Decision Optimization part 2

michal.kordyzon@pl.ibm.com

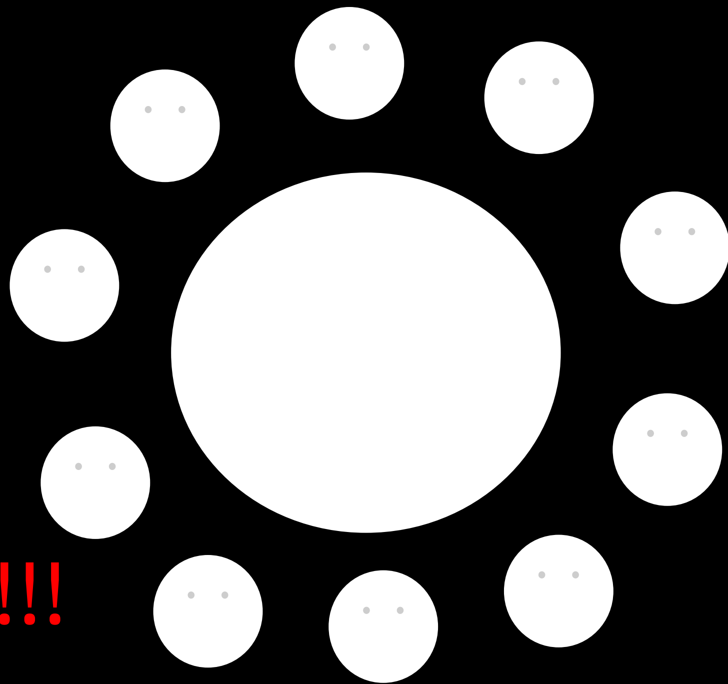
Agenda

1 – quick recap
2 – solving network problems
3 – integer programming, relaxation

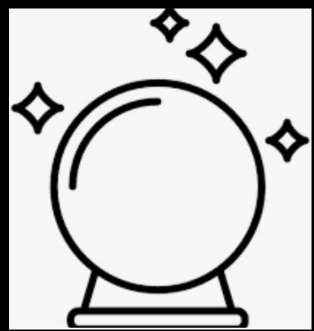
1 – quick recap

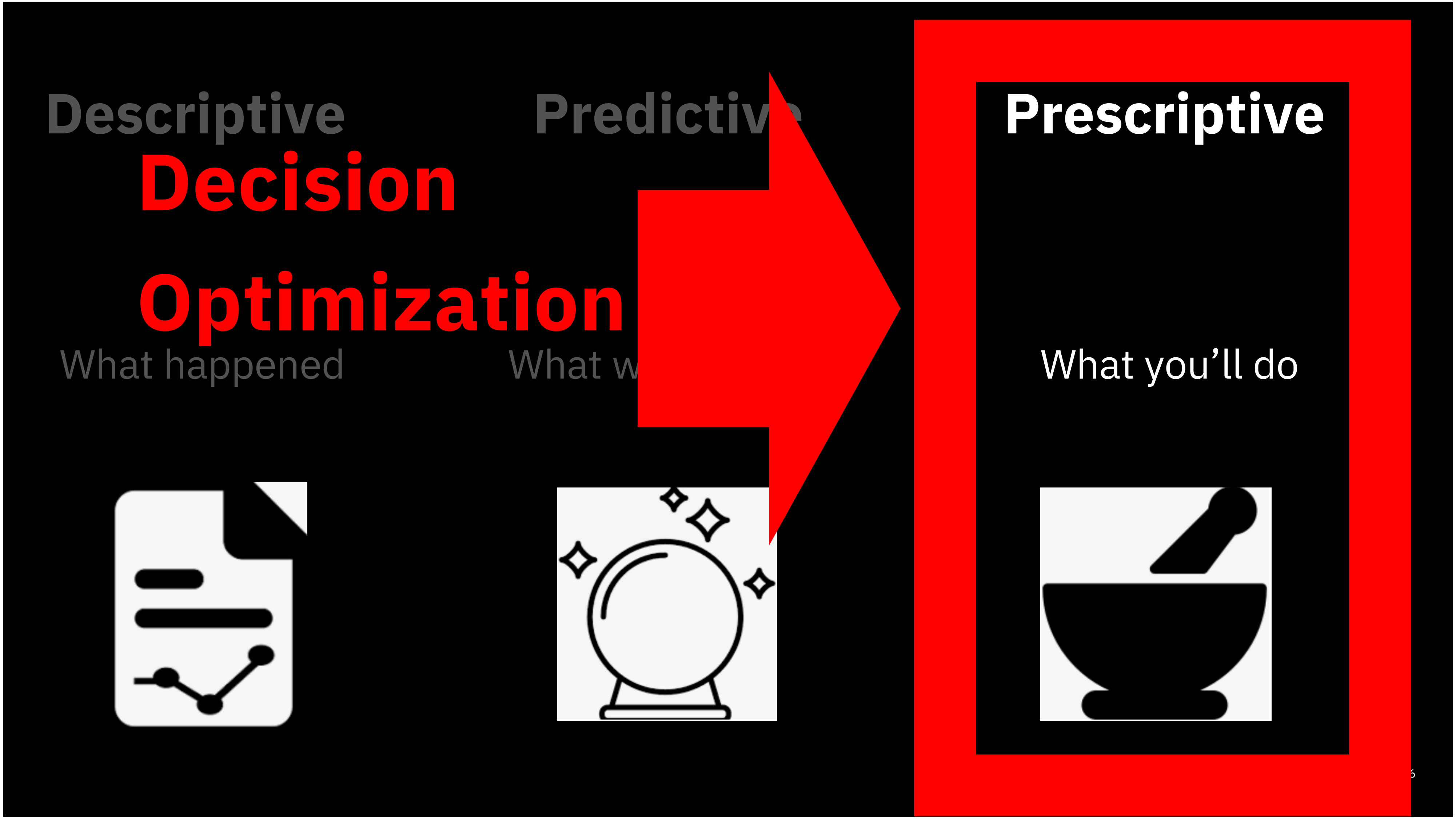
Optimization is hard:

- It tricks intuition
- It is hard to compute



10 factorial !!!

