دانشگاه صنعتی اصفهان
دانشکده مهندسی حمل و نقل

پژوهش عملیاتی

برنامه ریزی عدد صحیح

Integer Programming

مدرس: محمد تمنایی

پاییز ۱۳۹۴

When and how to use
Integer Programming ?

- If a practical problem has characteristics with types:
  - ✓ Problems with discrete (not continuous) inputs or outputs
  - ✓ Problems with logical conditions
  - ✓ Problems with yes / no decisions
  - ✓ ...

- ✓ Discrete quantities

    Variables to represent a quantity: People, cars, …

- ✓ Variables for Decision

    Usually in 0-1 (Binary) form

    Decision Variables: usually in Greek form (δ, γ, α, …)

    Continuous Variables: usually in Latin form

- ✓ Indicator Variables (If- Then)

    Usually in 0-1 (Binary) form

    Example: (continuous variable x >=0, binary variable δ)

$$\left\{\begin{array}{l}\delta = 1 \rightarrow x > 0 \\ \delta = 0 \rightarrow x = 0\end{array}\right\} \equiv \{\delta = 1 \leftrightarrow x > 0\} \xrightarrow{Constraints} \left\{\begin{array}{l} x - M\delta \le 0 \\ x + M(1-\delta) > 0\end{array}\right\}$$

✓ Either-Or Constraints

A choice can be made between two constraints, so that only one must hold.

Either $3x_1 + 2x_2 \leq 18$

Or $x_1 + 4x_2 \leq 16$

How ? Auxiliary variable y (Binary)

$$\begin{cases} 3x_1 + 2x_2 \leq 18 + My \\ x_1 + 4x_2 \leq 16 + M(1-y) \end{cases}$$

This formulation guarantees that 1 of the original constraints must hold while the other is, in effect, eliminated.

$$\begin{cases} 3x_1 + 2x_2 \leq 18 + My_1 \\ x_1 + 4x_2 \leq 16 + My_2 \\ y_1 + y_2 = 1 \end{cases}$$

✓ *K out of N Constraints Must Hold*

The overall model includes a set of N possible constraints, such that: only some K of them must hold (K < N).

$$f_1(x_1, x_2, \dots, x_n) \le d_1$$
$$f_2(x_1, x_2, \dots, x_n) \le d_2$$
$$\dots$$
$$f_N(x_1, x_2, \dots, x_n) \le d_N$$

Only K of N must hold

How ? Auxiliary variable $y_1, y_2, \dots, y_N$ (Binary)

$$\begin{cases} f_1(x_1, x_2, \dots, x_n) \le d_1 + My_1 \\ f_2(x_1, x_2, \dots, x_n) \le d_2 + My_2 \\ \dots \\ f_N(x_1, x_2, \dots, x_n) \le d_N + My_N \\ y_1 + y_2 + \dots + y_N = N - K \\ y_i \text{ binary } (i = 1, 2, \dots, N) \end{cases}$$

✓ *Functions with N Possible Values*

A given function is required to take on any one of N given values.

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

How ? Auxiliary variable $y_1, y_2, \ldots, y_N$ (Binary)

$$\begin{cases} f(x_1, x_2, . . . , x_n) = d_1 y_1 + d_2 y_2 + \ldots + d_N y_N \\ y_1 + y_2 + \ldots + y_N = 1 \\ y_i \text{ binary } (i = 1, 2, . . . , N) \end{cases}$$

✓ The Fixed-Charge Problem

to incur a fixed charge or setup cost when undertaking an activity.

