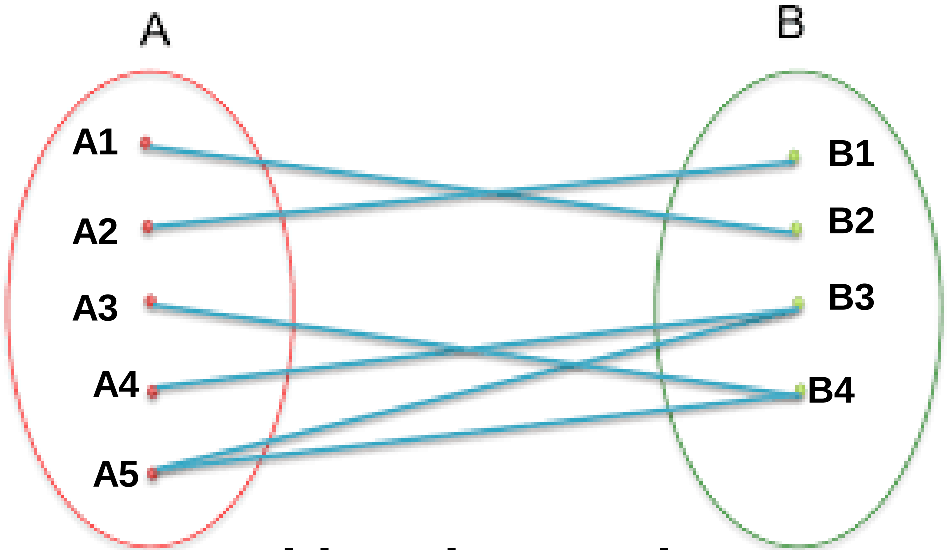
A bipartite graph with the partitions of vertices indicated by A and B.

A bipartite graph is one where the set of vertices can be partitioned into two disjoint sets (say A and B) such that any edge in the graph has an end point in A and another end point in B.

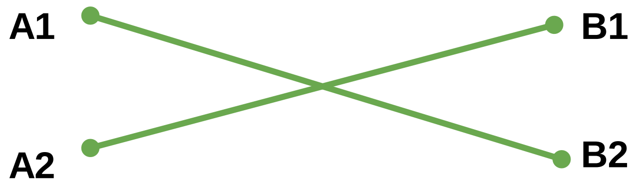
Bipartite graphs are a natural representation in many applications.

Eg. A could represent the **set of employees** and B the **set of projects,** an **edge** could indicate that the corresponding employee has skill sets to work on that project.

Bipartite matching



A bipartite graph

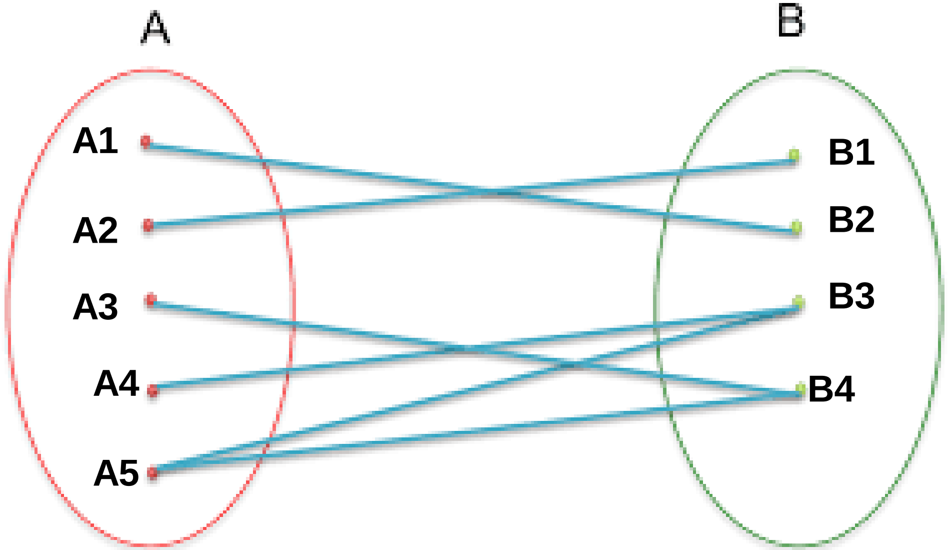


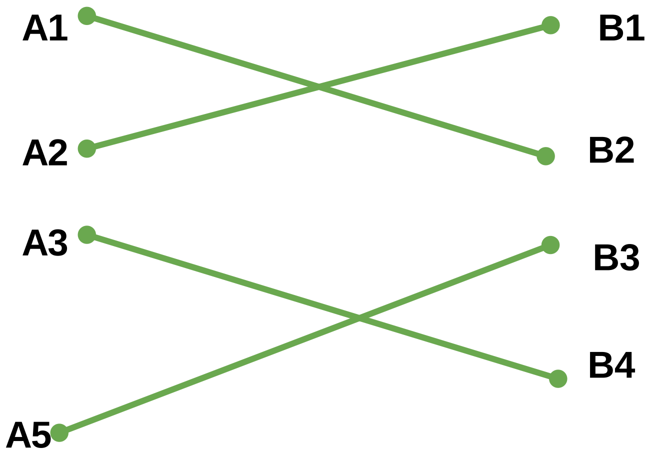
A matching on the bipartite graph

Given a graph (not necessarily bipartite), a subset of the edges is a **matching** if no two edges share (or are incident on) the same vertex.