Descriptive	Predictive	Prescriptive
What happened	What will happen	What you'll do
		

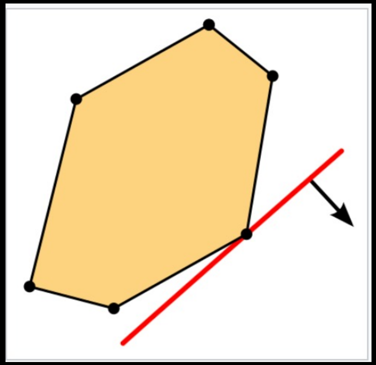

Descriptive
What happened
Predictive
What w
Decision
Optimization
Prescriptive
What you'll do

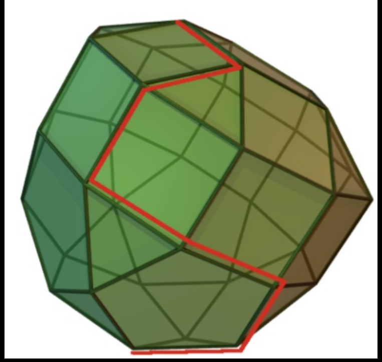
Basic concepts in Optimization

1# Linear Programming
("programming" in this context means way for solving mathematical problems).

2# Simplex algorithm

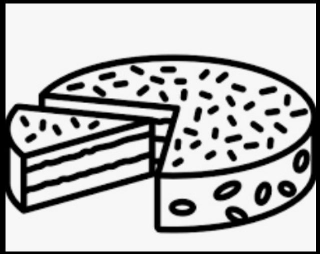
3# Linear Programming problem:
- Decision variables
- Constraints
- Objective function

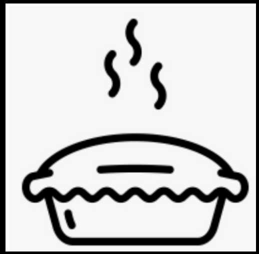




Lab 1

- minimize or maximize a linear objective
- subject to linear equalities and inequalities





Max is in a pie eating contest that lasts 1 hour.
Each torte that he eats takes 2 minutes.
Each apple pie that he eats takes 3 minutes.
He receives 4 points for each torte and 5 points for each pie.

What should Max eat to get the most points?

Source: MIT open course - https://ocw.mit.edu/courses/sloan-school-of-management/15-053-optimization-methods-in-management-science-spring-2013/lecture-notes/

Step 1. Determine the decision variables

- Let x be the number of tortes eaten by Max.
- Let y be the number of pies eaten by Max.

Step 2. Determine the objective function

Maximize $z = 4x + 5y$ (objective function)

Step 3. Determine constraints

subject to $2x + 3y \leq 60$ (constraint)

$x \geq 0$; $y \geq 0$ (non-negativity constraints)

Simplex algorithm

Solution.

A feasible solution satisfies all of the constraints.

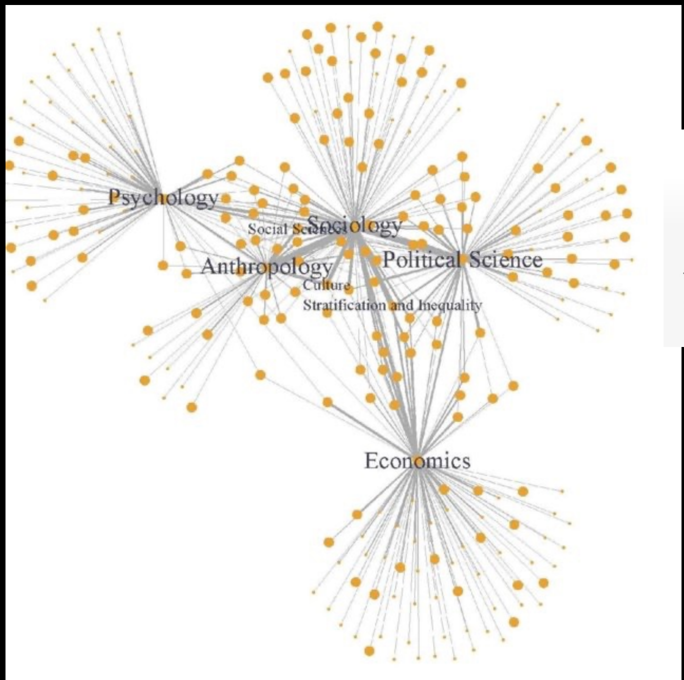
$x = 10$, $y = 10$ is feasible; $x = 10$, $y = 15$ is infeasible.

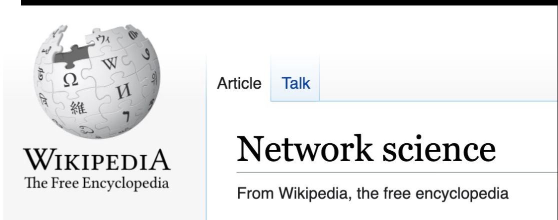
An optimal solution is the best feasible solution.

