Integer Programming (Modeling)

Based on Chapter 9 of

Operations Research: Applications and Algorithms, 4th edition

1. Introduction to Integer Programming

- Recall that one of the four basic assumptions of LP is Divisibility.
- An *integer (linear) programming problem* (IP) is an LP in which some or all of the variables are required to be non-negative integers.
- An IP in which all variables are required to be integers is call a **pure integer programming** problem.

Example:

$$\begin{aligned} \max \quad & z = 3x_1 + 2x_2 \\ \text{s.t.} \quad & x_1 + x_2 \le 6 \\ & x_1, x_2 \ge 0,\ x_1, x_2 \text{ integer} \end{aligned}$$

Integer Programming (cont'd)

- An IP in which only some of the variables are required to be integers is called a **mixed integer programming (MIP) problem**.

Example:

$$\begin{aligned} \max \quad & z = 3x_1 + 2x_2 \\ \text{s.t.} \quad & x_1 + x_2 \le 6 \\ & x_1, x_2 \ge 0, x_1 \text{ integer} \end{aligned}$$

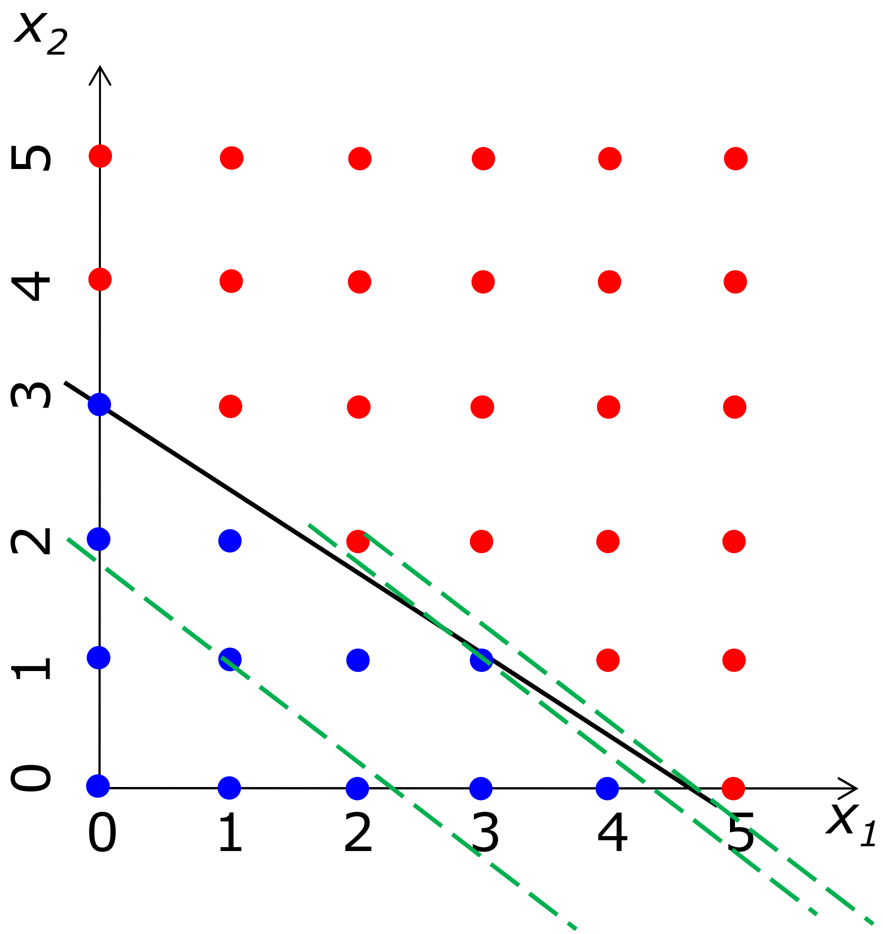
- An integer programming problem in which all the variables must be 0 or 1 is called a **0-1 IP**.

