Integer Programming

An LP problem in which the variables are restricted to be integers is known as an integer programming problem (IP). If only some of the variables are integer, it is a mixed integer program (MIP). IP's and MIP's are much more difficult to solve than LP's.

An IP in which all the variables are 0 or 1 (binary) is a 0-1 IP. These are surprisingly common.

Example IP – Cloth Co

Cloth Co manufactures shirts, shorts, and pants. Each of these lines needs a special machine, which has to be rented. Of courses the machine need not be rented if none of the line is produced. In addition each line needs labour and cloth as shown below.

	Requirements per item				**Machine cost**
Item	**Labour (hrs)**	**Cloth (m^2)**	**Price**	**Cost**	**Rent ($/week)**
Shirts	3	4	12	6	$200
Shorts	2	3	8	4	$150
Pants	6	4	15	8	$100
Resource available	150	160			

What is the optimal production per week?

Answer: As well as the usual decision variables, $x_1, x_2, x_3 =$ number of shirts, shorts and pants produced per week, introduce the binary decision variables

$$y_1 = \begin{cases} 1 & \textit{if any shirts are produced} \\ 0 & \textit{if no shirts are produced} \end{cases}$$
$$y_2 = \begin{cases} 1 & \textit{if any shorts are produced} \\ 0 & \textit{if no shorts are produced} \end{cases}$$
$$y_3 = \begin{cases} 1 & \textit{if any pants are produced} \\ 0 & \textit{if no pants are produced} \end{cases}$$

The profit is: $z = 6x_1 + 4x_2 + 7x_3 - 200y_1 - 150y_2 - 100y_3$

The negative coefficients on the y's means this would always give $y_1 = y_2 = y_3 = 0$. So we must force $y_i = 1$ if $x_i > 0$. We can do this by calculating the maximum number of each item that could be produced, subject to the resource constraints (for shirts this is 40) and adding constraints of the form: $x_1 \leq 40y_1$. If $y_1 = 1$ this does not impose any additional restriction on x_1. If $y_1 = 0$ then x_1 must also be zero.

The full formulation is thus:

$$\max \quad z = 6x_1 + 4x_2 + 7x_3 - 200y_1 - 150y_2 - 100y_3$$

Subject to:

$$\begin{aligned} 3x_1 + 2x_2 + 6x_3 &\leq 150 \\ 4x_1 + 3x_2 + 4x_3 &\leq 160 \\ x_1 &\leq 40y_1 \\ x_2 &\leq 53y_2 \\ x_3 &\leq 25y_3 \end{aligned}$$

$$x_i \geq 0, x_i \text{ integer}, y_i = 0 \text{ or } 1, i = 1, 2, 3$$

Example in python.

This is an example of a fixed charge problem.

Formulations – some revision

When solving a business problem in practice, we split the formulation and the data, unlike this formulation which combines the formulation and the data. What would you consider is data in the previous formulation?

First we would formulate the problem in ideas/words – perhaps not written down.

Maximise the profit: profit for items produced, less rental cost

Subject to

Limits on the available resources and

A product is not produced unless the appropriate machine is rented.

In this informal formulation we have not even defined variables!

For academic papers we would present the formulation in one of a number of ways.

Data	
R	Set of resources
P	Set of product types
v_r	Available amount of each resource $r \in R$ in each period
u_{rp}	Amount of resource r used to produce one unit of product type p
M_p	Maximum amount of product type p that could possibly be produced $= \min_{r \in R}(v_r/u_{rp})$
c_p	Profit per unit product type p
f_p	Cost per period of renting machine that produces product type p
Variables	
x_p	Units of product type p produced in each period
y_p	Binary indicator variable, set to 1 if the machine that produces product type p is rented

$$\max \quad z = \sum_{p \in P} c_p x_p - \sum_{p \in P} f_p y_p$$

Subject to:

$$\begin{aligned} \sum_{p \in P} u_{rp} x_p &\leq v_r & \forall\, r \in R \\ x_p &\leq M_p y_p & \forall\, p \in P \end{aligned}$$

$$x_p \geq 0, x_p \text{ integer}, y_p = 0 \text{ or } 1, p \in P$$

Alternative formulations could sum over ranges rather than over sets or we could use vector and matrix notation. However, as the problem gets more complicated, especially in the presence of sparse matrices, summing over carefully defined sets can keep the formulation compact.

