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 linear programming problem is --
- a pure integer programming problem if all the variables are integers or natural numbers.
- a mixed integer programming problem if some of the variables are integral, some not.
- a 0-1 programming problem if the variables are only 0 or 1. These are also called Boolean or decision variables.

Traditionally, 1 = "yes" and 0 = "no".

The optimal value v, the coefficients a, b, c and the slacks are not required to be integers, just the variables.

Let Z = the set of integers, N = {0, 1, 2, 3, ...} = the set of natural numbers. Besides "x unrestricted", "x≥0", or "x≤ 0" we may now require "x∈Z", "x∈N" or "x∈{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 w1, ..., wn. Unfortunately W<w1+...+wn, so you can’t take them all.

Assign values v1, ..., vn to the n items so 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 x1=1 if you decide to take the ith item, =0 if you do not.

Max v = v1x1 + v2x2 + ... + vnxn, with
w1x1 + w2x2 + ... + wnxn ≤ W
x1, ..., xn ∈ {0, 1}

Assignment problem. There are n workers (one task each) w1, ..., wn and n tasks (one worker each) t1, ..., tn.

rij = the revenue worker wi generates when doing task tj.

Build a model for deciding which worker should do which task in order to maximize the total revenue r.

Solution. Let xij = 1 if wi does task tj,
=0 otherwise.

max r = ∑ij rijxij
with
For i = 1...n, i:j xij + xij2 + ... + xijn = 1
For j = 1...n, j:i xij + xij2 + ... + xijm = 1
xij ∈ {0, 1}

There are 2n constraints, not 2 constraints.

Traveling salesmen problem. A salesman has to visit each of n cities c1, ..., cn. He starts and ends at c1. He must visit each city exactly once. Suppose dij is the distance between cities c1 and c2.

Find the route which minimizes the total distance.

Solution. Given a route, let xij = 1 if ci and cj are successive cities along the route; xij = 0 if not. For i=1,...,n, let u0 be the position of city c1 (1st, 2nd, 3rd, ... ) along the route. Let D = the total distance.

If the route is c1 → c4 → c2 → c1 → c1, then x14 =una =x23 =x31 =1, all other xij = 0, u1=1, u4 =2, u2 =3, u3 =4.

min D = ∑ {dijxij; i, j = 1 ... n}
with
i: ∑j=1n xij = 1 for i=1...n --every i has 1 successor
j: ∑i=1n xij = 1 for j=1...n --every j has 1 predecessor
i→j: u0 - u0 + nxij ≤ n-1 i,j ∈ {1, ..., n}, j ≠ 1.

xij ∈ {0, 1}, ui ∈ 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.)
c1 → c4 → c2 → c1 → c1 c1 → c4 → c1, c2 → c3 → c2
The Cutting Plane Method

Definition. For any real $r$,
- $\lfloor r \rfloor = \text{the floor of } r = \text{the closest integer } \leq r$.
- The number $r$ is integral iff $r \in \mathbb{Z} = \{..., -2, -1, 0, 1, 2, ... \}$.

- $\lfloor 3.66 \rfloor = 3$, .66 is the decimal part of 3.66.
- $\lfloor 3.00 \rfloor = 3$, $\lfloor -3 \rfloor = -3$, $\lfloor -3.3 \rfloor = -4$ (-4, not -3).

Facts.
- $r - 1 < \lfloor r \rfloor < r$.
- $\lfloor n \rfloor = n$ if $n \in \mathbb{Z}$.
- $s \leq r \Rightarrow \lfloor s \rfloor \leq r$.
- For $n \in \mathbb{Z}$, $n \leq r \Rightarrow n = \lfloor r \rfloor$.

Here’s a problem in standard form (with $\leq$ not $=$ or $\geq$).

max $z = 2x + y$ with $a$: $x + 1.5y \leq 2.9$ $x, y \in \mathbb{N}$

Optimal noninteger solution: $x = 2.9, y = 0, z = 5.8$.

Feasible integer solutions $[x, y, z]$: $[0, 0, 0], [0, 1, 1], [1, 0, 2], [1, 1, 3], [2, 0, 4]$.

Optimal integer solution: $x = 2, y = 0, z = 4$.

We want an additional constraint which
(a) cuts off the current optimal noninteger solution but
(b) doesn’t cut off any integer solution.

original constraint: $(1)x + (1.5)y \leq (2.9)$
added new constraint: $[1]x + [1.5]y \leq [2.9]$
which simplifies to $x + y \leq 2$ - the dotted line.

With the added new constraint
(a) The old nonintegral optimal solution $[2.9, 0]$ is cut off.
(b) No integer solution is cutoff. For any integral solution $[x, y]$: $(1)x + (1.5)y \leq (2.9)$ implies, for any $x, y \geq 0$, $[1]x + [1.5]y \leq (2.9)$ implies, for $x, y$ integral,
$[1]x + [1.5]y \leq [2.9]$ (see the fourth fact above).

Stated directly without the $[]$ symbols, the argument is:

$(1)x + (1.5)y \leq (2.9)$ implies, for any $x, y \geq 0$,
$1x + 1y \leq (2.9)$ implies, for $x, y$ integral,
$1x + 1y \leq 2$

The reasoning here is that for any integer $k$, $k$ is $\leq 2.9$
implies $k$ is $\leq$ the largest integer $\leq 2.9$
implies $k$ is $\leq [2.9]$ -- see fact four --
implies $k$ is $\leq 2$.

In order to assure that the slacks are also integral, clear
the fractions in the original constraints. Hence
$(1)x + (1.5)y \leq (2.9)$ would be multiplied by 10 to get
$10x + 15y \leq 29$
This doesn’t change the values of $x$ or $y$ but does make
the slacks integral when $x$ and $y$ are integral.