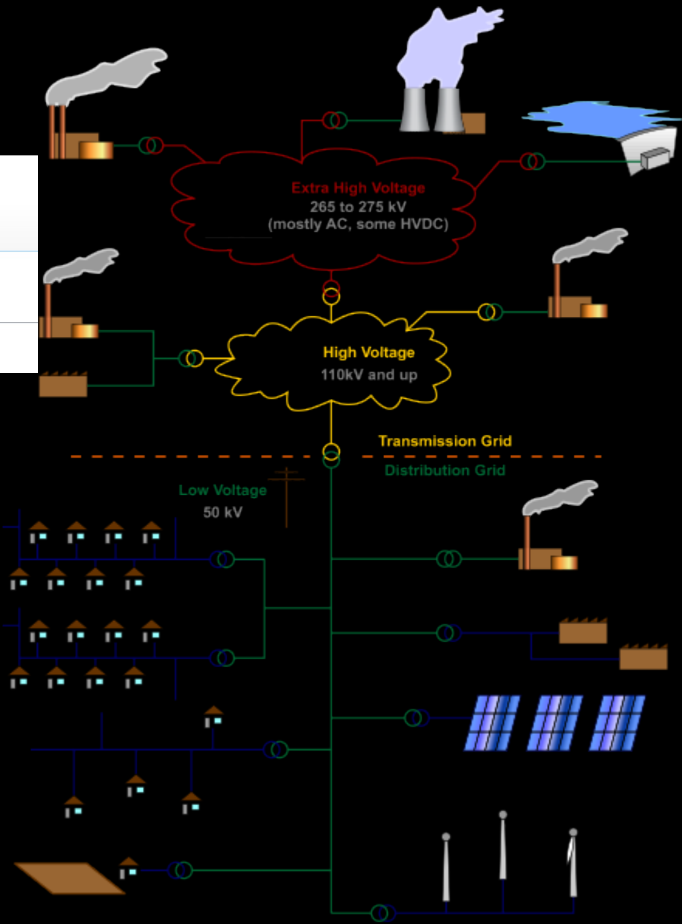
The optimal solution is $x = 30$, $y = 0$, $z = 120$

2 – solving network problems

Solving network problems is needed more than ever





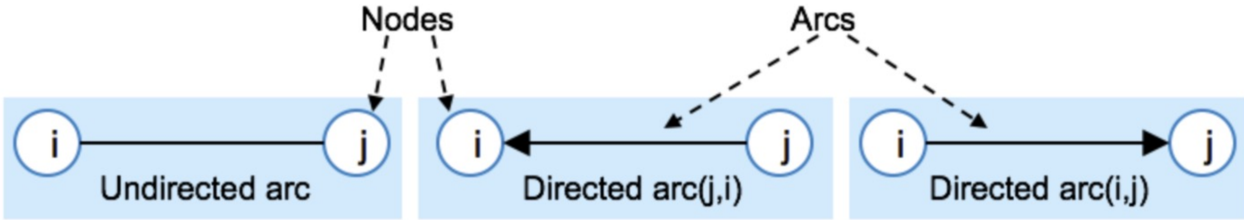


Network modeling concepts

Any network structure can be described using two types of objects:

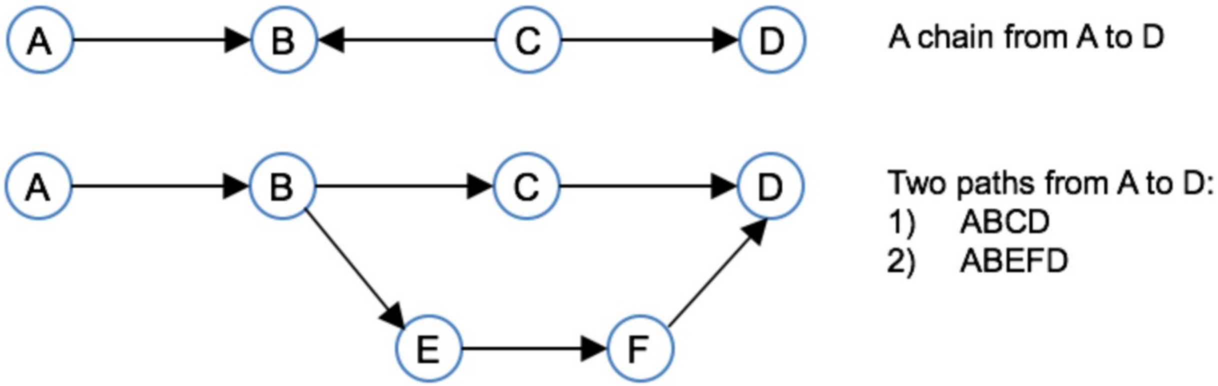
- Nodes: Defined points in the network, for example warehouses.
- Arcs: An arc connects two nodes, for example a road connecting two warehouses.

An arc can be *directed*, which means than an arc a_{ij} from node i to node j is different from arc $a_j i$ that begins at node j and ends at node i.



A sequence of arcs connecting two nodes is called a chain. Each arc in a chain shares exactly one node with the preceding arc.

When all the arcs in a chain are directed such that it is possible to traverse the chain in the directions of the arcs from the start node to the end node, it is called a path.



Different types of network problems

The following are some well-known types of network problems:

- Transportation problem
- Transshipment problem
- Assignment problem
- Shortest path problem
- Critical path analysis

Next, you'll learn how to recognize each of these, and how their special structure can be exploited.