- In practical applications a matching on a bipartite graph corresponds to a one is to one assignment.

Bipartite matching





A maximum matching on the bipartite graph to the left

A **maximum matching** is a matching of maximum cardinality (i.e. maximum possible number of edges).

Solving bipartite matching problems

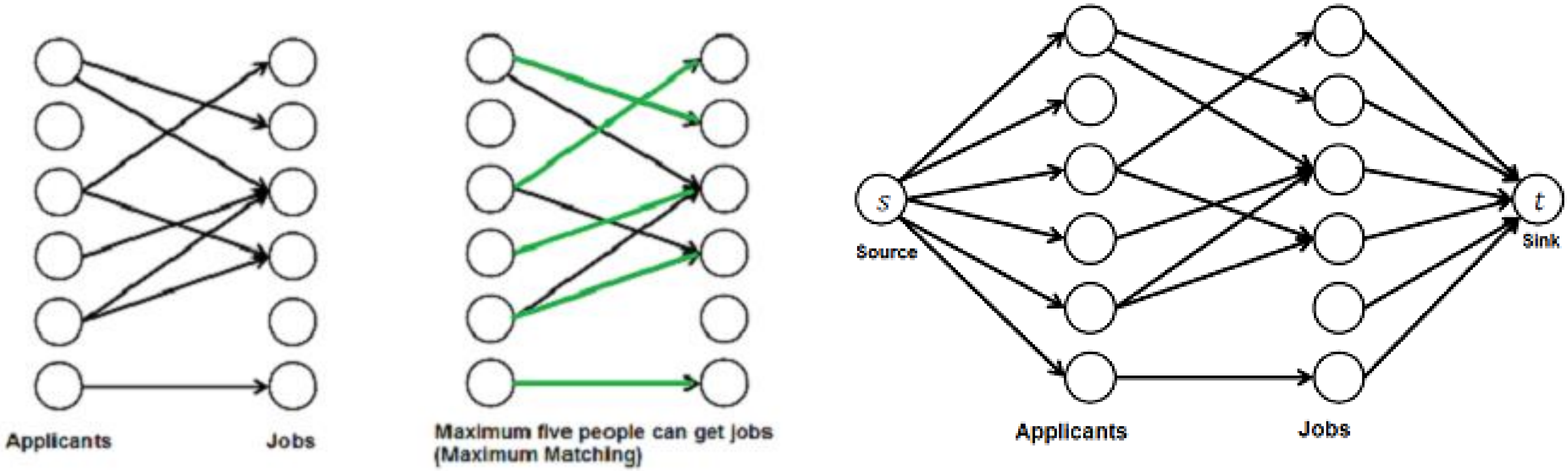


Image source : https://www.geeksforgeeks.org/maximum-bipartite-matching/

Bipartite matching problems can also be formulated as max flow problems (discussed next).
Typically additional nodes termed as **source** and **sink** are introduced.

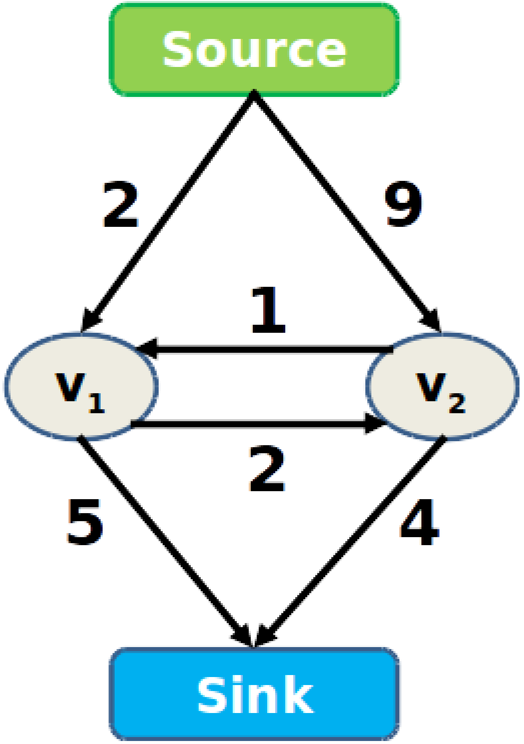
Another small detour ... Max flow

A **flow network** is a **directed graph** with :

- two special vertices, namely the source **s** and the sink **t**
- each edge is associated with a capacity, which is an upper bound on the flow rate (= units/time)

- Material originates from the source and terminates at the sink flowing through the various edges (the edges can be thought of as conduits).
    - At all nodes (except source and sink) **flow conservation** holds. i.e. no material can accumulate at any node. **Inflow = Outflow**
    - The actual flow along any edge is **non-negative** (along the edge direction) and cannot exceed the capacity of that edge.