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

x_j = level of activity j ($x_j \geq 0$)

k_j = setup cost

c_j = cost for each incremental unit

Suppose: there are n activities

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

How ? Auxiliary variable $y_1, y_2, \dots, y_n$ (Binary)

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

New constraints: y_j can be viewed as *contingent decisions*:

$$x_j \leq M . y_j \quad \forall j = 1,2,\dots,n$$

✓ ***Binary Representation of Integer Variables***

To change a pure IP problem to BIP problem, aimed at using an efficient BIP algorithms

bounds on an integer variable *x* are: $0 \le x \le u$

N is defined as the integer such that: $2^N \le u \le 2^{N+1}$

How ?

binary representation of *x* is: $x = \sum_{i=1}^{N} 2^i y_i$ → Binary

$$\begin{cases} x_1 \le 5 \\ 2x_1 + 3x_2 \le 30 \end{cases}$$

u=5 for x_1 → $x_1 = y_0 + 2y_1 + 4y_2$

u=10 for x_2 → $x_2 = y_3 + 2y_4 + 4y_5 + 8y_6$

$$\begin{cases} y_0 + 2y_1 + 4y_2 \le 5 \\ 2y_0 + 4y_1 + 8y_2 + 3y_3 + 6y_4 + 12y_5 + 24y_6 \le 30 \\ y_i \in \{0,1\} \quad \text{for } i=1,2,\ldots \end{cases}$$

Making Choices When Decision Variables Are Continuous

- We want to produce 3 possible new products.
- We have 2 plants

	Production Time Used for Each Unit Produced			Production Time Available per Week
	Product 1	Product 2	Product 3	
Plant 1	3 hours	4 hours	2 hours	30 hours
Plant 2	4 hours	6 hours	2 hours	40 hours
Unit profit	5	7	3	(thousands of dollars)
Sales potential	7	5	9	(units per week)

Restriction 1:
From 3 possible new products, at most 2 should be chosen to be produced.

Restriction 2:
Just 1 of 2 plants should be chosen to be the sole producer of new products.

	Production Time Used for Each Unit Produced			Production Time Available per Week
	Product 1	Product 2	Product 3	
Plant 1	3 hours	4 hours	2 hours	30 hours
Plant 2	4 hours	6 hours	2 hours	40 hours
Unit profit	5	7	3	(thousands of dollars)
Sales potential	7	5	9	(units per week)

Variables

x_1, x_2, x_3 production rates of products

Mathematical Model

$$Maximize\ Z = 5x_1 + 7x_2 + 3x_3$$

Subject to:

$$3x_1 + 4x_2 + 2x_3 \le 30$$
$$4x_1 + 6x_2 + 2x_3 \le 40$$
$$x_1 \le 7$$
$$x_2 \le 5$$
$$x_3 \le 9$$
$$x_1, x_2, x_3 \ge 0$$

Restriction 1 ?

Restriction 2 ?

Restriction 1:

From 3 possible new products, at most 2 should be chosen to be produced.

The number of strictly positive decision variables (x1, x2, x3) must be ≤ 2.

$$x_1 + x_2 + x_3 \leq 2$$

How ?

3 auxiliary binary variables (y_1, y_2, y_3)

$$y_j = \begin{cases} 1 & \text{if } x_j > 0 \text{ can hold (can produce product } j) \\ 0 & \text{if } x_j = 0 \text{ must hold (cannot produce product } j), \end{cases}$$

$$x_1 \leq My_1$$
$$x_2 \leq My_2$$
$$x_3 \leq My_3$$
$$y_1 + y_2 + y_3 \leq 2$$
$$y_j \text{ binary (for } j \ \ 1, 2, 3)$$

Restriction 2:

Just 1 of 2 plants should be chosen to be the sole producer of new products.

$$\begin{array}{ll} \text{Either} & 3x_1 + 4x_2 + 2x_3 \le 30 \\ \text{or} & 4x_1 + 6x_2 + 2x_3 \le 40 \end{array}$$

How ?