Definition: The LP obtained by relaxing integrality constraints – but not the range of the variables – is called the **LP relaxation** of the IP or MIP.

For a maximisation problem: optimal z-value for the LP relaxation ≥ optimal z-value for the IP (and vice versa for minimisation).

Set Covering

Kilroy council needs to build fire stations to service its six towns. It wants to build the minimum number of fire stations, yet ensure that each town is within 15 minutes of a fire station. For each city, they know all the cities that are within 15 minutes:

Town	Within 15 minutes
1	1,2
2	1,2,6
3	3,4
4	3,4,5
5	4,5,6
6	2,5,6

The decision variables will be x_i which will be 1 if a station is built at Town i.

The formulation is:

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

Subject to:

$$\begin{aligned}
x_1 + x_2 \quad\quad\quad\quad\quad\quad\quad\quad &\geq 1 \\
x_1 + x_2 \quad\quad\quad\quad\quad\quad + x_6 &\geq 1 \\
x_3 + x_4 \quad\quad\quad\quad &\geq 1 \\
x_3 + x_4 + x_5 \quad\quad &\geq 1 \\
x_4 + x_5 + x_6 &\geq 1 \\
x_2 + \quad\quad\quad\quad x_5 + x_6 &\geq 1 \\
x_1, x_2, x_3, x_4, x_5, x_6 \in \{0,1\}
\end{aligned}$$

A more general formulation for set covering problems is:

Data	
I	Set of covering sets
J	Set of objects to be covered
c_i	Cost of set i
a_{ij}	Set to 1 if set i contains object j
Variables	
x_i	Binary variable, set to 1 if set i is used in the solution.

$$\min \quad z = \sum_{i \in I} c_i x_i$$

Subject to:

$$\sum a_{ij} x_i \quad \geq \quad 1 \quad \forall\, j \in J$$

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

The trouble with IP's

Consider the IP:

$$\max \quad z = 21x_1 + 11x_2$$

Subject to:

$$7x_1 + 4x_2 \quad \leq \quad 13$$

$$x_1, x_2 \geq 0; \; x_1, x_2 \text{ integer}$$

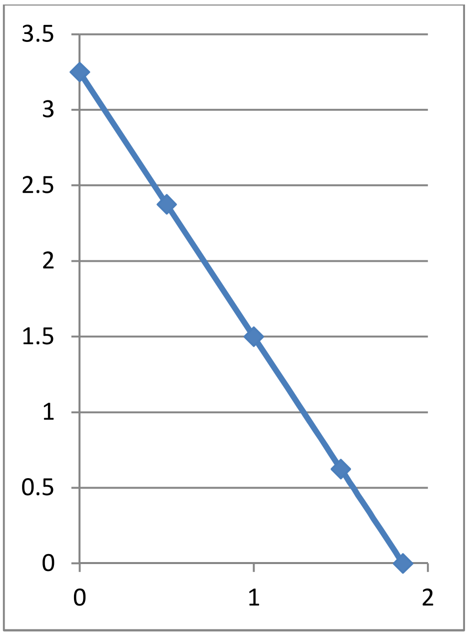

Figure 1

$$7x_1 + 4x_2 = 13$$

From figure 1 we can see that the feasible solutions of the problem are the following points (0,0), (0,1), (0,2), (0,3), (1,0) and (1,1). By computing the objective value and comparing, we see that the optimum solution is $z = 33$, $x_1 = 0, x_2 = 3$. Clearly an IP has a finite number of solutions, but as the number of variables increases this number of solutions increases exponentially. We need some smart way to search these.

Suppose we suggest taking the LP relaxation and rounding it. The LP solution to this problem is: $x_1 = \frac{13}{7}, x_2 = 0$. Rounding this solution yields the solution $x_1 = 2, x_2 = 0$, which is infeasible for the IP. If we round downward, we get a feasible solution, but it is a long way from the optimal solution. It is possible to produce an IP so that every rounding of the LP relaxation is infeasible.

Branch and Bound

The branch and bound method solves IP's by solving a series of LP's. The first LP is the linear relaxation of the IP. At each stage we find variables which are not integer and create two sub-problems – one where the variable in question is at least rounded up and one where it is at most rounded down.