Max flow problem



Example from MAP Estimation Algorithms in Computer Vision : Tutorial by Pushmeet Kohli, ICCV 2008

Available at : http://www.robots.ox.ac.uk/~pawan/eccv08_tutorial/Tutorial_Part2.ppt

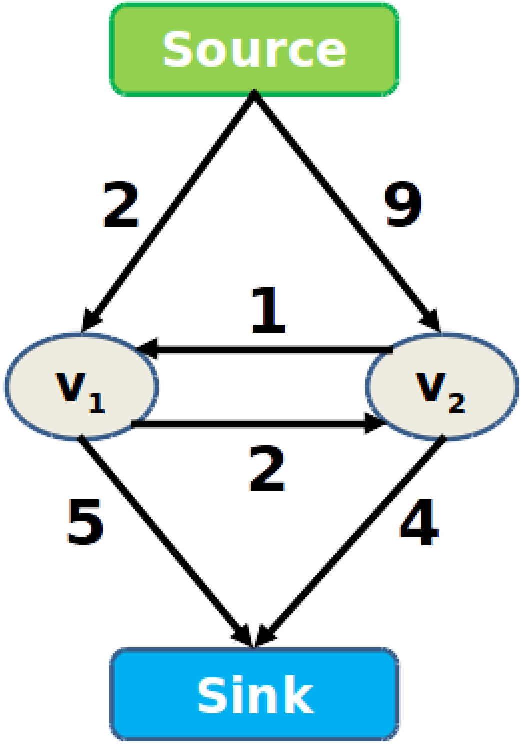
Given a network, compute the maximum flow that can be sent through the network.

Max flow techniques have been applied to various problems including image segmentation.

A variety of efficient techniques have been developed for the maximum flow problem. Eg. Augmented path, push-relabel etc.

Max flow computation can also be formulated as a Linear Program (refer next slide).

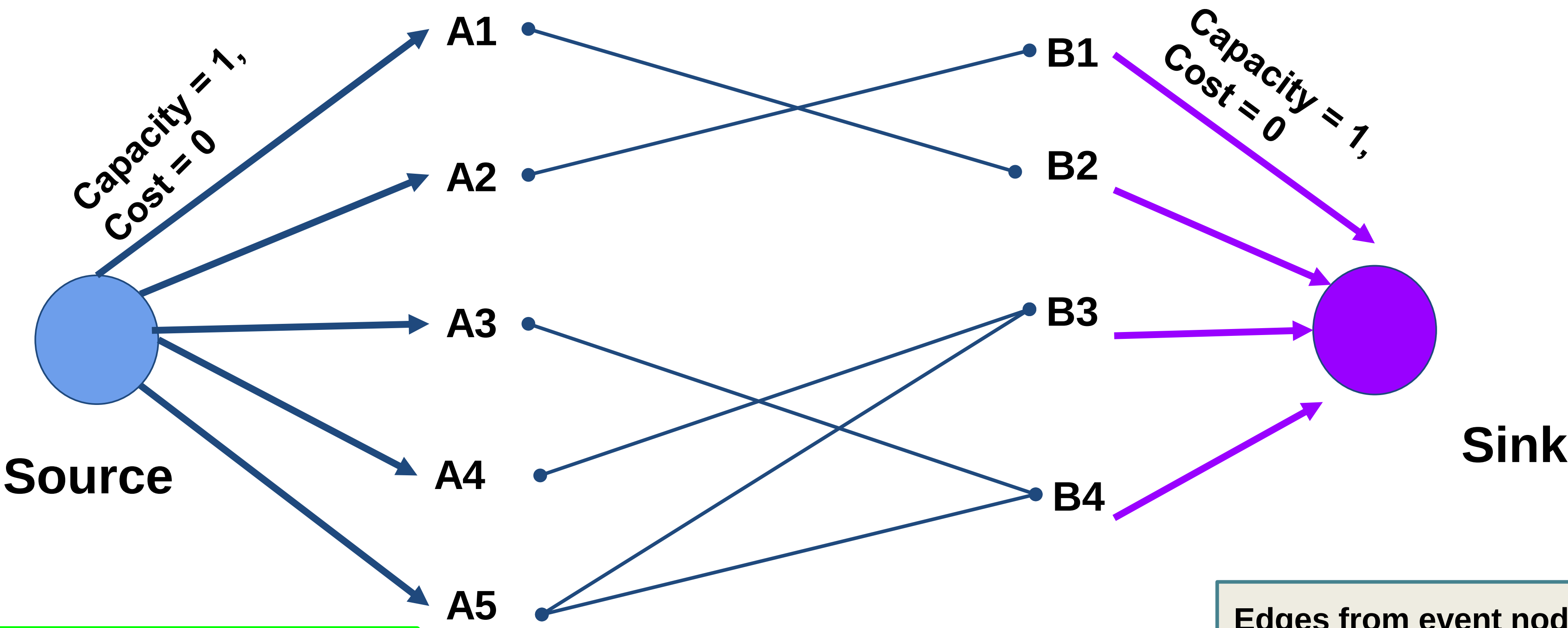
Exercise : Solve the max flow problem for the network by formulating as a linear program



Edge	**Capacity**	**Variable**
Source-v_1	2	x_1
Source-v_2	9	x_2
v_1-v_2	2	x_3
v_2-v_1	1	x_4
v_1 - Sink	5	x_5
v_2-Sink	4	x_6

Objective function : $x_1 + x_2$
or
(alternately) : $x_5 + x_6$ (Why?)

