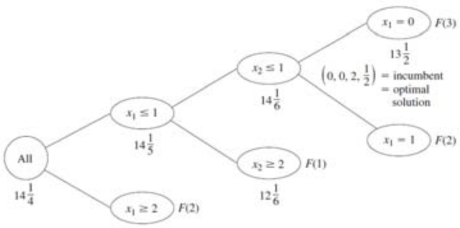
Lecture 13: Integer Programming



Integer Programming

- A key limitation that prevents many applications for LP is the assumption of divisibility, which requires that noninteger values be permissible for decision variables.
- If requiring integer values is the only way in which a problem deviates from a linear programming formulation, then it is an integer programming (IP) problem.
- If only *some of the variables are required to have integer values* (so the divisibility assumption holds for the rest), this model is referred to as mixed integer programming (MIP).

Binary Integer Programming

- Problems involving a number of interrelated "yes-or-no decisions."

$$x_j = \begin{cases} 1 & \text{if decision } j \text{ is yes} \\ 0 & \text{if decision } j \text{ is no.} \end{cases}$$

- Such variables are called binary variables (or 0–1 variables).
- Consequently, IP problems that contain only binary variables sometimes are called binary integer programming (BIP) problems (or 0–1 integer programming problems).

Prototype Example

- The CALIFORNIA MANUFACTURING COMPANY is considering expansion by building a new factory in either Los Angeles or San Francisco, or perhaps even in both cities. It also is considering building at most one new warehouse, but the choice of location is restricted to a city where a new factory is being built. The *net present value* (total profitability considering the time value of money) of each of these alternatives is shown in the fourth column of table below. The rightmost column gives the capital required (already included in the net present value) for the respective investments, where the total capital available is $10 million. The objective is to find the feasible combination of alternatives that maximizes the total net present value.

Data for the California Manufacturing Co. example

Decision Number	Yes-or-No Question	Decision Variable	Net Present Value	Capital Required
1	Build factory in Los Angeles?	x_1	\$9 million	\$6 million
2	Build factory in San Francisco?	x_2	\$5 million	\$3 million
3	Build warehouse in Los Angeles?	x_3	\$6 million	\$5 million
4	Build warehouse in San Francisco?	x_4	\$4 million	\$2 million
			Capital available:	\$10 million

The BIP Model

$$x_j = \begin{cases} 1 & \text{if decision } j \text{ is yes,} \\ 0 & \text{if decision } j \text{ is no,} \end{cases} \quad (j = 1, 2, 3, 4).$$

Maximize $Z = 9x_1 + 5x_2 + 6x_3 + 4x_4,$

subject to

$$\begin{aligned} 6x_1 + 3x_2 + 5x_3 + 2x_4 &\le 10 \\ x_3 + x_4 &\le 1 \\ -x_1 + x_3 &\le 0 \\ -x_2 + x_4 &\le 0 \\ x_j &\le 1 \\ x_j &\ge 0 \end{aligned}$$

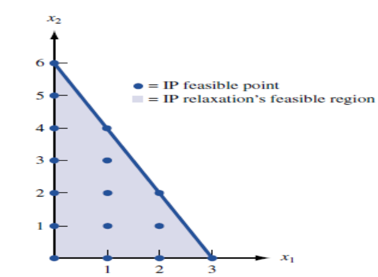
x_j is integer, for $j = 1, 2, 3, 4.$

x_j is binary, for $j = 1, 2, 3, 4.$

Some Perspectives on Solving IP Problems

- For a simple case of BIP problems with n variables, there are 2^n solutions to be considered.
- This pattern is referred to as the exponential growth of the difficulty of the problem.
- Because of exponential growth, even the best algorithms cannot be guaranteed to solve every relatively small problem.
- When the integer restriction is deleted, the problem is called LP relaxation.

Feasible Region for an IP and Its LP Relaxation



Some Perspectives on Solving IP Problems

- Two primary determinants of computational difficulty for an IP problem are
  - (1) the number of integer variables and
  - (2) any special structure in the problem.