The Transportation Problem

x (purple) – quantity of items to be transported

y (green) – demand for items

z (orange) – transportation cost/capacity

objective – minimize z (total transportation cost)

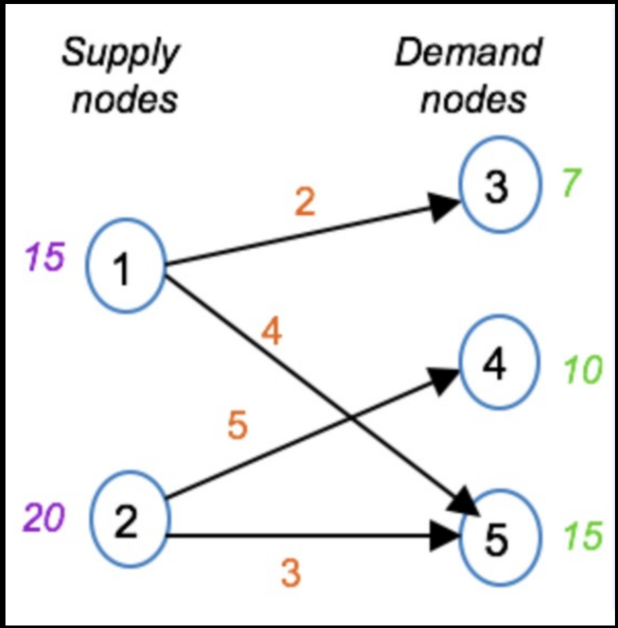
constraints:

1 – for each supply node:

total outbound flow < quantity

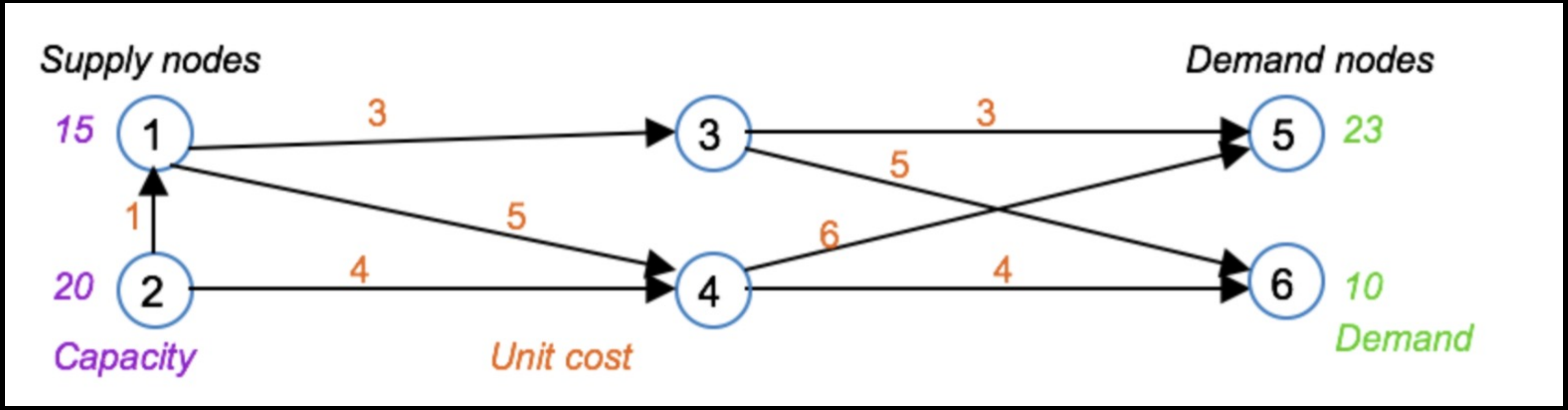
2 – for each demand node:

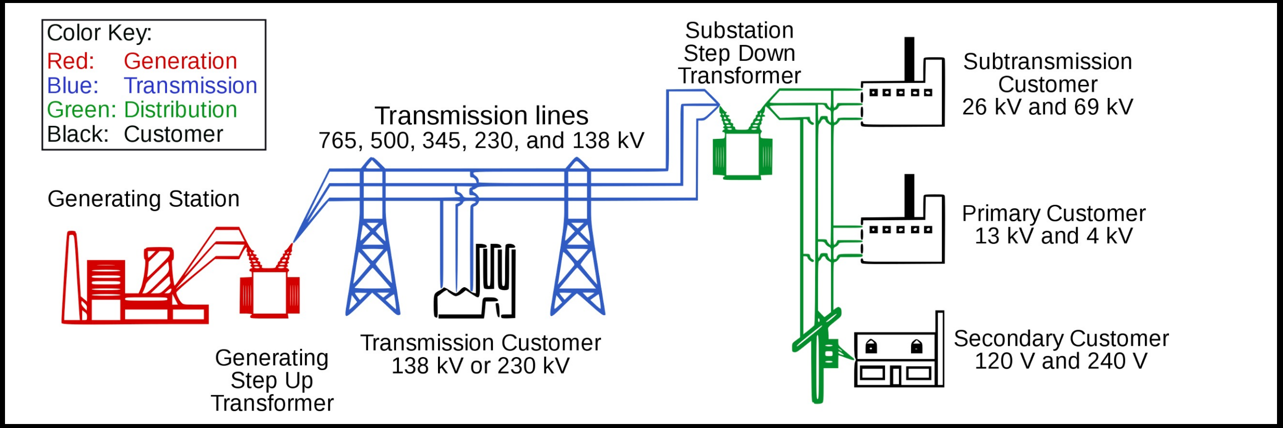
total inbound flow > demand



The Transshipment problem

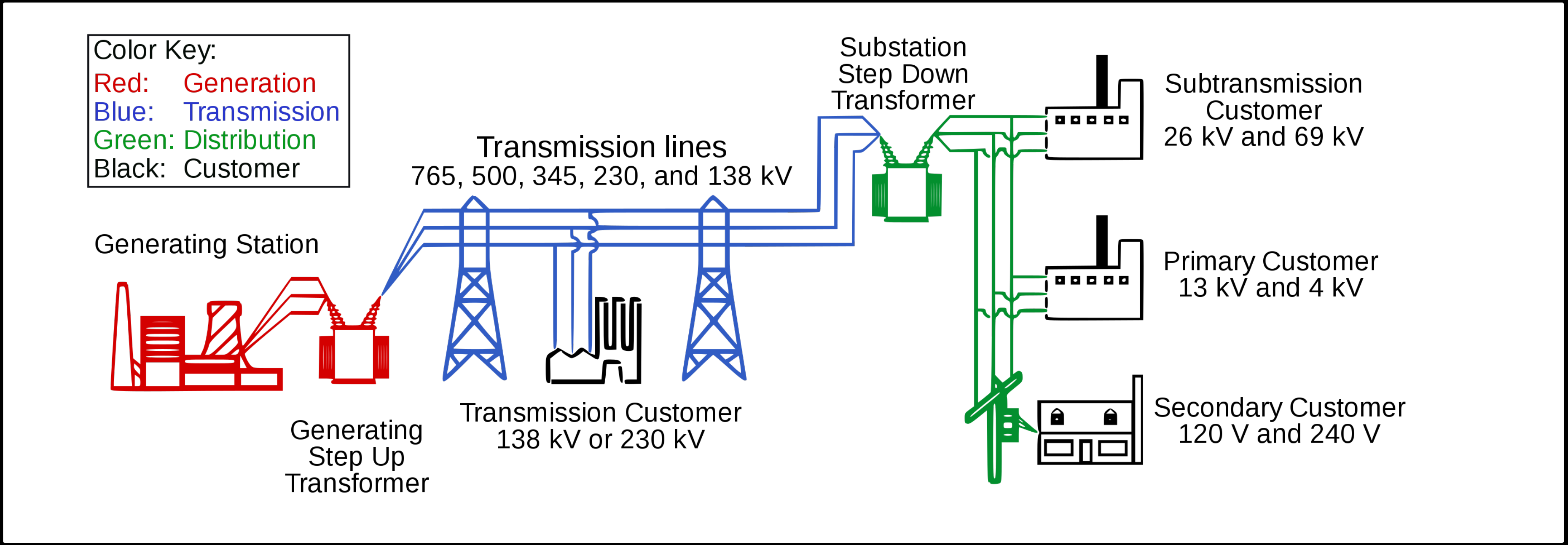
Same as Transportation Problem with some additional complexity – intermediate nodes





The Transshipment problem

Perfect example is electrical grid with its four types of nodes

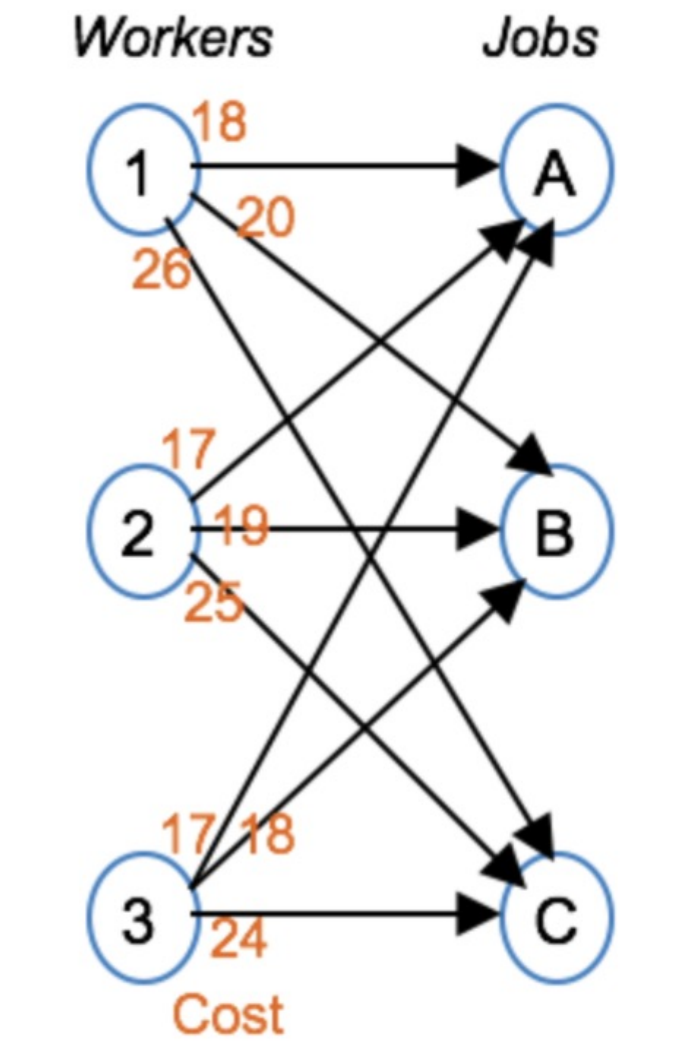


The Assigment problem

We assign one set of items to another, while optimizing a given objective.

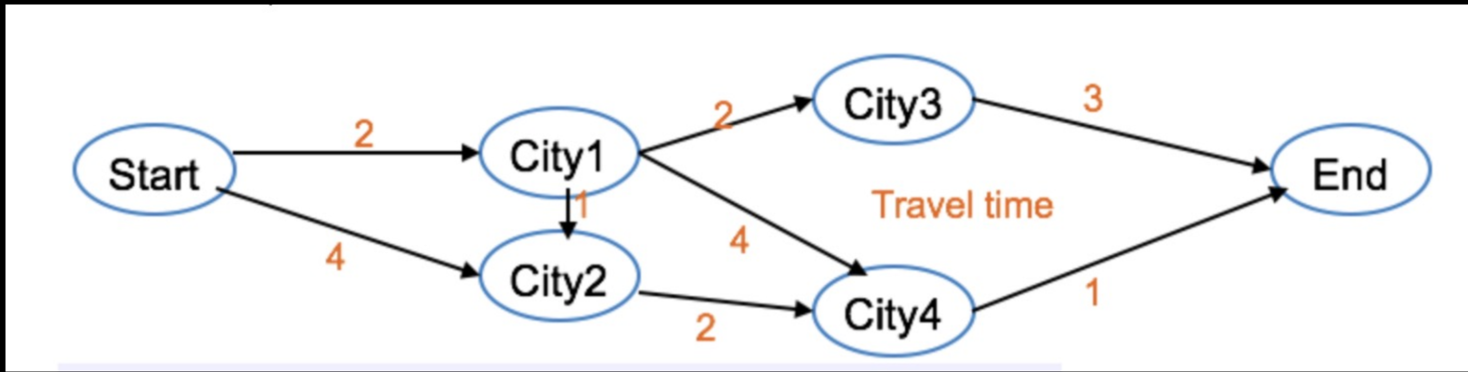
The cost of assigning a worker to a job is shown on each arc. The objective is to minimize the total assignment cost.