Flow network for the job assignment problem



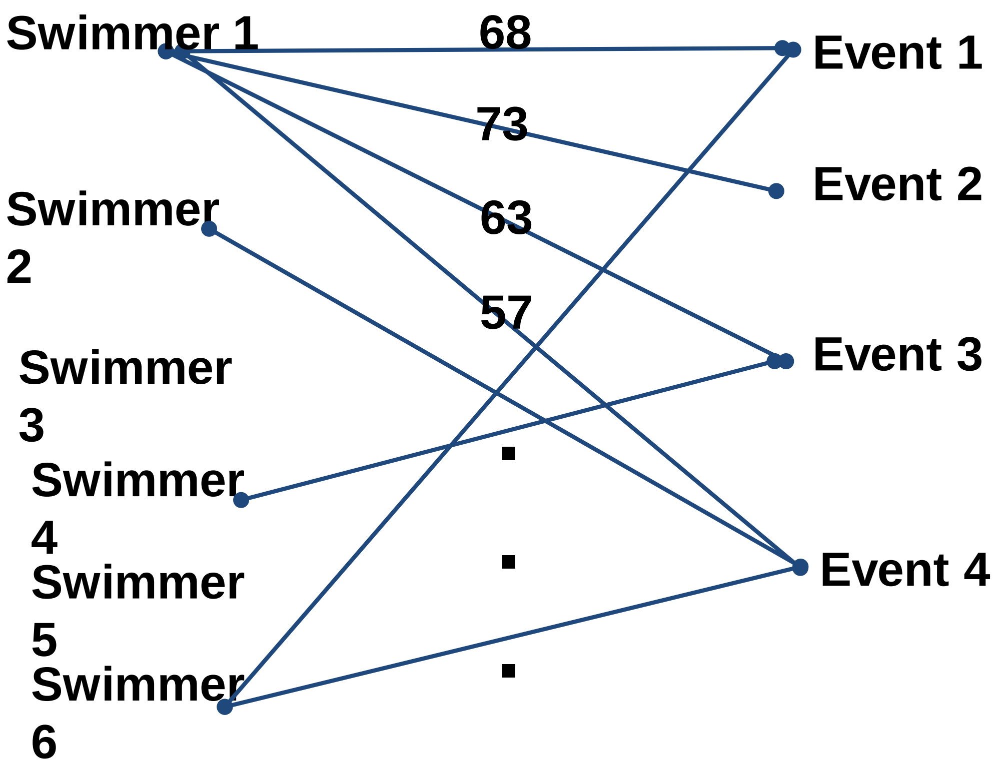
Edges from source to swimmer nodes

Capacity = 1 (to ensure 1 employee is assigned to at most one project)
Cost = 0

Edges from event nodes to sink

Capacity = 1 (to ensure one project is assigned to at most one employee)
Cost = 0