1 auxiliary binary variables (y_4)

$$y_4 = \begin{cases} 1 & \text{if } 4x_1 + 6x_2 + 2x_3 \le 40 \text{ must hold (choose Plant 2)} \\ 0 & \text{if } 3x_1 + 4x_2 + 2x_3 \le 30 \text{ must hold (choose Plant 1).} \end{cases}$$

$$3x_1 + 4x_2 + 2x_3 \le 30 + My_4$$
$$4x_1 + 6x_2 + 2x_3 \le 40 + M(1 - y_4)$$
$$y_4 \text{ binary}$$

Complete Model

$$\text{Maximize} \quad Z = 5x_1 + 7x_2 + 3x_3,$$

subject to

$$\begin{aligned}
x_1 &\le 7 \\
x_2 &\le 5 \\
x_3 &\le 9 \\
x_1 - My_1 &\le 0 \\
x_2 - My_2 &\le 0 \\
x_3 - My_3 &\le 0 \\
y_1 + y_2 + y_3 &\le 2 \\
3x_1 + 4x_2 + 2x_3 - My_4 &\le 30 \\
4x_1 + 6x_2 + 2x_3 + My_4 &\le 40 + M
\end{aligned}$$

and

$$x_1 \ge 0, \quad x_2 \ge 0, \quad x_3 \ge 0$$

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

❖ IP problems have far fewer solutions rather than LP problems.

❖ IP problems with bounded feasible region: finite number of solutions.

2 fallacies:

Fallacy 1 (Number of solutions):

having a finite number of feasible solutions ensures: problem is readily solvable.

Answer:

Finite numbers can be astronomically large.
BIP problem with n variables: 2^n solutions to be considered
(of course, some can be discarded due to violating constraints).
Each time n is increased by 1, the number of solutions is doubled.

- n = 10: about 1,000 solutions (1,024)
- n = 20: about 1,000,000 solutions
- n = 30: about 1,000,000,000 solutions

Exponential growth

Fallacy 2:

removing some feasible solutions (non-integer ones) from a LP problem makes it easier to solve.

Answer:

- To the contrary, it is only because all these feasible solutions are there that the guarantee usually can be given that there will be a corner-point feasible (CPF) solution that is optimal for the overall problem.

- This guarantee is the key to the remarkable efficiency of the simplex method.

For any given IP problem:

The corresponding LP problem commonly is referred to as its ***LP relaxation***.

Special situation

solving IP problem is no more difficult than solving its LP relaxation

When?

when the optimal solution of LP relaxation is integer.
this is optimal for the IP problem as well,
because it is the best solution among all the feasible solutions for the LP relaxation, which includes all the feasible solutions for the IP problem.

Example:

Special types of IP problems:
minimum cost flow problem (with integer parameters)
+
Its children (transportation, assignment, shortest-path, maximum flow problems).

determinants of computational difficulty of LP problem:

number of (functional) constraints is much more important than number of variables.

3 determinants of computational difficulty of IP problem:

(1) Number of integer variables
(2) Whether integer variables are binary or general integer variables
(3) Any special structure in the problem

- Number of constraints is strictly secondary to the other above factors !
- increasing number of constraints may decrease computation time because number of feasible solutions has been reduced.

determinants of computational difficulty of MIP problem:

Number of integer variables (rather than total number of variables)

Rounding ?

when tempting to solve LP relaxation and then rounding noninteger values to integers in the resulting solution.

2 pitfalls

Pitfall 1:

Optimal LP solution is not necessarily feasible after rounded.

Pitfall 2:

No guarantee that the rounded solution will be the optimal integer solution.

Pitfall 1:

Optimal LP solution is not necessarily feasible after rounded.