Example: (knapsack problem)

$$\begin{aligned} \max z = {} & 16x_1 + 22x_2 + 12x_3 + 8x_4 \\ \text{s.t.} \quad & 5x_1 + 7x_2 + 4x_3 + 3x_4 \le 14 \\ & x_j = 0 \text{ or } 1 \quad (j = 1,2,3,4) \end{aligned}$$

Integer Programming

Feasible region:



$$\begin{aligned} \max \quad & z = 3x_1 + 4x_2 \\ \text{s.t.} \quad & 5x_1 + 8x_2 \le 24 \\ & x_1, x_2 \ge 0, x_1, x_2 \text{ integer} \end{aligned}$$

What is the optimal integer solution? $x_1=3, x_2=1, z=13$

What is the optimal linear solution (ignore integrality)?
$x_1=24/5, x_2=0, z=72/5$

Round, get $x=5$, $y=0$, **infeasible!**

Truncate, get $x=4$, $y=0$, and $z=12$

LP Relaxation

- The LP obtained by omitting all integer or 0-1 constraints on variables is called **LP relaxation** of the IP.
  - Any IP may be viewed as LP relaxation plus additional constraints (the constraints that state which variables must be integers or be 0 or 1).
  - The feasible region for any IP must be contained in the feasible region for the corresponding LP relaxation.
  - For any max IP:

    Optimal z-value for LP relaxation $\geq$ optimal z-value for IP

    For any min IP:

    Optimal z-value for LP relaxation $\leq$ optimal z-value for IP

2. Formulating IP Problems

Example 1. Capital Budgeting IP

Stockco is considering four investments. Investment 1 will yield a net present value (NPV) of \$16000; investment 2, an NPV of \$22000; investment 3, an NPV of \$12000; and investment 4, an NPV of \$8000. Each investment requires a certain cash outflow at the present time: investment 1, \$5000; investment 2, \$7000; investment 3, \$4000; and investment 4, \$3000. Currently, \$14000 is available for investment. Formulate an IP whose solution will tell Stockco how to maximize the NPV obtained from investments 1-4.

Example 1: Solution

- Begin by defining a variable for each decision that Stockco must make.

$$x_j = \begin{cases} 1 & \text{if investment } j \text{ is made} \\ 0 & \text{otherwise} \end{cases}$$
$$(j = 1,2,3,4)$$

- The NPV obtained by Stockco (objective function) is

  Total NPV obtained by Stockco = $16x_1 + 22x_2 + 12x_3 + 8x_4$

- Stockco faces the constraint that at most \$14,000 can be invested.

- Stockco's 0-1 IP – ***Knapsack problem***

$$\begin{aligned} \max z &= 16x_1 + 22x_2 + 12x_3 + 8x_4 \\ \text{s.t.} \quad & 5x_1 + 7x_2 + 4x_3 + 3x_4 \leq 14 \\ & x_j = 0 \text{ or } 1 \quad (j = 1,2,3,4) \end{aligned}$$

Example 1: Cont'd

- By using the binary variables and logical constraints, the model can incorporate many complex interactions between projects (activities). The binary variables could be used for investment analysis, site selection, supply chain design, dispatching, scheduling, etc.

Example 1: Cont'd

Modify the Stockco formulation to account for the constraints:

1. a. Stockco can invest in at most two investments
   b. Invest in exactly two investments
   c. Mutually exclusive investments (exactly one alternative must be selected)
2. Contingency: (Investment 2 could be selected only if investment 1 is selected.)
3. If Stockco invests in investment 2, then they must also invest in investment 1.
4. If Stockco invests in investment 2, then they cannot invest in investment 4.
5. Either investment 3 is selected or investment 4 is selected, but not both.

Example 2. Fixed-Charge Problems

Gandhi Cloth Company is capable of manufacturing three types of clothing: shirts, shorts, and pants. The manufacture of each type of clothing requires that Gandhi have the appropriate type of machinery available. The machinery needed to manufacture each type of clothing must be rented at the following rates: shirt machinery, $200 per week; shorts machinery, $150 per week; pants machinery, $100 per week. The manufacture of each type of clothing also requires the amounts of cloth and labor given in the Table 1. Each week 150 hours of labor and 160 sq yd of cloth available. The variable unit cost and selling price for each type of clothing are shown in Table 2. Formulate an IP whose solution will maximize Gandhi's weekly profits.

Example 2. Cont’d

Table 1. Resource requirements

	Labor (Hours)	Cloth (sq yd)
Shirt	3	4
Shorts	2	3
Pants	6	4

Table 2. Selling price and variable cost

	Price	Variable Cost
Shirt	$12	$6
Shorts	$8	$4
Pants	$15	$8

Example 2: Solution

- Gandhi must decide how many of each type of clothing should be manufactured each week,

$$x_1 = \text{number of shirts produced each week}$$

$$x_2 = \text{number of shorts produced each week}$$

$$x_3 = \text{number of pants produced each week}$$

- The cost of renting machinery depends only on the types of clothing produced, not on the amount of each type of clothing.