Swimmer event assignment as bipartite matching

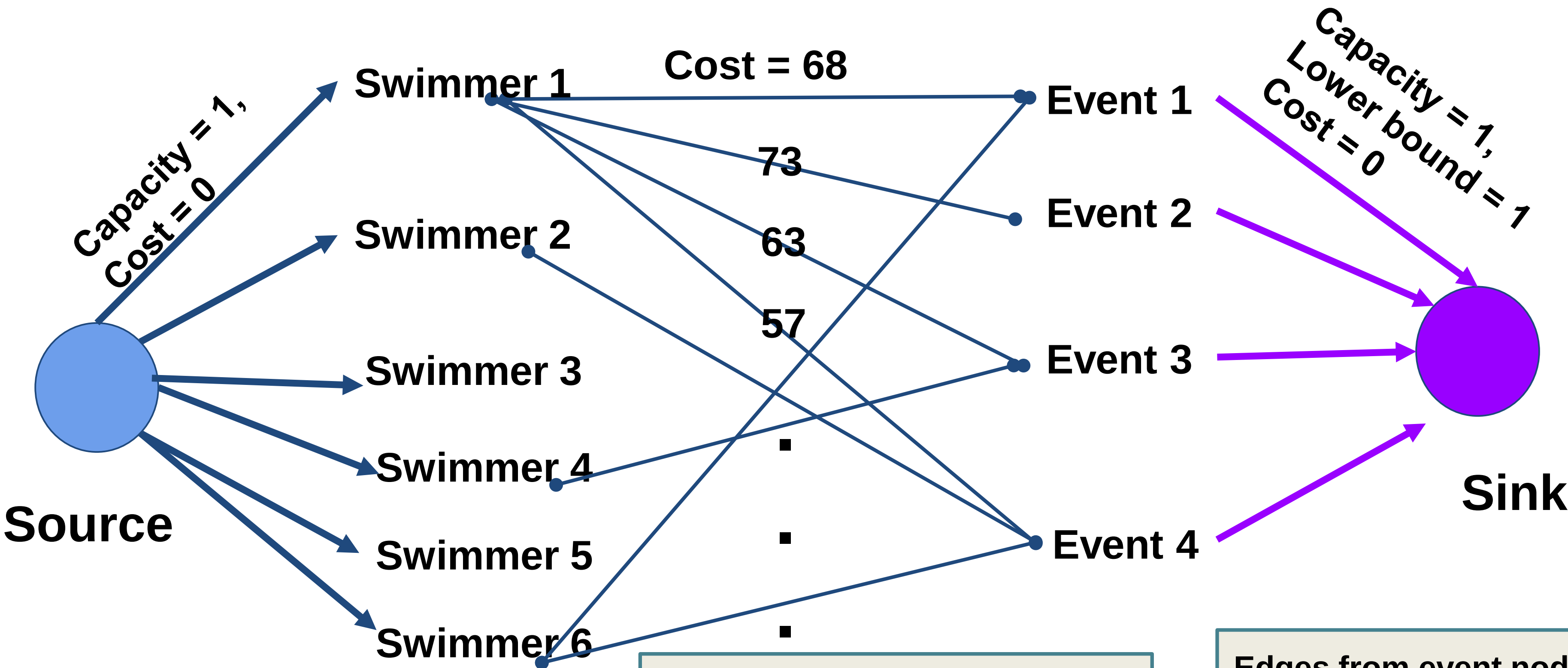


The swimmer event matrix can be represented as a **bipartite graph** where the **edge weights denote the respective times** (i.e. costs).

Note however the solution we are interested in is
NOT finding a maximum matching but rather
finding the maximum matching with lowest cost.

Algorithms more efficient than Linear Programming are available for bipartite maximum matching.
However, these methods cannot be directly applied to variants of the above.
Soln : Formulate as a **min cost max flow problem**
and solve the corresponding linear program.

Swimmer relay revisited : Flow network



Edges from source to swimmer nodes

Capacity = 1 (to ensure 1 swimmer is assigned to at most one event)
Cost = 0

Edges from swimmer nodes to event nodes

Capacity : Capacity : High value, can even be ∞
Cost edge from swimmer i to event j = Time Swimmer i takes for event j

Edges from event nodes to sink

Capacity = 1 (to ensure one event is assigned to at most one swimmer)
Cost = 0

Solution of variants of bipartite matching and max flow

- Algorithms more efficient than Linear Programming are available for **maximum matching in bipartite graphs** and **max flow problems**.
  - However these cannot be directly applied to the variants described earlier (eg. minimum cost maximum matching or the equivalent min cost max flow).

- These variants can be solved by formulating as **Linear Programs**.
  - Thus Linear Program together with conceptual tools such as flow networks serve as a useful framework for these (and many other problems).

More use cases and concepts

Transportation Problem

- Tropicsun currently has three citrus groves: 275,000 bags at Mt. Dora, 400,000 bags at Eustis, and 300,000 bags at Clermont. Tropicsun has three juice processing plants: Ocala, Orlando, and Leesburg with processing capacities: 200,000, 600,000, and 225,000 bags

- The transport company used by Tropicsun charges a flat rate for every mile that each bag of fruit must be transported. Each mile a bag of fruit travels is known as a bag-mile. The distances (in miles) between the groves and processing plants are summarized in the following table:

	Distances (in miles) Between Groves and Plants		
Grove	**Ocala**	**Orlando**	**Leesburg**
Mt. Dora	21	50	40
Eustis	35	30	22
Clermont	55	20	25

Table summarizing the data for the specific transportation problem

	Ocala	Orlando	Leesburg	Supply available
Mt. Dora	21	50	40	275000
Eustis	35	30	22	400000
Clermont	55	20	25	300000
Capacities	200000	600000	225000	

- Can you suggest how the fruit must be routed from each grove to processing plant to minimize cost and meet production need?

Formulate as a Linear Program with the design variables as shown in the table below.

	Ocala (200,000)	Orlando (600,000)	Leesburg (225,000)
Mt. Dora (275,000)	X_{11}	X_{12}	X_{13}
Eustis (400,000)	X_{21}	X_{22}	X_{23}
Clermont (300,000)	X_{31}	X_{32}	X_{33}

The objective function is the total cost incurred (measured as number of bag-miles) i.e.

$$\sum_{i=1}^{3} \sum_{j=1}^{3} D_{ij} X_{ij}$$

where,

$D_{ij} =$ distance from grove i to processing plant j

$X_{ij} =$ number of bushels transported from grove i to processing plant j

Should we minimize or maximize the objective function?