The general constraint here is that one worker can be assigned only to one job.



The Shortest Path Problem

We find a shortest path through a network



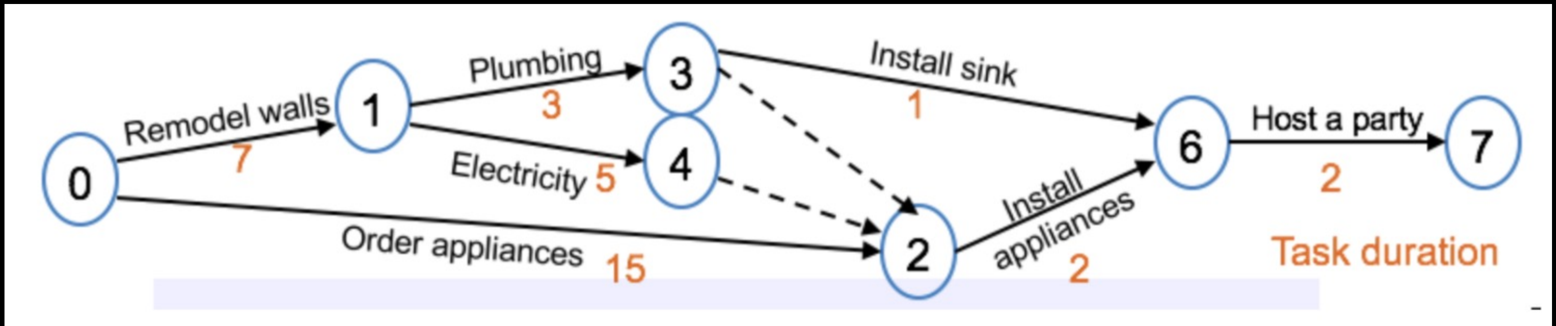
Number or arcs represent the travel time between cities.
The objective here is minimizing total travel time.
Constraints: one arc into each city, one arc out of each city.

Critical Path Analysis

Used f.e. in project planning. You find a set of critical activities where delay (in one of those) will cause overall project delay.

Tasks lying not on critical path may be delayed to a point wheb it becomes critical.

Example: Kitchen Remodeling Project



Arcs show the task duration in days, while the nodes show the task start time.
The objective here is minimizing total travel time.
Constraints: one arc into each city, one arc out of each city

CPLEX Network Optimizer

As you've now seen, many network problems are special types of LP problems. In many cases, using the Simplex or Dual-simplex Optimizers is the most efficient way to solve them. In some cases, specialized algorithms can solve such problems more efficiently.

CPLEX automatically invokes the Network Optimizer when it's likely that it would improve solution time compared to the other algorithms.

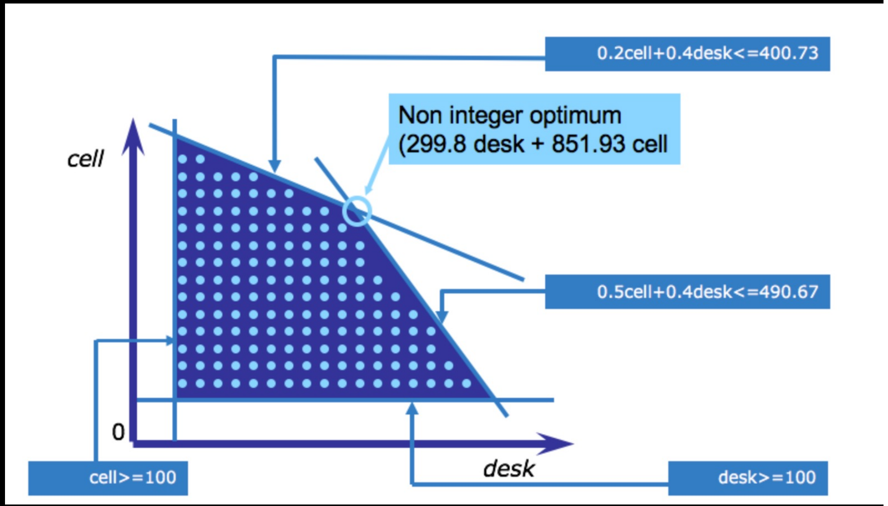
It is also possible to force the use (or not) of the Network Optimizer by setting the `lpopt` parameter of a DOcplex model to 3 (remember 1 was primal simplex, 2 was dual simplex, and 4 is for barrier).

3 – integer programming & relaxation

Integer Programming

If all of the unknown variables are required to be integers.

Complex IP Integer Programming cases are NP-hard or NP-complete



Mixed-Integer Programming

If only some of the unknown variables are required to be integers.