Example 2: Solution (cont'd)

- The cost of renting machinery depends only on the types of clothing produced, not on the amount of each type of clothing.

$$y_1 = \begin{cases} 1 & \text{if the machine is rented} \\ & \text{(i.e., any shirts are manufactured)} \\ 0 & \text{otherwise} \end{cases}$$

$$y_2 = \begin{cases} 1 & \text{if any shorts are manufactured} \\ 0 & \text{otherwise} \end{cases}$$

$$y_3 = \begin{cases} 1 & \text{if any pants are manufactured} \\ 0 & \text{otherwise} \end{cases}$$

- In short, if $x_j > 0$, then $y_j = 1$.

Example 2: Solution (cont'd)

- Weekly profits:

$$z = (12x_1 + 8x_2 + 15x_3) - (6x_1 + 4x_2 + 8x_3)$$
$$- (200y_1 + 150y_2 + 100y_3)$$
$$= 6x_1 + 4x_2 + 7x_3 - 200y_1 - 150y_2 - 100y_3$$

- Labor constraint:

$$3x_1 + 2x_2 + 6x_3 \leq 150$$

- Cloth constraint:

$$4x_1 + 3x_2 + 4x_3 \leq 160$$

Example 2: Solution (cont'd)

- If $x_j>0$, then $y_j=1$. If $x_j=0$, then $y_j=0$ and $y_j=1$.

$$x_1 \le M_1 y_1$$

$$x_2 \le M_2 y_2$$

$$x_3 \le M_3 y_3$$

M_1, M_2, and M_3 are three large positive numbers.

In general, M_i should be set equal to the maximum value that x_i can attain: $M_1=40$, $M_2=53$, $M_3=25$

Example 3. Set-Covering Problems

There are six cities (cities 1-6) in Kilroy County. The county must determine where to build fire stations. The county wants to build the minimum number of fire stations needed to ensure that at least one fire station is within 15 minutes (driving time) of each city. The times (in minutes) required to drive between the cities in Kilroy County are shown in Table 3. Formulate an IP that will tell Kilroy how many fire stations should be built and where they should be located.

Example 3: cont'd

Table 3. Time required to travel between cities

FROM	TO					
	City 1	*City 2*	*City 3*	*City 4*	*City 5*	*City 6*
City 1	0	10	20	30	30	20
City 2	10	0	25	35	20	10
City 3	20	25	0	15	30	20
City 4	30	35	15	0	15	25
City 5	30	20	30	15	0	14
City 6	20	10	20	25	14	0

Example 3: Solution

- For each city, Kilroy must determine whether or not to build a fire station in that city. We define the 0-1 variables x_i (i=1,…,6) by

$$x_i = \begin{cases} 1 & \text{if a fire station is built in city } i \\ 0 & \text{otherwise} \end{cases}$$

Then the total number of fire station that are built is given by $x_1+x_2+x_3+x_4+x_5+x_6$

Example 3: Solution (cont'd)

- What are the constraints? Kilroy must ensure that there is a fire station within 15 minutes of each city. Table 4 indicates which locations can reach the city in 15 minutes or less.

Table 4. Cities within 15 minutes of given city

City 1	1,2
City 2	1,2,6
City 3	3,4
City 4	3,4,5
City 5	4,5,6
City 6	2,5,6

To ensure that at least one fire station is within 15 minutes of city 1, we add $x_1 + x_2 \geq 1$ (City 1 constraint)

Example 3: Solution (cont'd)

$$x_i = \begin{cases} 1 & \text{if a fire station is built in city } i \\ 0 & \text{otherwise} \end{cases}$$

$$\begin{aligned}
\min \quad & z = x_1 + x_2 + x_3 + x_4 + x_5 + x_6 \\
\text{s.t.} \quad & x_1 + x_2 \geq 1 \\
& x_1 + x_2 + x_6 \geq 1 \\
& x_3 + x_4 \geq 1 \\
& x_3 + x_4 + x_5 \geq 1 \\
& x_4 + x_5 + x_6 \geq 1 \\
& x_2 + x_5 + x_6 \geq 1 \\
& x_i = 0 \text{ or } 1 \ (i = 1,2,3,4,5,6)
\end{aligned}$$

Example 3. Set-Covering Problems

- In a **set-covering problem**, each member of a given set (Set 1) must be "covered" by an acceptable member of some set (Set 2).
- The objective of a set-covering problem is to minimize the number of elements in Set 2 that are required to cover all the elements in Set 1.
- In the fire station example, Set 1 is the cities in Kilroy County, and Set 2 is the set of fire stations.

