Fixed Charge Problem

Suppose that a production planning problem involves \( N \) products. Each product \( j \) consists of a fixed cost \( c_j \) independent of the amount produced and a variable cost \( c_j \) per unit. If \( x_j \) is the production level of product \( j \), its total cost function may be written as:

\[
C_j(x_j) = \begin{cases} 
  k_j + c_j x_j & ; x_j > 0 \\
  0 & ; x_j = 0 
\end{cases}
\]

The complete problem is therefore,

\[
\begin{align*}
\text{Min } Z &= \sum_{j=1}^{N} C_j(x_j) \\
\text{Subject to: } & \text{ given LP constraints} \\
\end{align*}
\]

Intractable!

\[
y_j = \begin{cases} 
  0 & ; x_j = 0 \\
  1 & ; x_j > 0 
\end{cases}
\]

\[\Rightarrow x_j \leq M y_j\]
The obj. fun. can be written as:
\[ \text{Min } Z = \sum_{j=1}^{n} (c_j x_j + k_j y_j) \]
\[ \text{s.t.: } 0 \leq x_j \leq M y_j, \forall i \]
\[ y_j = 0 \text{ or } 1, \forall j \]

plus given linear constraints

**Integer programming (IP) problems**

(Some features/fallacies)

1. A pure IP problem with a bounded feasible region will have a finite number of feasible solutions.
   - True! but this finite # may be astronomically large.

2. Binary variables

\[
\begin{array}{cc}
X_1 & X_2 \\
0 & 0 \\
1 & 0 \\
0 & 1 \\
1 & 1 \\
\end{array}
\]

\[ 2^2 = 4 \]

5. BVs \[ 2^5 = 32 \]
10 BVs \[ 2^{10} = 1024 \]
15 BVs \[ 2^{15} = 32768 \]
20 BVs \[ 2^{20} = 1.048576 \times 10^{26} \]
8 groups $\Rightarrow 8! = 40,320$
9 groups $\Rightarrow 9! = 362,880$
10 groups $\Rightarrow 10! = 3,628,800$

\[ \text{Computational time} \]
\[ \text{\# of variables} \]
\[ \text{Solvable cases} \]
\[ \text{Unsolvable cases:} \]
\[ \text{NP hard} \]
\[ \text{NP polymonially hard} \]

\[ \Rightarrow \text{Solving a problem with a large \# of variables optimally may not be possible even with a sophisticated computer.} \]

2. It is easier to solve an IP problem by considering its LP relaxation and removing the noninteger feasible solutions.

\[ \Rightarrow \text{Disregarding the noninteger (basic) feasible solutions will not guarantee finding an optimal solution for the IP problem.} \]
3. The two most important factors in an IP problem are:

a. Structure of the problem

   The nature of the problem lends itself for formulating the model as a real LP (i.e., all \( x_j \geq 0 \)), which guarantees identifying an integer opt. soln.

   (ex: transportation problem, assignment problem, etc.)

b. \( \uparrow \) Number of integer variables \( \uparrow \) Complexity of the problem

4. The solution obtained by rounding off a noninteger opt. soln. to an integer soln. may not be feasible.

   \( -x_1 + x_2 \leq 6.5 \) \( \rightarrow \) \( \circ \)

   \( x_1 + x_2 \leq 19.5 \) \( \rightarrow \) \( \circ \)

   Suppose the real opt. soln. is:

   \[ (x_1^*, x_2^*) = (6.5, 13) \]
Difficulties

It is hard to see how to perform the rounding (up or down) to retain feasibility.

The rounded solution may not be feasible.

i.e.,

(i) \( x_1, x_2 = (6.13) \Rightarrow \text{const.}(1) \text{ in violated}

(ii) \( x_1, x_2 = (7.13) \Rightarrow \text{"(2)" violated.}

Even if the rounded solution is feasible, there is no guaranteed that it is optimal.

Max \( Z = 4x_1 + 15x_2 \)

\[ \begin{align*}
\text{st}:
& x_1 + 4x_2 \leq 8 \\
& x_1 + x_2 \leq 4 \\
& x_1, x_2 \text{ integers.}
\end{align*} \]

\( x_1 = 2\frac{2}{3}, \ x_2 = 1\frac{1}{3} \)

\( Z^* = 4\left(\frac{2}{3}\right) + 15\left(\frac{4}{3}\right) = \frac{32}{3} = 10.67 \)

\( x_1 \quad \text{①} \quad \text{②} \quad \text{③} \Rightarrow (2, 1), (6, 2) \)

\( x_2 \quad \text{①} \quad \text{②} \Rightarrow (3, 1), (3, 2) \)

\( (2, 2) \) and \( (3, 2) \) are not acceptable.

The rounded solutions are:

(2, 1) and (3, 1)
\[ z(2,1) = (4)(2) + (15)(1) = 23 \]
\[ z(3,1) = (4)(3) + (15)(1) = 27 \]

But the true integer opt. soln is:

\[ (x_1^*, x_2^*)_{\text{int}} = (0, 2) \]

\[ \bar{z}^*_{\text{int}} = (4)(0) + (15)(2) = 30 \]

Notice that \( \bar{z}^*_{\text{int}} > z(2,1) + z(3,1) \)

Also, \( z^*_{\text{real}} = \frac{92}{3} \geq \bar{z}^*_{\text{int}} \)

\[ \text{In general, } z^*_{\text{real}} \geq \bar{z}^*_{\text{int}} \]

We should implicitly enumerate the feasible integer solutions, and then determine the opt. integer soln. Such an approach is called the branch-and-bound algorithm.
For large problems, we should use efficient heuristic algorithms (i.e., tabu search, simulated annealing, and genetic algorithms) which can all be classified higher-level search algorithms to find near-optimal solutions.

Branch-and-Bound Algorithm