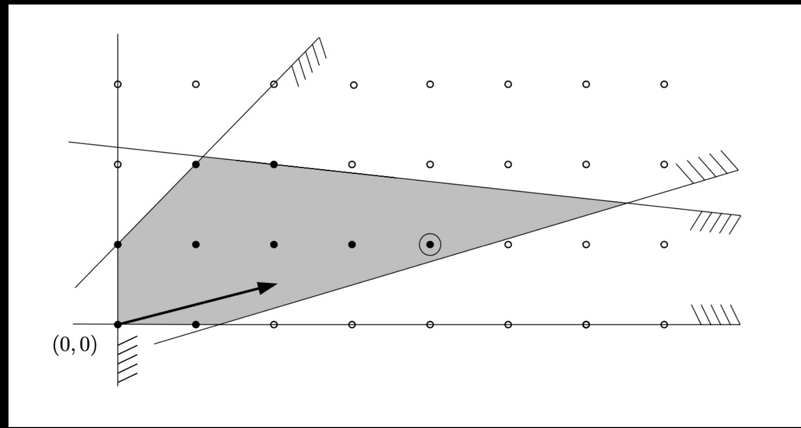
0-1 Integer Programming

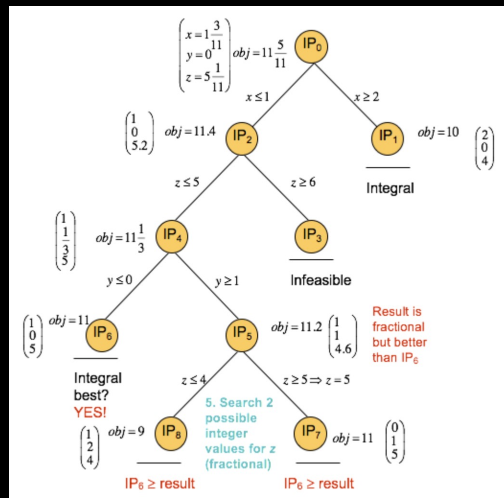
When integers take only bolean values.

How to deal with outcome of linear programming?

1# solve an LP problem and then round the fractional numbers in order to find an integer solution.

2# Relaxation (treating Integer Programming like Linear Programming) –
example is branch and bound method
(if all the variables take integer values, the solution is complete. If not, the algorithm begins a tree search)





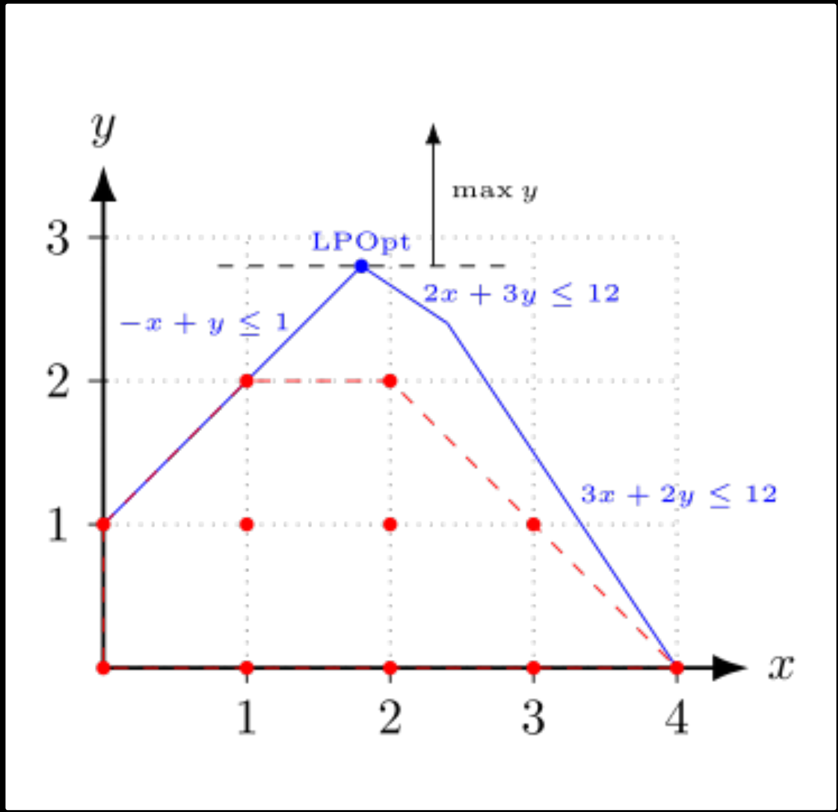
Why integer programs?

Advantages of restricting variables to take on integer values:

- More realistic
- More flexibility

Disadvantages:

- more difficult to model
- can be much more difficult to solve



Computation of Integer Programming problems

- Much, much harder than solving LPs
- Very good solvers can solve large problems – e.g., 50,000 columns 2 million non-zeros
- Hard to predict what will be solved quickly and what will take a long time.

Integer Programming is close to Combinatorial Optimization

Combinatorial optimization is a topic that consists of finding an optimal object from a finite set of objects. In many such problems, exhaustive search is not tractable. Set of feasible solutions is discrete or can be reduced to discrete

Applications:

- Logistics
- Supply chain optimization
- Developing the best airline network of spokes and destinations
- Deciding which taxis in a fleet to route to pick up fares
- Determining the optimal way to deliver packages
- Working out the best allocation of jobs to people
- Designing water distribution networks
- Earth Science problems