EITHER-OR Constraints

- We are given two constraints of the form

$$f(x_1, x_2, \ldots, x_n) \leq 0$$
$$g(x_1, x_2, \ldots, x_n) \leq 0$$

- We wish to ensure that one of the two constraints is satisfied, often called either-or constraints. Adding the following two constraints to the formulation:

$$f(x_1, x_2, \ldots, x_n) \leq My$$
$$g(x_1, x_2, \ldots, x_n) \leq M(1-y)$$

  y is a 0-1 variable, and M is a number large enough.

- **Example 1 cont'd**: The number of investments selected is not two.

Example 4. EITHER-OR Constraints

Dorian Auto is considering manufacturing three types of autos: compact, midsize, and large. The resources required for, and the profits yielded by, each type of car are shown below. Currently, 6000 tons of steel and 60000 hours of labor are available. For production of a type of car to be economically feasible, at least 1000 cars of that type must be produced. Formulate an IP to maximize Dorian's profit.

Resource	Car Type		
	Compact	Midsize	Large
Steel required	1.5 tons	3 tons	5 tons
Labor required	30 hours	25 hours	40 hours
Profit yielded	2000	3000	4000

Example 4: Solution

- Because Dorian must determine how many cars of each type should be built, we define

  x_1 = number of compact cars produced

  x_2 = number of midsize cars produced

  x_3 = number of large cars produced

- Objective function (in thousands of dollars)

  $z = 2x_1+3x_2+4x_3$

Example 4: Solution (cont'd)

- Constraints:

1: $x_1 \le 0$ or $x_1 \ge 1000$

2: $x_2 \le 0$ or $x_2 \ge 1000$

3: $x_3 \le 0$ or $x_3 \ge 1000$

4: The cars produced can use at most 6000 tons of steel

5: The cars produced can use at most 60000 hours of labor

Example 4: Solution (cont'd)

- Constraint 1: $x_1 \leq 0$ or $x_1 \geq 1000$

Define: $f(x_1, x_2, x_3) = x_1$ and $g(x_1, x_2, x_3) = 1000 - x_1$

We can replace constraint 1 by

$$x_1 \leq M_1 y_1$$

$$1000 - x_1 \leq M_1 (1 - y_1)$$

$$y_1 = 0 \text{ or } 1$$

We may choose $M_1 = 2000$ (Why?)

K out of N Constraints Must Hold

- An extension of Either-Or Constraints
- Consider the case where the overall model includes a set of N possible constraints such that only some K of these constraints must hold ($K < N$). Part of the optimization process is to choose the combination of K constraints that permits the objective function to reach its best possible value. The N - K constraints not chosen are, in effect, eliminated from the problem (become **redundant**).
- Given a set of N constraints (left), we formulate it as the right one:

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

- y is a 0-1 variable, and M is a number large enough. Note that y_i =0 makes My_i=0, which reduces the new constraint i to the original constraint i. When y_i =1 makes (d_i+My_i) so large that the new constraint i is redundant

Functions with N Possible Values

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

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

- Equivalent :

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

$$\sum_{i=1}^{N} y_i = 1$$

y_i is a 0-1 variable

- In this case, there are N yes-or-no questions being asked, namely, should *di* be the value chosen $(i = 1, 2, \ldots, N)$?

- Example: In order to leave any remaining capacity in usable blocks for these future products, management now wants to impose the restriction that the production time used by the two current new products be 6 *or 12 or 18 hours* per week. Thus, the third constraint of the original model ($3x_1+2x_2<=18$) now becomes $3x_1+2x_2$ =6 or 12 or 18. Equivalently,

$$3x_1 + 2x_2 = 6y_1 + 12y_2 + 18y_3$$
$$y_1 + y_2 + y_3 = 1$$

IF-THEN Constraints

In many applications, we want to ensure that

- if a constraint $f(x_1, x_2, \ldots, x_n) > 0$ is satisfied, then the constraint $g(x_1, x_2, \ldots, x_n) \geq 0$ must be satisfied,
- In other words, $f(x_1, x_2, \ldots, x_n) > 0$ and $g(x_1, x_2, \ldots, x_n) < 0$ can not happen at the same time.
- To ensure this, we include the following constraints:

$$-g(x_1, x_2, \ldots, x_n) \leq M(1-y)$$
$$f(x_1, x_2, \ldots, x_n) \leq My$$
$$y = 0 \text{ or } 1$$

Example 1 (cont'd):

(a) You must select at least one of investments 1 and 2 unless the NPV of the portfolio exceeds \$12,000

If $12 - NPV > 0$, then $x_1 + x_2 \geq 1$.