Linear Program formulation

$$
\begin{aligned}
\min \quad & 21X_{11} + 50X_{12} + 40X_{13} + \\
& 35X_{21} + 20X_{22} + 22X_{23} + \\
& 55X_{31} + 20X_{32} + 25X_{33}
\end{aligned}
$$

$$
\begin{aligned}
\text{subject to :} \quad & X_{11} + X_{21} + X_{31} \leq 200{,}000 && \text{(capacity Ocala)} \\
& X_{12} + X_{22} + X_{32} \leq 600{,}000 && \text{(capacity Orlando)} \\
& X_{13} + X_{23} + X_{33} \leq 225{,}000 && \text{(capacity Leesburg)} \\
& X_{11} + X_{12} + X_{13} \leq 275{,}000 && \text{(available max supply Mt. Dora)} \\
& X_{21} + X_{22} + X_{23} \leq 400{,}000 && \text{(available max supply Eustis)} \\
& X_{31} + X_{32} + X_{33} \leq 300{,}000 && \text{(available max supply Clermont)} \\
& X_{ij} \geq 0 \ \text{ for all } i, j && \text{non-negativity constraints}
\end{aligned}
$$

Solution obtained by solving the above Linear Program : X_{11}=X_{12}= ...X_{33}= 0

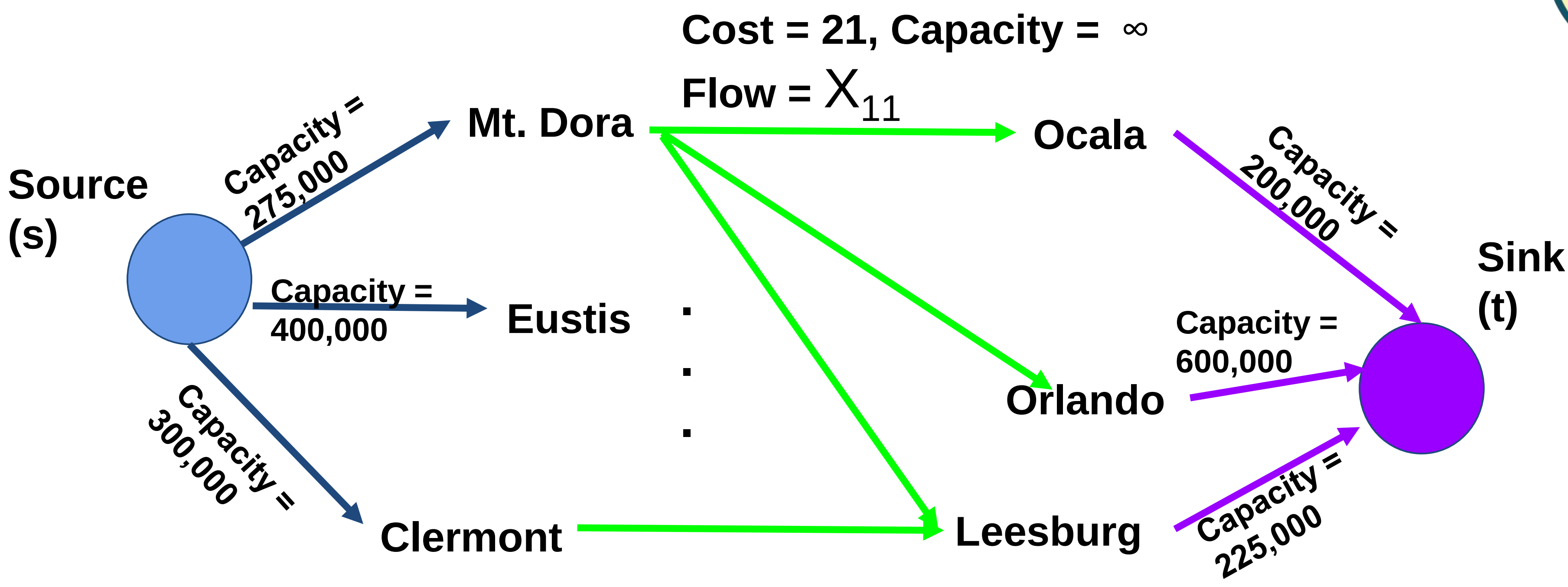
Linear Program : Attempt 1

Question : What went wrong?

Ans. The objective function can be minimized by minimizing the X_{ij} s as much as possible. Since there are no constraints forcing the X_{ij} s to be a certain value (other than $X_{ij} \geq 0$, the objective function is minimized by independently setting each of the X_{ij} s to 0.

Practically we wish to **transport as much fruit as possible** while **minimizing the cost**.
Hence, first need to solve the **Max Flow problem**.

Transportation problem flow network



Edges from source to grove nodes

Capacity = production capacity of respective grove
Cost = 0

Edges from grove nodes to plant nodes

Capacity : High value, can even be ∞
Cost edge from grove i to plant j = Distance in miles

Edges from plant nodes to sink

Capacity = Processing capacity of respective plant
Cost = 0

Solution to max flow problem

Solving max flow problem (not shown) we obtain outflows from the grove nodes as :

- Mt. Dora : 275,000
- Eustis : 400,000
- Clermont : 300,000

i.e. it is possible to process the maximum supply from each grove

Since outflows at the grove nodes are now known, we refer to the flow network and formulate the corresponding Linear Program as shown next.

