Set Covering

To illustrate this model, consider the following *location* problem: A city is reviewing the location of its fire stations. The city is made up of a number of neighborhoods, as illustrated in Figure 1.

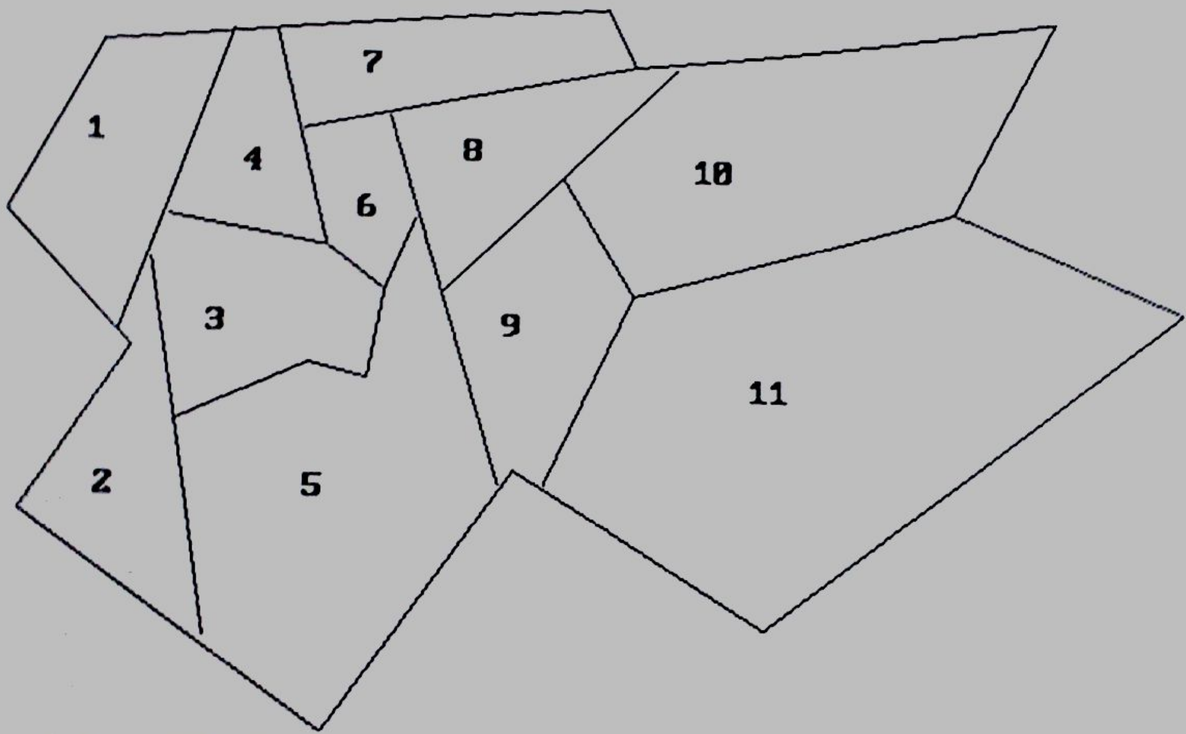

Figure 1: Map of the city

A fire station can be placed in any neighborhood. It is able to handle the fires for both its neighborhood and any adjacent neighborhood (any neighborhood with a non-zero border with its home neighborhood). The objective is to minimize the number of fire stations used.

We can create one variable x_j for each neighborhood *j*. This variable will be 1 if we place a station in the neighborhood, and will be 0 otherwise. This leads to the following formulation

Generated by CamScanner from intsig.com

$$a_{ij} = 1 \text{ if } i \in S_j$$
$$= 0 \text{ otherwise}$$

Let x_j be $0 - 1$ variables with $x_j = 1(0)$ to mean set S_j is included (respectively not included) in the cover. The problem is to

$$\text{Minimise} \quad \sum_{j=1}^{n} c_j x_j$$

$$\text{subject to} \sum_{j=1}^{n} a_{ij} x_j \geq 1 \qquad i = 1,\ldots m \tag{2}$$

$$x_j = 0 \text{ or } 1$$

The m inequality constraints have the following significance: since $x_j = 0$ or 1 and the coefficients a_{ij} are also 0 or 1 we see that $\sum_{j=1}^{n} a_{ij} x_j$ can be zero only if $x_j = 0$ for all j such that $a_{ij} = 1$. In other words only if no set S_j is chosen such that $i \in S_j$. The inequalities are put in to avoid this.