The Branch-and-bound Technique

- An enumeration procedure.
- The basic concept underlying the branch-and-bound technique is to *divide and conquer*.
- Since the original "large" problem is too difficult to be solved directly, it is divided into smaller and smaller subproblems until these subproblems can be conquered.
- The dividing (*branching) is done by partitioning the entire set of feasible solutions into smaller* and smaller subsets.
- The conquering (*fathoming) is done partially by bounding how good* the best solution in the subset can be and then discarding the subset if its bound indicates that it cannot possibly contain an optimal solution for the original problem.

Prototype Example

Maximize $Z = 9x_1 + 5x_2 + 6x_3 + 4x_4$,

subject to

(1) $6x_1 + 3x_2 + 5x_3 + 2x_4 \le 10$
(2) $x_3 + x_4 \le 1$
(3) $-x_1 + x_3 \le 0$
(4) $-x_2 + x_4 \le 0$

and

(5) x_j is binary, for $j = 1, 2, 3, 4$.

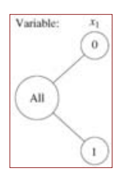
Branching

Subproblem 1:
Fix $x_1 = 0$ so the resulting subproblem is

Maximize $Z = 5x_2 + 6x_3 + 4x_4$,

subject to

(1) $3x_2 + 5x_3 + 2x_4 \le 10$
(2) $x_3 + x_4 \le 1$
(3) $x_3 \le 0$
(4) $-x_2 + x_4 \le 0$
(5) x_j is binary, for $j = 2, 3, 4$.



Subproblem 2:
Fix $x_1 = 1$ so the resulting subproblem is

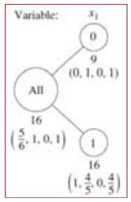
Maximize $Z = 9 + 5x_2 + 6x_3 + 4x_4$,

subject to

(1) $3x_2 + 5x_3 + 2x_4 \le 4$
(2) $x_3 + x_4 \le 1$
(3) $x_3 \le 1$
(4) $-x_2 + x_4 \le 0$
(5) x_j is binary, for $j = 2, 3, 4$.

Bounding

- Solve a simpler relaxation of the subproblem.
- LP relaxation of the whole problem:



$$(x_1, x_2, x_3, x_4) = \left(\frac{5}{6}, 1, 0, 1\right), \quad \text{with } Z = 16\frac{1}{2}.$$

Bound for whole problem: $Z \leq 16.$

LP relaxation of subproblem 1: $(x_1, x_2, x_3, x_4) = (0, 1, 0, 1)$ with $Z = 9.$

LP relaxation of subproblem 2: $(x_1, x_2, x_3, x_4) = \left(1, \frac{4}{5}, 0, \frac{4}{5}\right)$ with $Z = 16\frac{1}{5}.$

Bound for subproblem 1: $Z \leq 9,$
Bound for subproblem 2: $Z \leq 16.$

Bangladesh University of Eng. & Tech. Slide 13 of 39 Industrial &Production Engineering

Fathoming

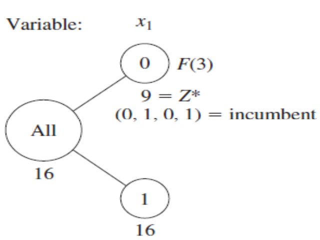
- A subproblem can be conquered (fathomed), and thereby dismissed from further consideration, in three ways.

Summary of Fathoming Tests. A subproblem is *fathomed* (dismissed from further consideration) if

Test 1: Its bound $\leq Z^*$,
or
Test 2: Its LP relaxation has no feasible solutions,
or
Test 3: The optimal solution for its LP relaxation is *integer.* (If this solution is better than the incumbent, it becomes the new incumbent, and test 1 is reapplied to all unfathomed subproblems with the new larger Z^*.)

Bangladesh University of Eng. & Tech. Slide 14 of 39 Industrial &Production Engineering

Fathoming



Bangladesh University of Eng. & Tech. Slide 15 of 39 Industrial &Production Engineering

Summary of BIP Branch-and-Bound Algorithm

- This algorithm assumes that all coefficients in the objective function are integer and that the ordering of the variables for branching is $x_1, x_2, \ldots, x_n$.