Maximize $Z = x_2$

Subject to:

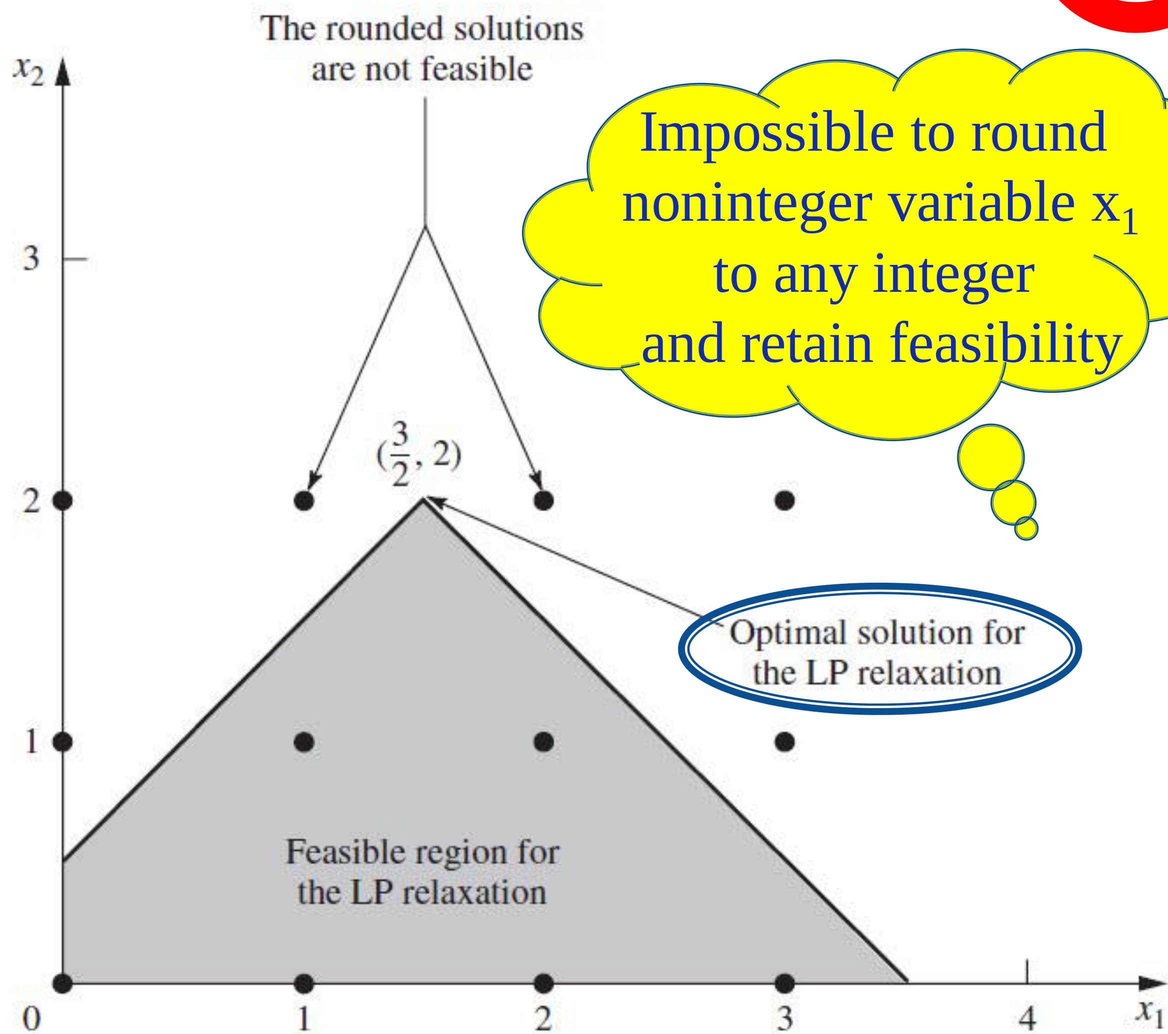
$$-x_1 + x_2 \leq \frac{1}{2}$$

$$x_1 + x_2 \leq 3\frac{1}{2}$$

and

$$x_1 \geq 0, \qquad x_2 \geq 0$$

x_1, x_2 are integers.