Add the constraints: $-(x_1 + x_2 - 1) \leq M(1 - y)$

$$12 - NPV \leq My$$

$$y = 0 \text{ or } 1$$

M=12 and recall that NPV = $16x_1 + 22x_2 + 12x_3 + 8x_4$

(b) If investment 1 and investment 2 are both selected, then investment 3 must be selected.

Representing Non-linear functions

- 0-1 variables can be used to model optimization problems involving piecewise linear functions.
- A **piecewise linear function** consists of several straight line segments.
- The points where the slope of the piecewise linear function changes are called the **break points** of the function.
- A piecewise linear function is not a linear function so linear programming can not be used to solve the optimization problem.
- If a piecewise linear function f(x) involved in a formulation has the property that the slope of the f(x) becomes less favorable to the decision maker as x increases (i.e., convex for min problems and concave for max problems), then the tedious IP formulation is unnecessary.

- In order to build up an LP model, all of cost functions must be linear in objective function. Thus, we assume that all related cost functions are linear.
- Convex piecewise linear function is OK too.

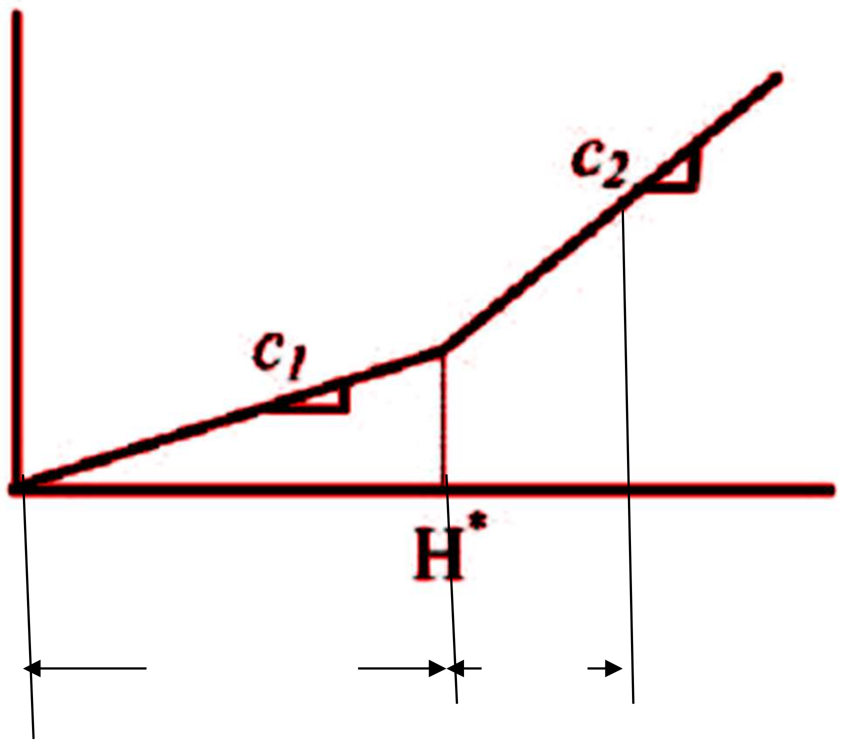
$$\min \quad + \sum_{t=1}^{\bar{t}} (c_1 H_{1t} + c_2 H_{2t})$$

$$\text{s.t.} \quad H_t = H_{1t} + H_{2t}$$

$$0 \le H_{1t} \le H^*$$

$$0 \le H_{2t}$$

$$c_1 < c_2$$



- By using 0-1 variables, however, a piecewise linear function can be represented in linear form.
- Suppose the piecewise linear function $f(x)$ has break points $b_1, b_2, \ldots, b_n$:
  - **Step 1** Wherever $f(x)$ occurs in the optimization problem, replace $f(x)$ with $z_1 f(b_1) + z_2 f(b_2) + \cdots + z_n f(b_n)$.
  - **Step 2** Add the following constraints to the problem:

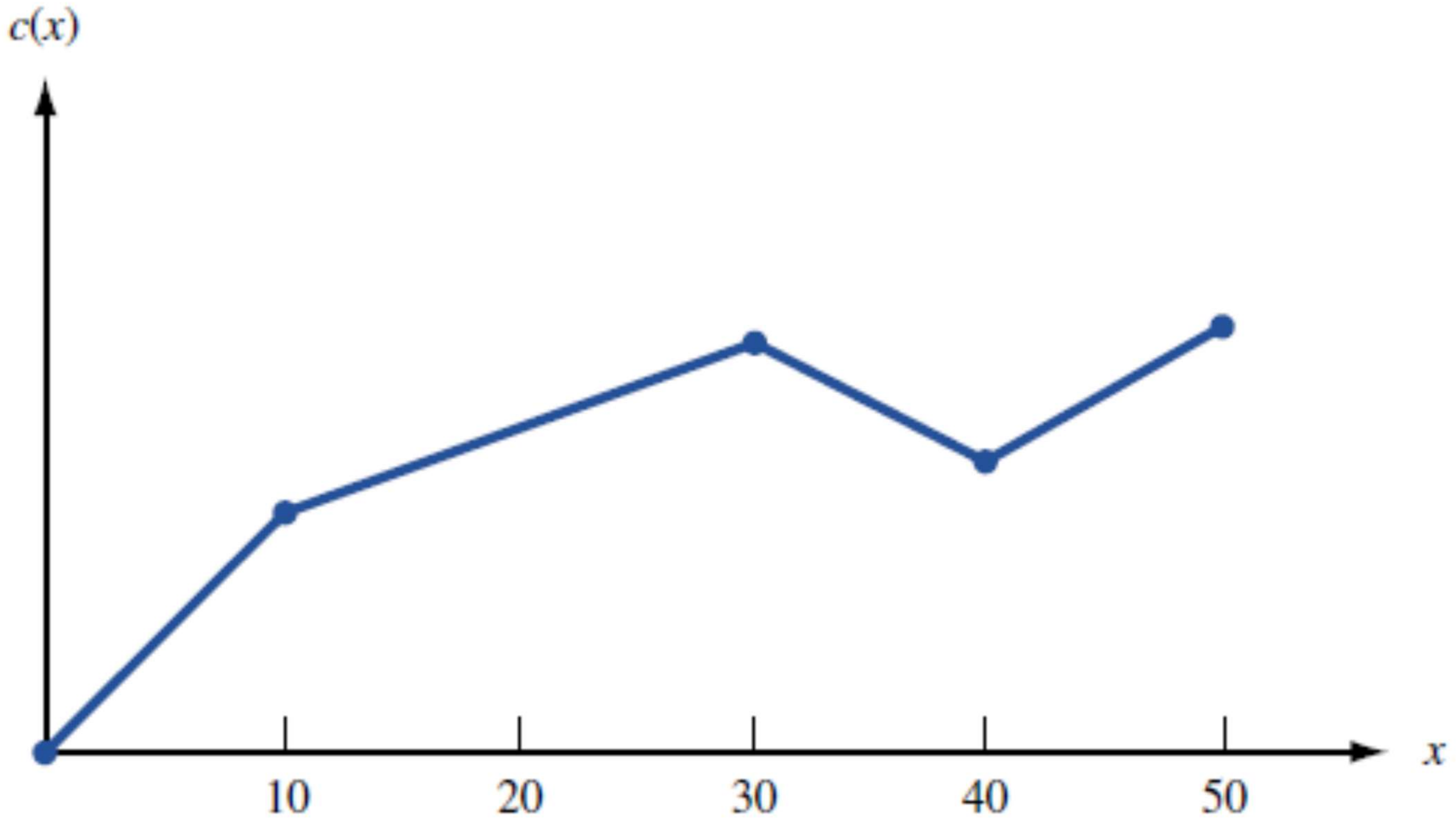
$$y_i = 0 \text{ or } 1 \text{ for } i = 1,2,\ldots,n-1; \; z_i \geq 0 \text{ for } i = 1,2,\ldots,n$$

$$y_1 + y_2 + \cdots + y_{n-1} = 1,$$

$$z_1 \leq y_1, z_2 \leq y_1 + y_2, z_3 \leq y_2 + y_3, \ldots, z_{n-1} \leq y_{n-2} + y_{n-1}, z_n \leq y_{n-1},$$