Linear Program : Attempt 2

$$\begin{aligned}
\min \quad & 21X_{11} + 50X_{12} + 40X_{13} + \\
& 35X_{21} + 20X_{22} + 22X_{23} + \\
& 55X_{31} + 20X_{32} + 25X_{33}
\end{aligned}$$

$$\begin{aligned}
\text{subject to :} \quad & X_{11} + X_{21} + X_{31} \leq 200,000 && \text{(capacity Ocala)} \\
& X_{12} + X_{22} + X_{32} \leq 600,000 && \text{(capacity Orlando)} \\
& X_{13} + X_{23} + X_{33} \leq 225,000 && \text{(capacity Leesburg)} \\
& X_{11} + X_{12} + X_{13} = 275,000 && \text{(available max supply Mt. Dora)} \\
& X_{21} + X_{22} + X_{23} = 400,000 && \text{(available max supply Eustis)} \\
& X_{31} + X_{32} + X_{33} = 300,000 && \text{(available max supply Clermont)} \\
& X_{ij} \geq 0 \ \text{ for all } i, j && \text{non-negativity constraints}
\end{aligned}$$

Solve the resulting LP

Exercise : Suppose the supply and production capacities are as shown in the table below. Formulate the corresponding Linear Program and solve it.

	Ocala	Orlando	Leesburg	Supply available
Mt. Dora	21	50	40	225000
Eustis	35	30	22	600000
Clermont	55	20	25	200000
Capacities	400000	300000	275000	

The Linear Program for the modified case (bottleneck on the processing side) is :

$$\min \quad 21X_{11} + 50X_{12} + 40X_{13} + 35X_{21} + 20X_{22} + 22X_{23} + 55X_{31} + 20X_{32} + 25X_{33}$$

$$\begin{aligned}
\text{subject to :} \quad & X_{11} + X_{21} + X_{31} = 400{,}000 && \text{(capacity Ocala)} \\
& X_{12} + X_{22} + X_{32} = 300{,}000 && \text{(capacity Orlando)} \\
& X_{13} + X_{23} + X_{33} = 275{,}000 && \text{(capacity Leesburg)} \\
& X_{11} + X_{12} + X_{13} \leq 225000 && \text{(available max supply Mt. Dora)} \\
& X_{21} + X_{22} + X_{23} \leq 600{,}000 && \text{(available max supply Eustis)} \\
& X_{31} + X_{32} + X_{33} \leq 200{,}000 && \text{(available max supply Clermont)} \\
& X_{ij} \geq 0 \ \text{ for all } i, j && \text{non-negativity constraints}
\end{aligned}$$

Solve the resulting LP

Capital budget allocation

- In his position as vice president of research and development (R&D) for CRT Technologies, Mark Schwartz is responsible for evaluating and choosing which R&D projects to support. The company received 18 R&D proposals from its scientists and engineers, and identified six projects as being consistent with the company's mission.
- However, the company does not have the funds available to undertake all six projects. Mark must determine which of the projects to select. The funding requirements for each project are summarized in the following table along with the NPV the company expects each project to generate.

Project	Expected NPV (in $1,000s)	Capital (in $1,000s) Required in				
		Year 1	Year 2	Year 3	Year 4	Year 5
1	$141	$ 75	$25	$20	$15	$10
2	$187	$ 90	$35	$ 0	$ 0	$30
3	$121	$ 60	$15	$15	$15	$15
4	$ 83	$ 30	$20	$10	$ 5	$ 5
5	$265	$100	$25	$20	$20	$20
6	$127	$ 50	$20	$10	$30	$40

- The company currently has $250,000 available to invest in new projects. It has budgeted $75,000 for continued support for these projects in year 2 and $50,000 per year for years 3, 4, and 5. Surplus funds in any year are re-appropriated for other uses within the company and may not be carried over to future years.

$$
\begin{array}{ll}
\text{MAX:} & 141X_1 + 187X_2 + 121X_3 + 83X_4 + 265X_5 + 127X_6 \\
\text{Subject to:} & 75X_1 + 90X_2 + 60X_3 + 30X_4 + 100X_5 + 50X_6 \le 250 \\
 & 25X_1 + 35X_2 + 15X_3 + 20X_4 + 25X_5 + 20X_6 \le 75 \\
 & 20X_1 + 0X_2 + 15X_3 + 10X_4 + 20X_5 + 10X_6 \le 50 \\
 & 15X_1 + 0X_2 + 15X_3 + 5X_4 + 20X_5 + 30X_6 \le 50 \\
 & 10X_1 + 30X_2 + 15X_3 + 5X_4 + 20X_5 + 40X_6 \le 50 \\
 & \text{All } X_i \text{ must be binary}
\end{array}
$$

Staff allocation problem

- *Air-Express is an express shipping service that guarantees overnight delivery of packages anywhere in the continental United States. The company has various operations centers, called hubs, at airports in major cities across the country. Packages are received at hubs from other locations and then shipped to intermediate hubs or to their final destinations.*
- *The manager of the Air-Express hub in Baltimore, Maryland, is concerned about labor costs at the hub and is interested in determining the most effective way to schedule workers. The hub operates seven days a week, and the number of packages it handles each day varies from one day to the next. Using historical data on the average number of packages received each day, the manager estimates the number of workers needed to handle the packages as:*

Day of Week	Workers Required
Sunday	18
Monday	27
Tuesday	22
Wednesday	26
Thursday	25
Friday	21
Saturday	19

- The package handlers working for Air-Express are unionized and are guaranteed a five-day work week with two consecutive days off. The base wage for the handlers is $655 per week. Because most workers prefer to have Saturday or Sunday off, the union has negotiated bonuses of $25 per day for its members who work on these days. The possible shifts and salaries for package handlers are:

Shift	Days Off	Wage
1	Sunday and Monday	$680
2	Monday and Tuesday	$705
3	Tuesday and Wednesday	$705
4	Wednesday and Thursday	$705
5	Thursday and Friday	$705
6	Friday and Saturday	$680
7	Saturday and Sunday	$655

- The manager wants to keep the total wage expense for the hub as low as possible. With this in mind, how many package handlers should be assigned to each shift if the manager wants to have a sufficient number of workers available each day?