More formally we want to solve the IP:

$$\max z = c^T x$$

Subject to:

$$Ax = b$$
$$x \geq 0$$
$$x \text{ integer}$$

0. Initialise: Let problem p be linear relaxation of IP. Set the best solution found so far: $p^* = null,\ z^* = -\infty$. Initialise the set of pending problems: $P = \{p\}$.
1. Begin Main Loop: If P is empty, then **stop**. Otherwise, pick a pending problem $p' \in P$ and remove it from P. Solve p' (an LP) and denote the objective value as z'.
2. Fathomed sub-problem: If $z' \leq z^*$ then go to step 1.
3. Update incumbent solution: If the solution of p' is integer then set $p^* = p',\ z^* = z'$ and go to step 1.
4. Branch: Choose x_i and integer $\overline{x_i}$ such that $\overline{x_i} < x_i < \overline{x_i} + 1$. Create a new problem p^1 by adding the constraint $x_i \leq \overline{x_i}$ to p'. Create a new problem p^2 by adding the constraint $x_i \geq \overline{x_i} + 1$ to p'. Set $P = P \cup \{p^1, p^2\}$.
5. Go to Step 1.

The critical steps in the process are:

- Step 1: Which pending problem do we pick?
- Step 4: Which variable do we pick?

Branch and Bound Example:

Telfa Pty. Ltd. makes tables and chairs. Each of these items needs labour and wood. The unit profit and resources needed for each item are:

	Table	Chairs	Available
Profit($)	8	5	
Labour (hrs)	1	1	6
Wood	9	5	45

Telfa need to decide how many tables and how many chairs to make.

Knapsack Problem

A knapsack problem is a problem with one constraint and positive coefficients in the objective function and the constraint.

Example: We need to pack a knapsack with volume 17 litres to carry as much weight as possible. We may choose any number of the following three objects.

	Object 1	Object 2	Object 3
Weight	$\frac{1}{5}$	$\frac{1}{3}$	1
Volume	3	5	9
Density	$\frac{1}{15}$	$\frac{1}{15}$	$\frac{1}{9}$

Using the obvious variable notation we get:

$$\max \frac{1}{5}x_1 + \frac{1}{3}x_2 + x_3$$
$$\text{s.t. } 3x_1 + 5x_2 + 9x_3 \leq 17$$
$$x_1, x_2, x_3 \geq 0 \text{ and integer}$$

In any branch and bound solution to such a problem, we branch on the densest item first. Say $x_i = n + \theta, 0 < \theta < 1$. It cannot be possible to have $n+1$ items (otherwise the LP solution would have used this many), so we add the two branches: $x_i = n$ and $x_i \leq n - 1$. This is a specialist form of branching that works well for knapsack problems. With this approach, a knapsack problem can be easily solved by a branch and bound algorithm without the use of an LP solver.

Formulation Issues

Sometimes the formulation is more important than any cuts the solver may be able to add. Consider the following facility location problem.

The distance between all pairs of N towns is given by a distance matrix. We wish to position M ($M < N$) facilities at towns so as to minimise the sum of the distances from each town to its nearest facility.

Let d_{ij} be the distance between towns i and j. Define the variables y_j which will be 1 if a facility is located at town j, 0 otherwise, and the variables x_{ij} which will be 1 if town i utilises the facility in town j. Then the problem can be written as:

$$\min \sum_{i,j=1..N} d_{ij} x_{ij}$$

Subject to:

$$\sum_{i=1..N} x_{ij} \leq N y_j \quad \forall\, j$$

$$\sum_{j} x_{ij} = 1 \quad \forall\ i$$

$$\sum_{j} y_j \leq M$$

$$x_{ij}, y_j \in \{0,1\}$$

The problem can be reformulated as:

$$\min \sum_{i,j=1..N} d_{ij} x_{ij}$$

Subject to:

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

$$\sum_{j} x_{ij} = 1 \quad \forall\ i$$

$$\sum_{j} y_j \leq M$$

$$x_{ij}, y_j \in \{0,1\}$$

This formulation has many more constraints, but for some solvers it is MUCH faster. It is straightforward to prove the second formulation is “tighter” than the first formulation. That is, every solution to the LP relaxation of the second formulation is also a solution to the LP relaxation of the first formulation, but the reverse does not apply.