Integer Programming (IP)

The general linear mathematical programming problem

Mixed IP Problem - MIP

\[
\begin{align*}
\max & \quad c^T x + h^T y \\
\text{s.t.} & \quad Ax + Gy \leq b \\
& \quad x \in \mathbb{Z}_+^n, y \in \mathbb{R}_+^p
\end{align*}
\]

where

\( x \) vector of positive integer variables
\( y \) vector of positive real variables
**Integer Programming (IP)**

Problems with integer variables are said *Combinatorial Problems*

Examples of applications:

- **Production Scheduling**
  - Assign tasks to machines or production lines and determine their sequence on the machines

- **Logistics**
  - Determine the optimal routing for a fleet of vehicles in order to deliver or pickup goods at customer locations

- **Facility location**
  - Determine the best location for one or more distribution centres in order to optimally reach the customers

- **Network design (transportation and telecommunication)**

- **Planning of investments**
**Integer Programming (IP)**

- Use integer values if fractional values are not acceptable approximations (e.g., determine the yearly production of airplanes vs nail)
- Integer variables can model the use of indivisible resources or the selection among discrete alternatives
- When integer variables are restricted to binary values $\Rightarrow$ 0-1 LP
- Binary variables model the choice between two alternatives or the occurrence or not of events

\[
x = \begin{cases} 
1 & \text{the event occurs} \\
0 & \text{the event does not occur}
\end{cases}
\]

- Classical problems modelled with integer variables
  - Knapsack Problem
  - Matching/Assignment Problem
  - Fixed Charge Problem
  - Sequencing Problem
IP: knapsack problem

Possible definitions:

• Choose which objects carry in a knapsack for a campsite
• Select which goods to transport in a container

• Select which activities activate among a set of alternatives (e.g., projects to be funded):
  – $n$ possible projects to be funded with a maximum available budget $b$
  – If project $j$, $j=1,...,n$, is funded the required investment is $a_j$
  – Project $j$ provides a revenues equal to $c_j$
  – Each project cannot be partially funded: it may be completely funded or not funded

• The problem
  – Determine the best projects to be funded in order to maximize the total revenue without exceeding the available budget
**IP: knapsack problem**

- Knapsack is different from Product Mix (no fractional solution is allowed)
- Binary variables are needed

\[ x_j = \begin{cases} 
1 & j = 1, \ldots, n \\
0 & \text{otherwise}
\end{cases} \]

- Objective function: the total revenue for the funded project

\[ \max \sum_{j=1}^{n} c_j x_j \]
**IP: knapsack problem**

- Constraints impose that variables $x_j$ assume values consistent with their meaning
  - The available budget is not exceed (*knapsack constraint*)
    $$\sum_{j=1}^{n} a_jx_j \leq b$$
  - Variables $x_j$ assume binary values $x_j \in \{0,1\} = B \quad \forall j = 1,\ldots, n$

- The Knapsack problem formulation
  $$\max \sum_{j=1}^{n} c_jx_j$$
  $$\sum_{j=1}^{n} a_jx_j \leq b$$
  $$x_j \in \{0,1\} = B \quad \forall j = 1,\ldots, n$$
IP: knapsack problem

The multi-dimensional KP

- The realization of the projects require funding over \( m \) periods (months)
  - \( a_{ij} \) the fund needed for project \( j \) in period \( i, j=1,...,n, i=1,...,m \)
  - \( b_i \) available budget for period \( i, i=1,...,m \)

\[
\text{max} \sum_{j=1}^{n} c_j x_j \\
\sum_{j=1}^{n} a_{ij} x_j \leq b_i \quad \forall i = 1,...,m \\
x_j \in \{0,1\} \quad \forall j = 1,...,n
\]
IP: assignment problem

• Definition
  – Assign \( m \) activities to \( n \) workers (\( n \geq m \))
  – Each worker can at most perform a single activity
  – The cost for assigning to worker \( j, j=1,\ldots,n \), activity \( i, i=1,\ldots,m \), is \( c_{ij} \)

The problem
  – Determine the assignment of all the activities to the workers such that the total cost is minimized
Graph representation

Activities ($i$)  $c_{ij}$  Workers ($j$)

The problem:
• Determine which links to introduce between activity and worker nodes such that each activity node is exactly paired to a worker node
IP: assignment problem