Summary of the BIP Branch-and-Bound Algorithm.

Initialization: Set $Z^* = -\infty$. Apply the bounding step, fathoming step, and optimality test described below to the whole problem. If not fathomed, classify this problem as the one remaining "subproblem" for performing the first full iteration below.

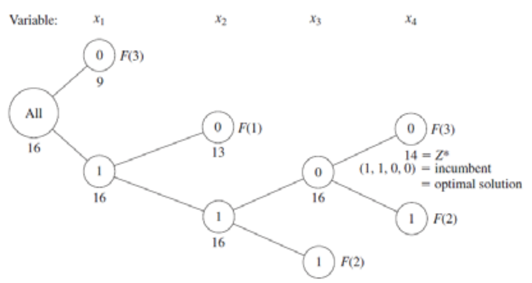
Steps for each iteration:

1. *Branching:* Among the *remaining* (unfathomed) subproblems, select the one that was created *most recently.* (Break ties according to which has the *larger bound.*) Branch from the node for this subproblem to create two new subproblems by fixing the next variable (the branching variable) at either 0 or 1.
2. *Bounding:* For each new subproblem, obtain its *bound* by applying the simplex method to its LP relaxation and rounding down the value of Z for the resulting optimal solution.
3. *Fathoming:* For each new subproblem, apply the three fathoming tests summarized above, and discard those subproblems that are fathomed by any of the tests.

Optimality test: Stop when there are *no remaining* subproblems; the current *incumbent* is optimal.[1] Otherwise, return to perform another iteration.

Bangladesh University of Eng. & Tech. Slide 16 of 39 Industrial &Production Engineering

The solution tree after final (fourth) iteration of the BIP branch-and-bound algorithm



Bangladesh University of Eng. & Tech. | Slide 17 of 39 | Industrial &Production Engineering

Example

D,I **12.6-1.*** Use the BIP branch-and-bound algorithm presented in Sec. 12.6 to solve the following problem interactively.

Maximize $Z = 2x_1 - x_2 + 5x_3 - 3x_4 + 4x_5,$

subject to

$$3x_1 - 2x_2 + 7x_3 - 5x_4 + 4x_5 \le 6$$
$$x_1 - x_2 + 2x_3 - 4x_4 + 2x_5 \le 0$$

and

x_j is binary, for $j = 1, 2, \dots, 5.$

Bangladesh University of Eng. & Tech. | Slide 18 of 39 | Industrial &Production Engineering

A Branch-and-bound Algorithm for Mixed Integer Programming

Maximize $Z = \sum_{j=1}^{n} c_j x_j,$

subject to

$$\sum_{j=1}^{n} a_{ij} x_j \le b_i, \quad \text{for } i = 1, 2, \dots, m,$$

and

$x_j \ge 0,$ for $j = 1, 2, \dots, n,$
x_j is integer, for $j = 1, 2, \dots, I;\ I \le n.$

(When $I = n$, this problem becomes the pure IP problem.)

Bangladesh University of Eng. & Tech. | Slide 19 of 39 | Industrial &Production Engineering

A Branch-and-bound Algorithm for Mixed Integer Programming

- The structure of this algorithm was first developed by R. J. Dakin, based on a pioneering branch-and-bound algorithm by A. H. Land and A. G. Doig.
- Only four changes are needed in the BIP algorithm to deal with the generalizations from *binary to general integer variables and from pure IP to mixed IP.*
  - The choice of the *branching variable.*
  - The values assigned to the branching variable for creating the new smaller subproblems.
  - The *bounding step.*
  - The fathoming test 3.

Bangladesh University of Eng. & Tech. | Slide 20 of 39 | Industrial &Production Engineering

The values assigned to the branching variable

- Let x_j *be the current branching variable, and* let $x_j{}^*$ *be* its (noninteger) value in the optimal solution for the LP relaxation of the current subproblem.