Impossible to round noninteger variable x_1 to any integer and retain feasibility

Pitfall 2:

No guarantee that the rounded solution will be the optimal integer solution.

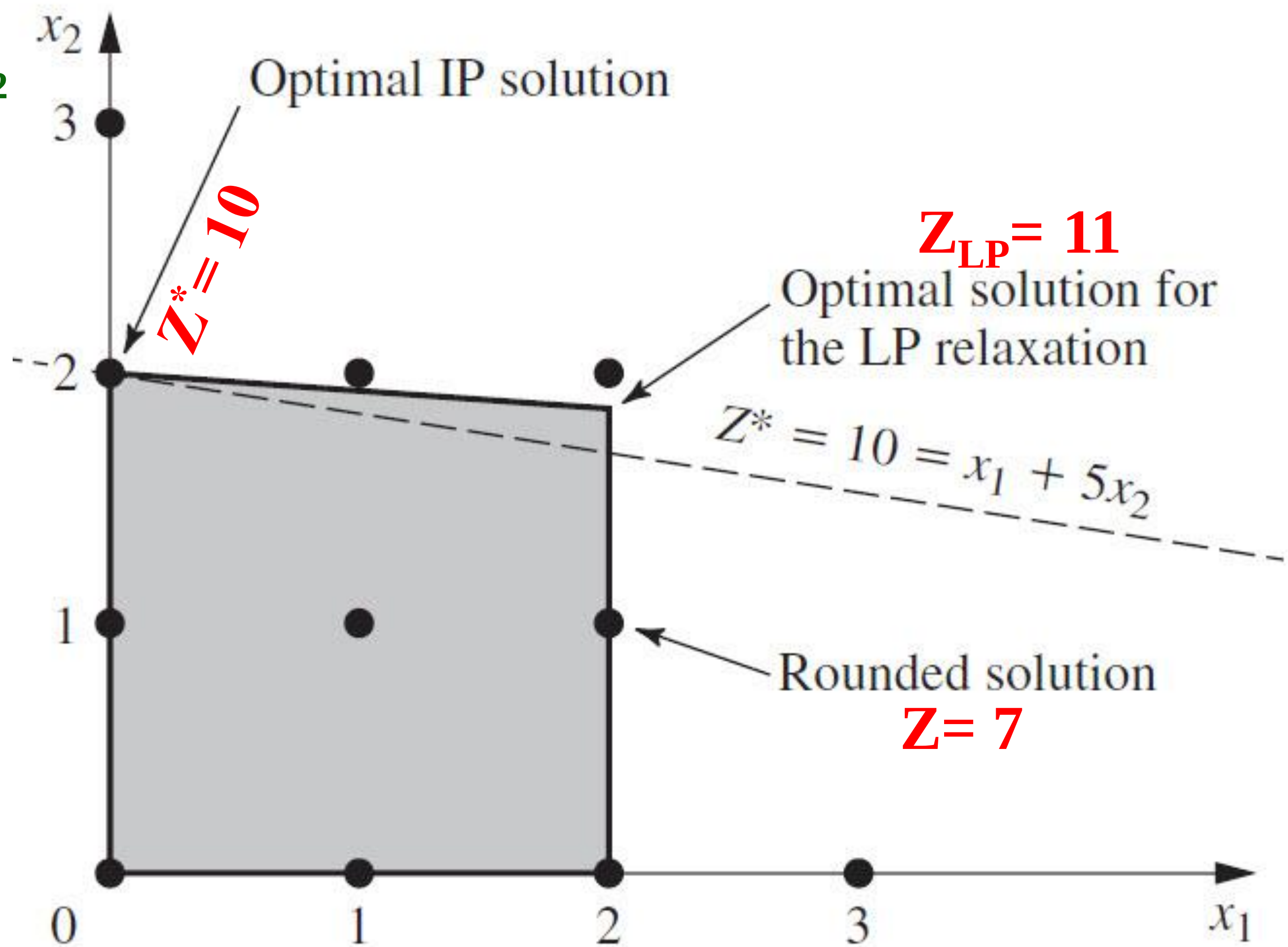
Maximize $Z = x_1 + 5x_2$

Subject to:

$$x_1 + 10x_2 \le 20$$
$$x_1 \le 2$$

and

$$x_1 \ge 0, \quad x_2 \ge 0$$
$$x_1, x_2 \text{ are integers.}$$



Branch-and-Bound Algorithm

A kind of enumeration procedure for finding an optimal solution.