Demo 1

Network problem
IP problem

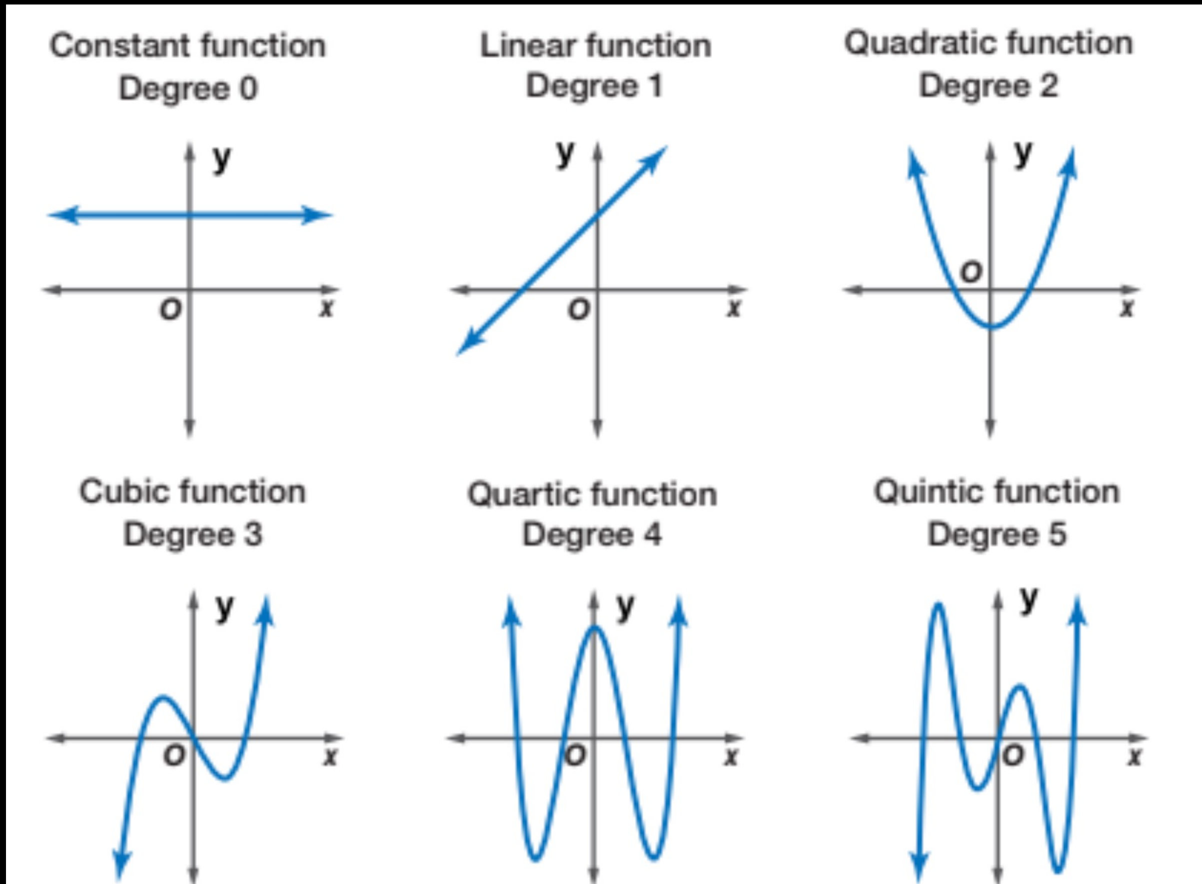
Demo 2

solving optimization problem with Watson Studio

optimization demo:
https://developer.ibm.com/tutorials/optimize-inventory-based-on-demand-with-decision-optimization/

this demo is a part o larger, 3-chapters demo:
https://developer.ibm.com/articles/develop-an-intelligent-inventory-and-distribution-strategy-using-ai/

Thank you


Constant function
Degree 0
y
O
x
Linear function
Degree 1
y
O
x
Quadratic function
Degree 2
y
O
x
Cubic function
Degree 3
y
O
x
Quartic function
Degree 4
y
O
x
Quintic function
Degree 5
y
O
x

How to deal with outcome of linear programming?

What is a Quadratic Program?

Quadratic Programs (or QPs) have quadratic objectives and linear constraints. A model that has quadratic functions in the constraints is a Quadratically Constrained Program (or QCP). The objective function of a QCP may be quadratic or linear.
A simple formulation of a QP is:

$$
\begin{gathered}
minimize\ \frac{1}{2} x^t Q x + c^t x \\
subject\ to \\
Ax \geq b \\
lb \leq x \leq ub
\end{gathered}
$$

Convexity

No straight lines means there are some quadratic equations that created them.

In such cases we are simply trying to find best fit by fitting the lines into to convex.

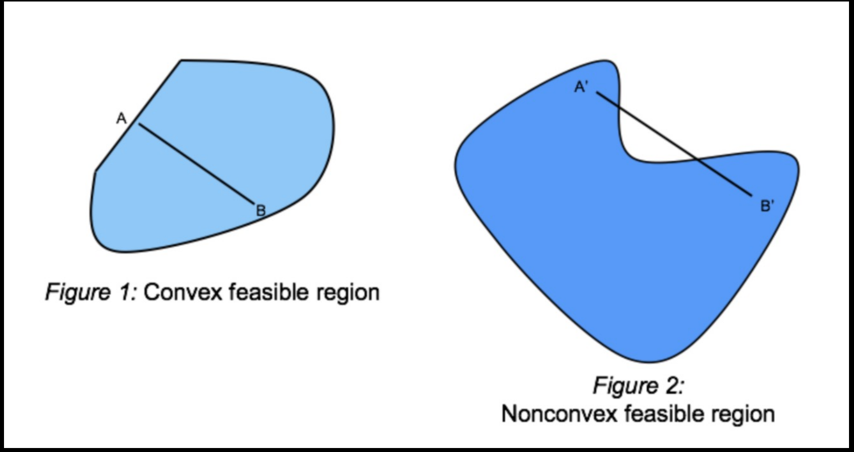


Figure 1: Convex feasible region

Figure 2:
Nonconvex feasible region