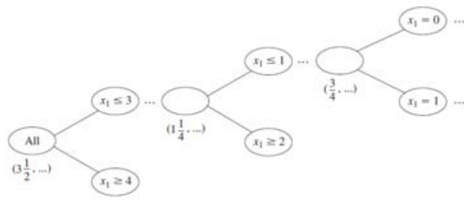
$$[x_j^*] = \text{greatest integer} \leq x_j^*,$$

$$x_j \leq [x_j^*] \quad \text{and} \quad x_j \geq [x_j^*] + 1,$$

For example, if $x_j^* = 3\frac{1}{2}$, then

$$x_j \leq 3 \quad \text{and} \quad x_j \geq 4$$

Recurring branching variable



The fathoming test 3

- With a *mixed IP problem,* the test requires only that the *integer-restricted variables be integer in the optimal solution* for the subproblem's LP relaxation.

Summary of the MIP Branch-and-Bound Algorithm

Initialization: Set $Z^* = -\infty$. Apply the bounding step, fathoming step, and optimality test described below to the whole problem. If not fathomed, classify this problem as the one remaining subproblem for performing the first full iteration below.

Steps for each iteration:

1. *Branching:* Among the *remaining* (unfathomed) subproblems, select the one that was created *most recently.* (Break ties according to which has the *larger bound.*) Among the *integer-restricted* variables that have a *noninteger* value in the optimal solution for the LP relaxation of the subproblem, choose the *first one* in the natural ordering of the variables to be the *branching variable.* Let x_j be this variable and x_j^* its value in this solution. Branch from the node for the subproblem to create two new subproblems by adding the respective constraints $x_j \leq [x_j^*]$ and $x_j \geq [x_j^*] + 1$.
2. *Bounding:* For each new subproblem, obtain its bound by applying the simplex method (or the dual simplex method when reoptimizing) to its LP relaxation and using the value of Z for the resulting optimal solution.
3. *Fathoming:* For each new subproblem, apply the three fathoming tests given below, and discard those subproblems that are fathomed by any of the tests.
   Test 1: Its bound $\leq Z^*$, where Z^* is the value of Z for the current *incumbent.*
   Test 2: Its LP relaxation has no feasible solutions.
   Test 3: The optimal solution for its LP relaxation has *integer* values for the *integer-restricted* variables. (If this solution is better than the incumbent, it becomes the new incumbent and test 1 is reapplied to all unfathomed subproblems with the new larger Z^*.)

Optimality test: Stop when there are no remaining subproblems; the current *incumbent* is optimal.[†] Otherwise, perform another iteration.

[†]If there is no incumbent, the conclusion is that the problem has no feasible solutions.

Example

D,I **12.7-9.** Use the MIP branch-and-bound algorithm presented in Sec. 12.7 to solve the following MIP problem interactively.

Maximize $Z = 3x_1 + 4x_2 + 2x_3 + x_4 + 2x_5,$

subject to

$$2x_1 - x_2 + x_3 + x_4 + x_5 \le 3$$
$$-x_1 + 3x_2 + x_3 - x_4 - 2x_5 \le 2$$
$$2x_1 + x_2 - x_3 + x_4 + 3x_5 \le 1$$

and

$x_j \ge 0,$ for $j = 1, 2, 3, 4, 5$
x_j is binary, for $j = 1, 2, 3.$

An MIP Example

Maximize $Z = 4x_1 - 2x_2 + 7x_3 - x_4,$

subject to

$$x_1 + 5x_3 \le 10$$
$$x_1 + x_2 - x_3 \le 1$$
$$6x_1 - 5x_2 \le 0$$
$$-x_1 + 2x_3 - 2x_4 \le 3$$

and

$x_j \ge 0,$ for $j = 1, 2, 3, 4$
x_j is an integer, for $j = 1, 2, 3.$

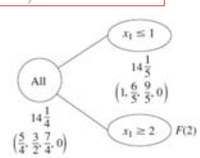
An MIP Example

LP relaxation of whole problem: $(x_1, x_2, x_3, x_4) = \left(\frac{5}{4}, \frac{3}{2}, \frac{7}{4}, 0\right)$, with $Z = 14\frac{1}{4}$.

Iteration 1.