Variables:
• $n \times m$ binary assignment variables (associated with links)
  
$$x_{ij} = \begin{cases} 
1 & \Rightarrow \text{Activity } i \text{ assigned to worker } j \quad i = 1, \ldots, m \\
0 & \Rightarrow \text{Activity } i \text{ not assigned to worker } j \quad j = 1, \ldots, n 
\end{cases}$$

Objective:
• Minimize the assignment cost

$$\min \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij}$$
IP: assignment problem

Constraints:

– Each activity must be assigned

\[ \sum_{j=1}^{n} x_{ij} = 1 \quad \forall i = 1, \ldots, m \]

– Each worker can at most perform a single activity

\[ \sum_{i=1}^{m} x_{ij} \leq 1 \quad \forall j = 1, \ldots, n \]

– Variables are binary to ensure that an activity is entirely assigned to a single worker

\[ x_{ij} \in \{0, 1\} = B \quad \forall i = 1, \ldots, m \quad \forall j = 1, \ldots, n \]
IP: assignment problem

The assignment problem formulation

\[
\begin{align*}
\text{min} & \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij} \\
\sum_{j=1}^{n} x_{ij} &= 1 \quad \forall i = 1, \ldots, m \\
\sum_{i=1}^{m} x_{ij} &\leq 1 \quad \forall j = 1, \ldots, n \\
x_{ji} &\in \{0, 1\} = B \quad \forall i = 1, \ldots, m \quad \forall j = 1, \ldots, n
\end{align*}
\]
Transportation problem with fixed charge

A modified version of the transportation problem:

– Transportation of a good (e.g., gas or oil) between \( m \) suppliers and \( n \) customers
– Each supplier connected to each customer by a transportation channel (e.g., pipeline)
– A flow between a supplier \( i, i=1,\ldots,m \), and a customer \( j=1,\ldots,n \), can be established if:
  • A fixed charge \( f_{ij} \) is paid for renting the channel between \( i \) and \( j \)
  • A transportation cost \( c_{ij} \) is paid for each unit of good transported from \( i \) to \( j \)
– The maximum availability of supplier \( i \) is \( s_i, i=1,\ldots,m \).
– The demand of customer \( j \) is \( r_j, j=1,\ldots,n \).

The problem

– Determine which channels must be rented and the flow in them so that the customers’ demand is satisfied at the minimum total cost
Transportation problem with fixed charge

The formulation extends that of transportation problem

- \( m \times n \) continuous variables for the flow between \( i \) and \( j \)
  \[ x_{ij} \in \mathbb{R}, \quad i=1,...,m \quad j=1,...,n \]

- Binary variables are introduced associated with each channel \((i,j)\)
  \[
y_{ij} = \begin{cases} 
    1 &\quad \text{if } (i,j) \text{ is used } \quad i = 1,...,m \\
    0 &\quad \text{if } (i,j) \text{ is not used } \quad j = 1,...,n
  \end{cases}
\]
Transportation problem with fixed charge

The cost for each channel has a discontinuity

The objective function is non linear but it can be still linearly modelled
Transportation problem with fixed charge

- **Objective function**
  - It includes both marginal and fixed cost
    \[
    \min \sum_{i=1}^{m} \sum_{j=1}^{n} (c_{ij}x_{ij} + f_{ij}y_{ij})
    \]

- **Transportation problem constraints**
  - Supplier capacities
    \[
    \sum_{j=1}^{n} x_{ij} \leq s_i \quad \forall \ i = 1, \ldots, m
    \]
  - Customer demands
    \[
    \sum_{i=1}^{m} x_{ij} = r_j \quad \forall \ j = 1, \ldots, n
    \]

- **Variables** \(x_{ij}\) and \(y_{ij}\) must be linked: transportation is allowed only on active (rented) channels
  \[
  x_{ij} \leq M y_{ij} \quad \forall \ i, j
  \] (fixed charge constraints)

\(M\) is a Big-M
Transportation problem with fixed charge

• If a maximum channel capacity $q_{ij}$ is given for the channels we can use $q_{ij}$ instead of $M$

$$x_{ij} \leq q_{ij} y_{ij} \quad \forall i, j$$

• Definitions of variables

$$x_{ij} \in \mathbb{R}_+ \quad y_{ij} \in \mathbb{B} \quad \forall i = 1, \ldots, m \quad \forall j = 1, \ldots, n$$
Transportation problem with fixed charge