It is imperative that any enumeration procedure be cleverly structured so that only a tiny fraction of feasible solutions need be examined.

3 basic steps of B&B
- Branching
- Bounding
- Fathoming

Example:

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

subject to

$$
\begin{aligned}
&(1) \quad 6x_1 + 3x_2 + 5x_3 + 2x_4 \le 10\\
&(2) \quad x_3 + x_4 \le 1\\
&(3) \quad -x_1 + x_3 \le 0\\
&(4) \quad -x_2 + x_4 \le 0
\end{aligned}
$$

and

$$(5) \quad x_j \text{ is binary, } \quad \text{for } j = 1, 2, 3, 4.$$

1) Branching

Dividing into smaller subproblems.

(binary variables)
the most straightforward way to partition set of feasible solutions into subsets: to fix the value of one of the variables (say, x1): 0 or 1

Subproblem 1: Fix $x_1 = 0$

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

subject to

$$\begin{aligned}
&(1)\quad 3x_2 + 5x_3 + 2x_4 \le 10\\
&(2)\quad x_3 + x_4 \le 1\\
&(3)\quad x_3 \le 0\\
&(4)\quad -x_2 + x_4 \le 0\\
&(5)\quad x_j \text{ is binary,} \quad \text{for } j = 2, 3, 4.
\end{aligned}$$