Subproblem 1:
Original problem plus additional constraint
$x_1 \le 1.$

Subproblem 2:
Original problem plus additional constraint
$x_1 \ge 2.$



LP relaxation of subproblem 1: $(x_1, x_2, x_3, x_4) = \left(1, \frac{6}{5}, \frac{9}{5}, 0\right)$, with $Z = 14\frac{1}{5}$.

Bound for subproblem 1: $Z \le 14\frac{1}{5}$.

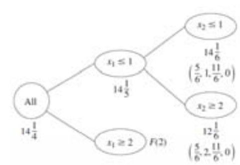
LP relaxation of subproblem 2: No feasible solutions.

An MIP Example

Iteration 2.

Subproblem 3:
Original problem plus additional constraints
$x_1 \le 1,$ $x_2 \le 1.$

Subproblem 4:
Original problem plus additional constraints
$x_1 \le 1,$ $x_2 \ge 2.$



LP relaxation of subproblem 3: $(x_1, x_2, x_3, x_4) = \left(\frac{5}{6}, 1, \frac{11}{6}, 0\right)$, with $Z = 14\frac{1}{6}$.

Bound for subproblem 3: $Z \le 14\frac{1}{6}$.

LP relaxation of subproblem 4: $(x_1, x_2, x_3, x_4) = \left(\frac{5}{6}, 2, \frac{11}{6}, 0\right)$, with $Z = 12\frac{1}{6}$.

Bound for subproblem 4: $Z \le 12\frac{1}{6}$.

An MIP Example

Iteration 3.

Subproblem 5:
Original problem plus additional constraints

$x_1 \le 1$
$x_2 \le 1$
$x_1 \le 0$ (so $x_1 = 0$).

Subproblem 6:
Original problem plus additional constraints

$x_1 \le 1$
$x_2 \le 1$
$x_1 \ge 1$ (so $x_1 = 1$).

LP relaxation of subproblem 5: $(x_1, x_2, x_3, x_4) = \left(0, 0, 2, \frac{1}{2}\right)$, with $Z = 13\frac{1}{2}$.

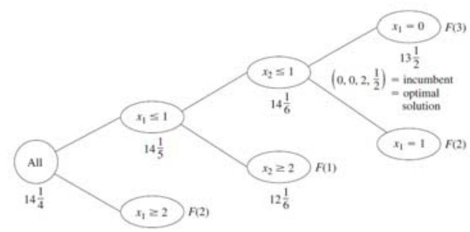
Bound for subproblem 5: $Z \le 13\frac{1}{2}$.

LP relaxation of subproblem 6: No feasible solutions.

Incumbent $= \left(0, 0, 2, \frac{1}{2}\right)$ with $Z^* = 13\frac{1}{2}$.

An MIP Example

Optimal solution $= \left(0, 0, 2, \frac{1}{2}\right)$ with $Z = 13\frac{1}{2}$.



Example

$$\begin{aligned}
\max z &= 4x_1 + 3x_2 + x_3 \\
\text{s.t.} \quad & 3x_1 + 2x_2 + x_3 \le 7 \\
& 2x_1 + x_2 + 2x_3 \le 11 \\
& x_2, x_3 \text{ integer}, x_1, x_2, x_3 \ge 0
\end{aligned}$$

Innovative Uses of Binary Variables in Model Formulation

- x_j denote the original variables of the problem (they may be either continuous or integer variables) and the y_i denote the auxiliary binary variables that are introduced for the reformulation.

Either-Or Constraints

Either $3x_1 + 2x_2 \le 18$
or $x_1 + 4x_2 \le 16,$

Either
$$3x_1 + 2x_2 \le 18$$
$$x_1 + 4x_2 \le 16 + M$$
or
$$3x_1 + 2x_2 \le 18 + M$$
$$x_1 + 4x_2 \le 16.$$

$$3x_1 + 2x_2 \le 18 + My$$
$$x_1 + 4x_2 \le 16 + M(1 - y).$$

K out of *N* Constraints Must Hold

- Consider the case where the overall model includes a set of *N* possible constraints such that only some *K* of these constraints must hold. (Assume that $K < N$)