Decision variables

- X1 _ the number of workers assigned to shift 1
- X2 _ the number of workers assigned to shift 2
- X3 _ the number of workers assigned to shift 3
- X4 _ the number of workers assigned to shift 4
- X5 _ the number of workers assigned to shift 5
- X6 _ the number of workers assigned to shift 6
- X7 _ the number of workers assigned to shift 7

Objective and constraints

The LP model for the Air-Express scheduling problem is summarized as:

MIN: $680X_1 + 705X_2 + 705X_3 + 705X_4 + 705X_5 + 680X_6 + 655X_7$ } total wage expense

Subject to:

$0X_1 + 1X_2 + 1X_3 + 1X_4 + 1X_5 + 1X_6 + 0X_7 \geq 18$ } workers required on Sunday

$0X_1 + 0X_2 + 1X_3 + 1X_4 + 1X_5 + 1X_6 + 1X_7 \geq 27$ } workers required on Monday

$1X_1 + 0X_2 + 0X_3 + 1X_4 + 1X_5 + 1X_6 + 1X_7 \geq 22$ } workers required on Tuesday

$1X_1 + 1X_2 + 0X_3 + 0X_4 + 1X_5 + 1X_6 + 1X_7 \geq 26$ } workers required on Wednesday

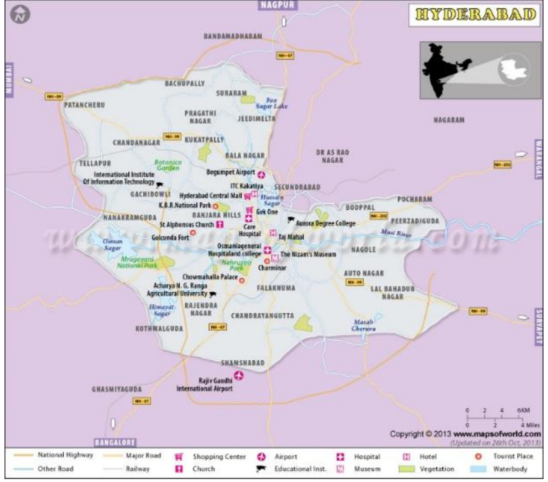
$1X_1 + 1X_2 + 1X_3 + 0X_4 + 0X_5 + 1X_6 + 1X_7 \geq 25$ } workers required on Thursday

$1X_1 + 1X_2 + 1X_3 + 1X_4 + 0X_5 + 0X_6 + 1X_7 \geq 21$ } workers required on Friday

$1X_1 + 1X_2 + 1X_3 + 1X_4 + 1X_5 + 0X_6 + 0X_7 \geq 19$ } workers required on Saturday

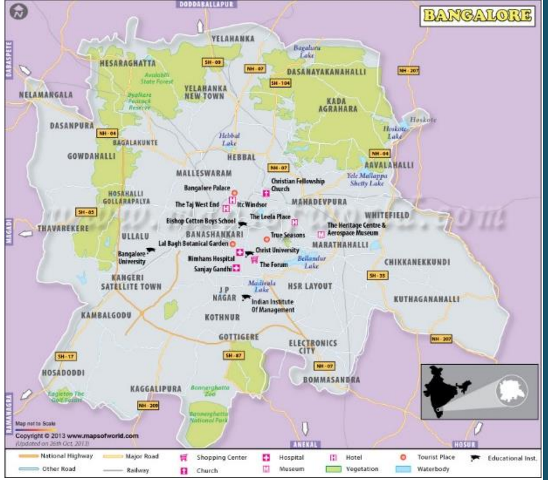
$X_1, X_2, X_3, X_4, X_5, X_6, X_7 \geq 0$

All X_i must be integers



INSOFE

Inspire...Educate...Transform.



HYDERABAD

2nd Floor, Jyothi Imperial, Vamsiram Builders, Old Mumbai Highway, Gachibowli, Hyderabad - 500 032

+91-9701685511 (Individuals)

+91-9618483483 (Corporates)

BENGALURU

Floors 1-3, L77, 15th Cross Road, 3A Main Road, Sector 6, HSR Layout, Bengaluru – 560 102

+91-9502334561 (Individuals)

+91-9502799088 (Corporates)

Social Media

Web:	
Facebook:	https://www.facebook.com/insofe
Twitter:	https://twitter.com/Insofeedu
YouTube:	
SlideShare:	
LinkedIn:	