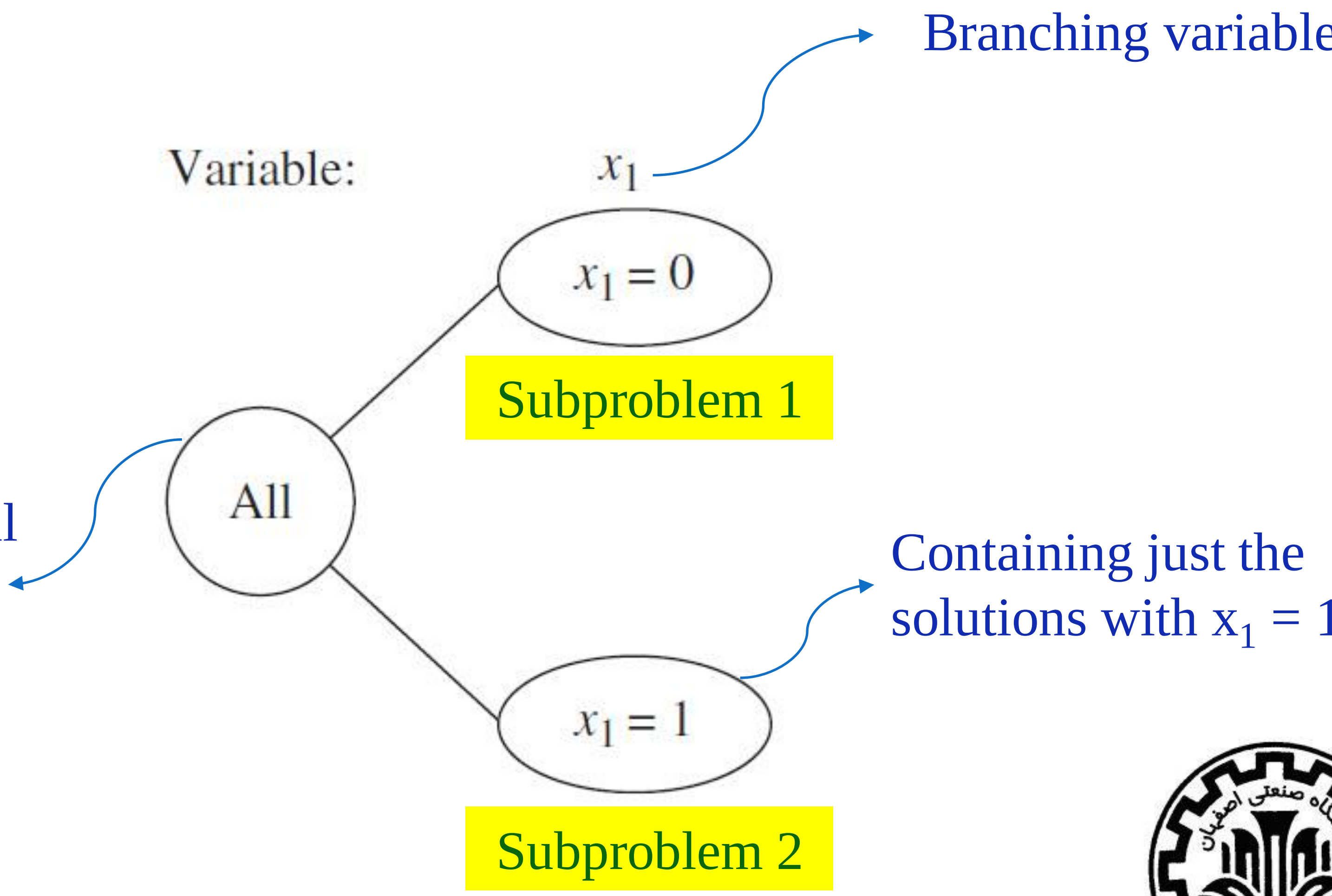
Subproblem 2: Fix $x_1 = 1$

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

subject to

$$\begin{aligned}
&(1)\quad 3x_2 + 5x_3 + 2x_4 \le 4\\
&(2)\quad x_3 + x_4 \le 1\\
&(3)\quad x_3 \le 1\\
&(4)\quad -x_2 + x_4 \le 0\\
&(5)\quad x_j \text{ is binary,} \quad \text{for } j = 2, 3, 4.
\end{aligned}$$

1) Branching


Branching variable
Variable:
x_1
$x_1 = 0$
Subproblem 1
All
Containing all solutions
Containing just the solutions with $x_1 = 1$
$x_1 = 1$
Subproblem 2

2) Bounding

For each of these subproblems, we now need to obtain a bound on how good its best feasible solution can be.

Standard way of Bounding:

To solve a simpler relaxation of the subproblem.

- In most cases, relaxation by deleting ("relaxing") one set of constraints that had made the problem difficult to solve.
- For IP problems: integrality constraints
- Most widely used relaxation for IP: "LP relaxation"

for the example, relaxation for constraint (5):

(5) $0 \le x_j \le 1, \quad \text{for } j = 1, 2, 3, 4.$

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 $j = 1, 2, 3, 4.$

2) Bounding

Using the simplex method to quickly solve this LP relaxation yields its optimal solution:

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

Bound for whole problem: $Z \leq 16$

obtain the bounds for the two subproblems:

Subproblem 1: at $x_1 = 0$,

this can be conveniently expressed in its LP relaxation by adding the constraint that $x_1 \leq 0$

since combining with $0 \leq x_1 \leq 1$ forces $x_1 = 0$.

Subproblem 1: at $x_1 = 1$,

this can be conveniently expressed in its LP relaxation by adding the constraint that $x_1 \geq 1$

since combining with $0 \leq x_1 \leq 1$ forces $x_1 = 1$.

3) Fathoming

F. Hillier, G. J. Lieberman, "Introduction to Operations Research", Ninth Edition, 2010.