EXAMPLE 3: SET COVERING - AIRLINE CREW SCHEDULING

Consider the following simplified airline crew scheduling problem. An airline has m scheduled flight-legs per week in its current service. A flight-leg being a single flight flown by a single crew e.g. London - Paris leaving Heathrow at 10.30 am. Let S_j, $j = 1,\ldots n$ be the collection of all possible weekly sets of flight-legs that can be flown by a single crew. Such a subset must take account of restrictions like a crew arriving in Paris at 11.30 am. cannot take a flight out of New York at 12.00 pm. and so if c_j is the cost of set S_j of flight-legs then the problem of minimising cost subject to covering all flight-legs is a set covering problem. Note that if crews are not allowed to be passengers on a flight i.e. so that they can be flown to their next flight, then we have to make (2) an equality – the set partitioning problem.

EXAMPLE 4: SET COVERING - BUILDING FIRE STATIONS

There are six cities in region R. The region must determine where to build fire stations. The region wants to build the minimum number of fire stations and ensure that at least one fire station is near 15 minutes of each city. The times (in minutes) required to drive between cities are:

From	To					
	1	2	3	4	5	6
1	0	10	20	30	30	20
2	10	0	25	35	20	10
3	20	25	0	15	30	20
4	30	35	15	0	15	25
5	20	20	30	15	0	14
6	20	10	20	25	14	0

$$x_i = \begin{cases} 1 \text{ if a fire station is built in city i} \\ 0 \text{ otherwise} \end{cases}$$

Objective function: Total number of fire stations to be built

$$x_0 = x_1 + x_2 + x_3 + x_4 + x_5 + x_6$$

Constraints: A fire station within 15 mins of each city.

The locations that can reach each city in 15 minutes:

City 1   1, 2 $\Rightarrow x_1 + x_2 \geq 1$ (City 1 constraint)

Generated by CamScanner from intsig.com

City 2	1, 2, 6	$\Rightarrow x_1 + x_2 + x_6 \geq 1$ (City 2 constraint)
City 3	3, 4	
City 4	3, 4, 5	
City 5	4, 5, 6	
City 6	2, 5, 6	

$$
\begin{array}{llcl}
\text{IP: min} & x_0 = & x_1 + x_2 + x_3 + x_4 + x_5 + x_6 & \\
\text{subject to} & & x_1 + x_2 & \geq 1 \text{ (City 1)} \\
& & x_1 + x_2 + x_6 & \geq 1 \text{ (City 2)} \\
& & x_3 + x_4 & \geq 1 \text{ (City 3)} \\
& & x_3 + x_4 + x_5 & \geq 1 \text{ (City 4)} \\
& & x_4 + x_5 + x_6 & \geq 1 \text{ (City 5)} \\
& & x_2 + x_5 + x_6 & \geq 1 \text{ (City 6)} \\
& & x_i = 1 \text{ or } 0;\ i = 1, \dots, 6. &
\end{array}
$$

Optimal solution: $x_0 = 2$; $x_2 = x_4 = 1$, $x_1 = x_3 = x_5 = x_6 = 0$.

In a set covering problem each member of a given set (Set 1) must be "covered" by another member of some set (e.g. Set 2). The objective is to minimize the number of elements in Set 2 that are required to cover Set 1.

2. GENERAL TERMINOLOGY FOR INTEGER PROGRAMMING

The most general problem called the **mixed integer programming problem** can be specified as

$$
\begin{array}{ll}
\min & x_0 = c^T x \\
\text{subject to} & A x = b \\
& x_j \geq 0 \qquad j = 1, \dots, n \\
& x_j \text{ integer for } j \in IN
\end{array}
$$

where IN is some subset of $N_0 = \{0, 1, \dots n\}$.