$$z_1 + z_2 + \cdots + z_n = 1,$$

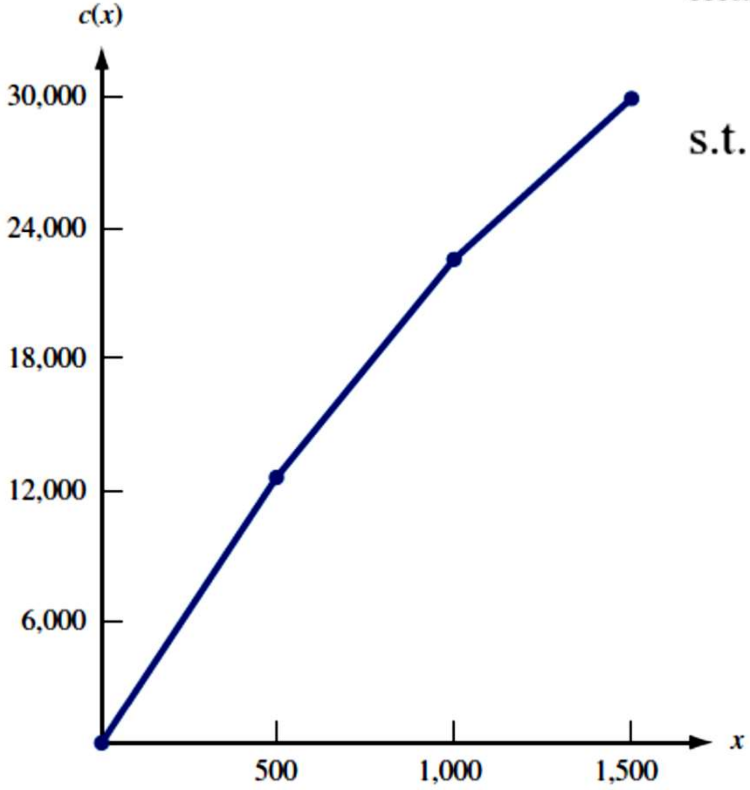
$$x = z_1 b_1 + z_2 b_2 + \cdots + z_n b_n,$$


c(x)
x
10
20
30
40
50

- Euing Gas produces two types of gasoline (gas 1 and gas 2) from two types of oil (oil 1 and oil 2). Each gallon of gas 1 must contain at least 50 percent oil 1, and each gallon of gas 2 must contain at least 60 percent oil 1.
- 500 gallons of oil 1 and 1,000 gallons of oil 2 are available.
- Each gallon of gas 1 can be sold for 12¢, and each gallon of gas 2 can be sold for 14¢.
- As many as 1,500 more gallons of oil 1 can be purchased at the following prices: first 500 gallons, 25¢ per gallon; next 500 gallons, 20¢ per gallon; next 500 gallons, 15¢ per gallon.
- Formulate an IP that will maximize Euing's profits (revenues - purchasing costs).

x = amount of oil 1 purchased

x_{ij} = amount of oil i used to produce gas j $\quad (i, j = 1, 2)$



$$c(x) = \begin{cases} 25x & (0 \le x \le 500) \\ 20x + 2{,}500 & (500 \le x \le 1{,}000) \\ 15x + 7{,}500 & (1{,}000 \le x \le 1{,}500) \end{cases}$$

$$
\begin{aligned}
\max z = {} & 12x_{11} + 12x_{21} + 14x_{12} + 14x_{22} - z_1 c(0) - z_2 c(500) \\
& - z_3 c(1{,}000) - z_4 c(1{,}500) \\
\text{s.t.} \quad & x_{11} + x_{12} \le x + 500 \\
& x_{21} + x_{22} \le 1{,}000 \\
& 0.5x_{11} - 0.5x_{21} \ge 0 \\
& 0.4x_{12} - 0.6x_{22} \ge 0 \\
& x = 0z_1 + 500z_2 + 1{,}000z_3 + 1{,}500z_4 \\
& z_1 \le y_1 \\
& z_2 \le y_1 + y_2 \\
& z_3 \le y_2 + y_3 \\
& z_4 \le y_3 \\
& y_1 + y_2 + y_3 = 1 \\
& z_1 + z_2 + z_3 + z_4 = 1 \\
& y_i = 0 \text{ or } 1 \quad (i = 1, 2, 3);\ z_i \ge 0 \quad (i = 1, 2, 3, 4) \\
& x_{ij} \ge 0
\end{aligned}
$$

Integer Programming (Algorithms)

Outline:

- Branch-and-Bound (B&B) for solving pure IP
- B&B for solving mixed IP
- Heuristics for solving TSP
- Cutting plane methods