The problem formulation

\[
\min \sum_{i=1}^{m} \sum_{j=1}^{n} (c_{ij} x_{ij} + f_{ij} y_{ij})
\]

\[
\sum_{j=1}^{n} x_{ij} \leq s_i \quad \forall i = 1, \ldots, m
\]

\[
\sum_{i=1}^{m} x_{ij} = r_j \quad \forall j = 1, \ldots, n
\]

\[
x_{ij} \leq M y_{ij} \quad \forall i, j
\]

\[
x_{ij} \in \mathbb{R}_+ \quad y_{ij} \in \mathbb{B} \quad \forall i = 1, \ldots, m \quad \forall j = 1, \ldots, n
\]
Sequencing problem

A common problem in detailed planning of production activities:

• Determine the optimal sequence for a set of operations (jobs or tasks) of a productive resource (machine):
  – \( n \) independent jobs must be sequenced on a single machine
  – The machine can process a single job at a time
  – Job processing cannot be suspended and restarted (no preemption is allowed)
  – The processing time of job \( i, i=1,\ldots,n \), is \( p_i \)

• The objective is a cost usually associated with job completion time, e.g.:
  – Minimize the average job completion time
  – Minimize the maximum job completion time (makespan)
  – Minimize the sum of delays w.r.t. jobs’ due dates

• No assumption on the cost: we focus on constraints only
Sequencing problem

Variables:
• Continuous variables for the start time of jobs
  \[ t_i \in \mathbb{R}, \quad i=1,\ldots,n \quad t_i \geq 0 \]

We must model the fact that two jobs cannot overlap:
• Given two jobs \( i \) and \( j \) one of the following alternative constraints must hold

1. if \( i \) precedes \( j \) \( \Rightarrow \) \( t_j \geq t_i + p_i \)

2. if \( j \) precedes \( i \) \( \Rightarrow \) \( t_i \geq t_j + p_j \)
Sequencing problem

Only one constraint can hold: disjunctive constraints

– According to the sequence only one of the two disjunctive constraints must be active (and the other one must be not binding)

Disjunctive constraints are modelled with binary (sequencing) variables

\[ y_{ij} \in \mathbb{B} \quad i = 1, \ldots, n \quad j = 1, \ldots, n \quad y_{ij} = \begin{cases} 1 & \Rightarrow \text{i precedes j} \\ 0 & \Rightarrow \text{i not precede j} \end{cases} \]

For each unordered pair of job, \( i, j, (i<j) \) the disjunctive constraints are

\[
\begin{align*}
  t_j & \geq t_i + p_i - M (1 - y_{ij}) \\
  t_i & \geq t_j + p_j - M y_{ij}
\end{align*}
\]

where \( M >> \sum_{i=1}^{n} p_i \)

Variables \( y_{ij} \) give the sequencing order of jobs
IP methods - introduction

- Solving an IP problem could seem easier than a LP one: IP polyhedra include a finite number of solutions.
- However, let consider a 2D example ...

\[ X = \{ x \in \mathbb{R}^n : A x \leq b \} \]

LP optimum is a vertex

\[ X = \{ x \in \mathbb{Z}^n : A x \leq b \} \]

Often IP optimum is an internal point
IP methods - introduction

• Very often Simplex Alg. does not generate a feasible optimal solution to IP problems

• From the 2D example it could seem easy to find the IP optimum from the LP one:
  – By approximating the LP optimum
  – Finding the integer point closest to the LP optimum

• Both such strategies can fail ....
IP methods - introduction

- Approximated solutions may be not integer feasible
IP methods - introduction

- The closest integer solution may be not optimal
IP methods - introduction

- A serious difficulty: when the problem dimension (number of variables) is not small, even if finite, the number of solutions can be huge.

- Binary problems (0-1 LP):
  - The number of possible solutions double for each added variable.
  - It grows as $2^n$ where $n$ is the number of variables.
  - Example:
    - $n=4 \Rightarrow 16$ possible combinations (solutions)
    - $n=10 \Rightarrow 1024$
    - $n=30 \Rightarrow 1.07 \times 10^9$

- Integer problems: even worse situation.
  Example: 10 variables that can assume 10 integer values: the number of combinations is $10^{10}$