When $IN = N_0$ we have what is called a **pure integer programming problem.** For such a problem, one generally has all given quantities c_j, a_{ij}, b_i integer. One has to be careful here. Consider for example

$$
\begin{array}{ll}
\min & x_0 = -\frac{1}{3} x_1 - \frac{1}{2} x_2 \\
\text{subject to} & \frac{2}{3} x_1 + \frac{1}{3} x_2 \leq \frac{4}{3} \\
& \frac{1}{2} x_1 - \frac{3}{2} x_2 \leq \frac{2}{3} \\
& x_0, x_1, x_2 \geq 0 \text{ and integer}
\end{array}
$$

As defined this is *not* a pure problem. For a start x_0 will not necessarily be integer and neither will the slack variables. If we want to use an algorithm for solving pure problems we must scale the objective and constraints to give:

$$
\begin{array}{ll}
\min & x_0 = -2x_1 - 3x_2 \\
\text{subject to} & 2x_1 + x_2 + x_3 = 4 \\
& 3x_1 - 9x_2 + x_4 = 4 \\
& x_1, \dots x_4 \geq 0 \text{ and integer.}
\end{array}
$$

A final class of problems is the **pure 0 – 1 programming problem**

$$
\begin{array}{ll}
\max & x_0 = c^T x \\
\text{subject to} & A x \leq b \\
& x_j = 0 \text{ or } 1 \quad \text{for } j = 1, \dots, n.
\end{array}
$$

3. FURTHER USES OF INTEGER VARIABLES

1. If a variable x can only take a finite number of values $p_1, \dots, p_m$ we can replace x by the expression

Generated by CamScanner from intsig.com

$$\begin{array}{ll}
\text{Minimize} & x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8 + x_9 + x_{10} + x_{11} \\
\text{subject to} & x_1 + x_2 + x_3 + x_4 \geq 1 \\
& x_1 + x_2 + x_3 + x_5 \geq 1 \\
& x_1 + x_2 + x_3 + x_4 + x_5 + x_6 \geq 1 \\
& x_1 + x_3 + x_4 + x_6 + x_7 \geq 1 \\
& x_2 + x_3 + x_5 + x_6 + x_8 + x_9 \geq 1 \\
& x_3 + x_4 + x_5 + x_6 + x_7 + x_8 \geq 1 \\
& x_4 + x_6 + x_7 + x_8 \geq 1 \\
& x_5 + x_6 + x_7 + x_8 + x_9 + x_{10} \geq 1 \\
& x_5 + x_8 + x_9 + x_{10} + x_{11} \geq 1 \\
& x_8 + x_9 + x_{10} + x_{11} \geq 1 \\
& x_9 + x_{10} + x_{11} \geq 1 \\
& x_j \in \{0,1\} \quad j = 1, \ldots, 11
\end{array}$$

The first constraint states that there must be a station either in neighborhood 1 or in some adjacent neighborhood. The next constraint is for neighborhood 2 and so on. Notice that the constraint coefficient a_{ij} is 1 if neighborhood i is adjacent to neighborhood j or if $i=j$ and 0 otherwise. The jth column of the constraint matrix represents the set of neighborhoods that can be served by a fire station in neighborhood j. We are asked to find a set of such subsets j that *covers* the set of all neighborhoods in the sense that every neighborhood appears in the service subset associated with *at least* one fire station.

One optimal solution to this is $x_3 = x_8 = x_9 = 1$ and the rest equal to 0.

This is an example of the *set covering problem*. The set covering problem is characterized by having binary variables, $\geq$ constraints each with a right hand side of 1, and having simply sums of variables as constraints. In general, the objective function can have any coefficients, though here it is of a particularly simple form.

- Set Packing and Partitioning

Michael A. Trick
Sun Jun 14 12:49:07 EDT 1998

Generated by CamScanner from intsig.com