$$f_1(x_1, x_2, \ldots, x_n) \le d_1$$
$$f_2(x_1, x_2, \ldots, x_n) \le d_2$$
$$\vdots$$
$$f_N(x_1, x_2, \ldots, x_n) \le d_N.$$

$$f_1(x_1, x_2, \ldots, x_n) \le d_1 + My_1$$
$$f_2(x_1, x_2, \ldots, x_n) \le d_2 + My_2$$
$$\vdots$$
$$f_N(x_1, x_2, \ldots, x_n) \le d_N + My_N$$
$$\sum_{i=1}^{N} y_i = N - K,$$

and

y_i is binary, for $i = 1, 2, \ldots, N,$

Functions with *N* Possible Values

- Consider the situation where a given function is required to take on any one of *N* given values.

$$f(x_1, x_2, \ldots, x_n) = d_1 \quad \text{or} \quad d_2, \ldots, \quad \text{or} \quad d_N.$$

$$f(x_1, x_2, \ldots, x_n) = \sum_{j=1}^{n} a_j x_j,$$

$$f(x_1, x_2, \ldots, x_n) = x_j$$

$$f(x_1, x_2, \ldots, x_n) = \sum_{i=1}^{N} d_i y_i$$
$$\sum_{i=1}^{N} y_i = 1$$

and

y_i is binary, for $i = 1, 2, \ldots, N.$

The Fixed-Charge Problem

- It is quite common to incur a fixed charge or setup cost when undertaking an activity.
- Frequently the variable cost will be at least roughly proportional to the level of the activity.

$$f_j(x_j) = \begin{cases} k_j + c_j x_j & \text{if } x_j > 0 \\ 0 & \text{if } x_j = 0, \end{cases}$$

- Suppose that there are *n* activities, each with the preceding cost structure (with $k_j \ge 0$ in every case and $k_j > 0$ for some $j = 1, 2, \ldots, n$).

Minimize $Z = f_1(x_1) + f_2(x_2) + \cdots + f_n(x_n),$

subject to

given linear programming constraints.

The Fixed-Charge Problem

$$Z = \sum_{j=1}^{n} (c_j x_j + k_j y_j),$$

where

$$y_j = \begin{cases} 1 & \text{if } x_j > 0 \\ 0 & \text{if } x_j = 0. \end{cases}$$

- y_j can be viewed as contingent decisions.

$$x_j \le M y_j \qquad \text{for } j = 1, 2, \ldots, n$$

- The above constraints ensure that y_j = 1 rather than 0 whenever $x_j > 0$.
- The one difficulty remaining is that these constraints leave y_j free to be either 0 or 1 when $x_j = 0$.

The Fixed-Charge Problem

To summarize, the MIP formulation of the fixed-charge problem is

$$\text{Minimize} \quad Z = \sum_{j=1}^{n} (c_j x_j + k_j y_j),$$

subject to

the original constraints, plus
$x_j - M y_j \le 0$

and

y_j is binary, for $j = 1, 2, \ldots, n$.

If the x_j also had been restricted to be integer, then this would be a *pure* IP problem.

Binary Representation of General Integer Variables

$$0 \le x \le u$$

and if N is defined as the integer such that

$$2^N \le u < 2^{N+1},$$

then the **binary representation** of x is

$$x = \sum_{i=0}^{N} 2^i y_i,$$

$$x_1 \le 5$$
$$2x_1 + 3x_2 \le 30.$$

$u = 5$ for x_1 $\quad u = 10$ for x_2

$N = 2$ for x_1 (since $2^2 \le 5 < 2^3$)

$N = 3$ for x_2 (since $2^3 \le 10 < 2^4$).

$$x_1 = y_0 + 2y_1 + 4y_2$$
$$x_2 = y_3 + 2y_4 + 4y_5 + 8y_6.$$

$$y_0 + 2y_1 + 4y_2 \le 5$$
$$2y_0 + 4y_1 + 8y_2 + 3y_3 + 6y_4 + 12y_5 + 24y_6 \le 30.$$