Math 414 Lecture 25

Integer Programming

In many problems the variables must range over integers: the number of people you must hire, the number of chairs you must rent, a price in cents, ...

DEFINITION.

- A *pure integer* programming problem is a linear programming problem in which all the variables are required to be integers or natural numbers.
- In a *mixed integer* programming problem some of the variables may be required to be integral. The optimal value z, the coefficients a, b, c and the slacks are not required to be integers, just the variables.
- In a *0-1 programming* problem the variables must be only 0 or 1. *0-1 variables* are also called *Boolean* or *decision* variables.

Let Z = the set of integers, $N = \{0, 1, 2, 3, ...\}$ be the set of natural numbers. Hence instead of x unrestricted or $x \geq 0$ we may require $x \in Z$, $x \in N$ or $x \in \{0,1\}$.

Write the following as pure, mixed or 0-1 integer programming problems.

KNAPSACK PROBLEM. Let W = max weight you can carry on a hike. You have n items with weights $w_1, ..., w_n$. Unfortunately $W < w_1 + ... + w_n$, so you can't take them all. Assign values $v_1, ..., v_n$ to the n items such that the more important ones get higher values. Build a model for the problem of deciding for each item whether to take it or not in a way that maximizes the total value v.

Solution.

Let $x_i = 1$ if you decide to take the ith item,
$= 0$ if you do not.

Max $v = v_1 x_1 + v_2 x_2 + ... + v_n x_n$,
with
$$w_1 x_1 + w_2 x_2 + ... + w_n x_n \leq W$$
$$x_1, ..., x_n \in \{0,1\}$$

ASSIGNMENT PROBLEM. There are n workers (one task each) $w_1, ..., w_n$ and n tasks (one worker each) $t_1, ..., t_n$. r_{ij} = the revenue worker w_i generates when doing task t_j. Build a model for deciding which worker should do which task in order to maximize the total revenue r.

Solution. Let $x_{ij} = \begin{cases} 1 \text{ if } w_i \text{ does task } t_j, \\ 0 \text{ otherwise.} \end{cases}$

max $r = \Sigma_{i,j}\, r_{ij} x_{ij}$
with

For $i = 1...n$, i: $x_{i1} + x_{i2} + ... + x_{in} = 1$ $\quad \Sigma_{j=1}^{n} x_{ij} = 1$, or $\leq$

For $j = 1...n$, j: $x_{1j} + x_{2j} + ... + x_{nj} = 1$ $\quad \Sigma_{i=1}^{n} x_{ij} = 1$, or $\leq$

$x_{ij} \in \{0,1\}$ There are $2n$ constraints, not 2 constraints.

STOCK CUTTING PROBLEM. Pipe comes in 8′ and 12′ lengths. You need 10 3′ lengths and 7 4′ lengths. How many 8′ and 12′ lengths should be bought to minimize waste?

Solution.

Cutting patterns for 8′ lengths.

pattern #	pattern	waste	#times pattern is used
1:	3	+5	x_1
2:	4	+4	x_2
3:	3+3	+2	x_3
4:	4+3	+1	x_4
5:	4+4	+0	x_5

Cutting patterns for 12′ lengths.

pattern #	pattern	waste	
6:	3+3+3	+3	x_6
7:	3+3+3+3	+0	x_7
8:	4+3+3	+2	x_8
9:	4+4+3	+1	x_9
10:	4+4+4	+0	x_{10}

Let x_j be the number of times pattern j is used.

min w = ________________ where w = total waste.
with

(three ft.) t: __________________________________ = 10

(four ft.) f: __________________________________ = 7

$x_1, ..., x_{10} \in$ N

TRAVELING SALESMAN PROBLEM. A salesman has to visit each of n cities $c_1, ..., c_n$. He starts and ends at c_1. He must visit each city exactly once. Suppose d_{ij} is the distance between cities c_i and c_j.

Find the route which minimizes the total distance.

Solution. Given a route, let $x_{ij} = 1$ if c_i and c_j are successive cities along the route; $x_{ij} = 0$ otherwise. For $i = 1, ..., n$, let u_i = the position of city c_i (1st, 2nd, 3rd, ...) along the route. Let D = the total distance.

If the route is $c_1 \to c_4 \to c_2 \to c_3 \to c_1$,
then $x_{14} = x_{42} = x_{23} = x_{31} = 1$, all other $x_{ij} = 0$,
$u_1 = 1$, $u_4 = 2$, $u_2 = 3$, $u_3 = 4$.

min $D = \Sigma\{d_{ij} x_{ij} : i, j = 1 ... n\}$
with

i: $\quad \Sigma_{j=1}^{n} x_{ij} = 1 \qquad$ for $i = 1 ... n$

j: $\quad \Sigma_{i=1}^{n} x_{ij} = 1 \qquad$ for $j = 1 ... n$

i–j: $\quad u_i - u_j + n x_{ij} \leq n - 1 \quad i, j \in \{1, ..., n\}, j \neq 1$.

$x_{ij} \in \{0,1\}$, $u_i \in N$.

The last condition rules out short cut paths with two or more loops.

Full loop example (o.k.)	Two loop example (not o.k.)
$c_1 \to c_4 \to c_2 \to c_3 \to c_1$	$c_1 \to c_4 \to c_1$,
	$c_2 \to c_3 \to c_2$