Based on Chapter 9.3-9.6 of
Operations Research: Applications and Algorithms, 4th edition

3. B&B Method for Solving Pure IP Problems

- Recall: For any max IP:

  Optimal z-value for LP relaxation

  $\geq$ optimal z-value for IP

- Important observation: If you solve the LP relaxation of a pure IP and obtain a solution in which all variables are integers, then the optimal solution to the LP relaxation is also the optimal solution to the IP

B&B Method for Solving Pure IP Problems

- In practice, most IPs are solved by some versions of the branch-and-bound procedure. Branch-and-bound methods implicitly enumerate all possible solutions to an IP.
- Fathoming rules: a procedure to determine if a subset of feasible solutions can be discarded without optimal solution. By solving a single subproblem, many possible solutions may be eliminated from consideration.
  - Lower and upper bounds
- branching rules: e.g., subproblems are generated by branching on an appropriately chosen fractional-valued variable.
- Node selection rules: e.g., LIFO (last-in-first-out), which subproblem to select next

Example 1

- Using B&B method to solve the following pure IP:

$$\max z = 8x_1 + 5x_2$$

$$\text{s.t.} \quad x_1 + x_2 \le 6$$

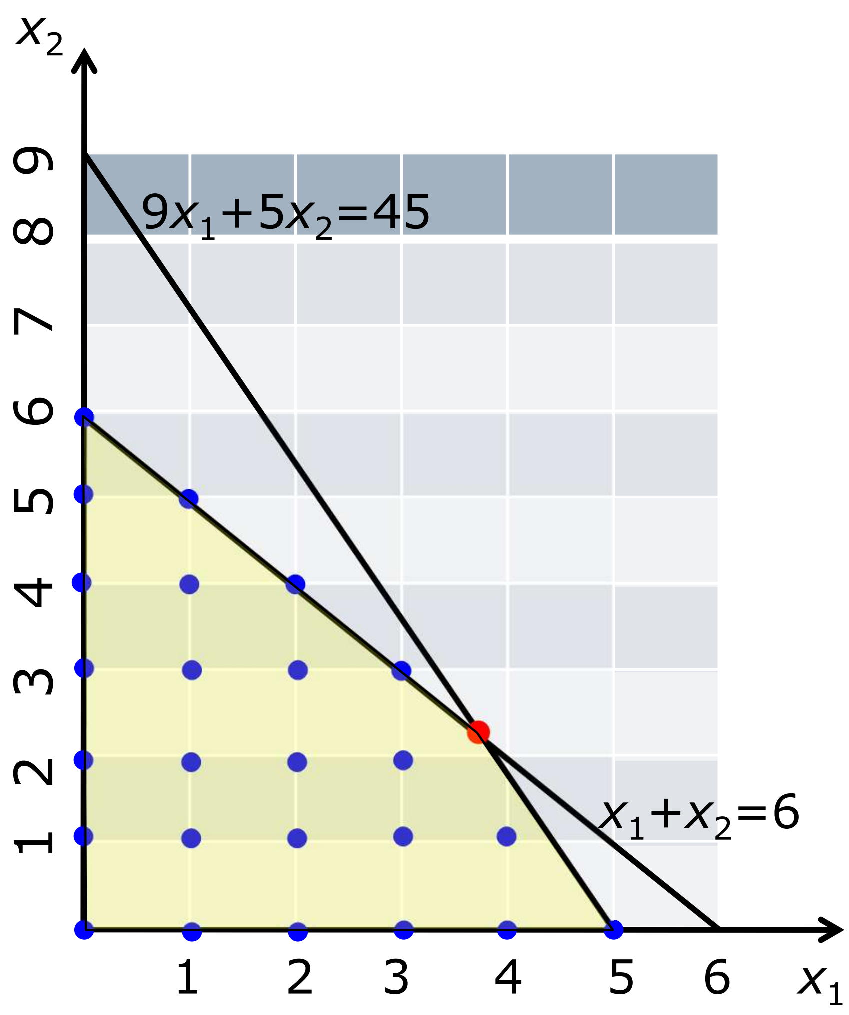
$$9x_1 + 5x_2 \le 45$$

$$x_1, x_2 \ge 0;\ x_1, x_2 \text{ integer}$$

Example 1: Cont'd

- Subproblem 1: the LP relaxation

The B&B method begins by solving the LP relaxation of the IP. If all the variables assume integer values in the optimal solution to the LP relaxation, then the optimal solution to the LP relaxation is also the optimal solution to the IP.

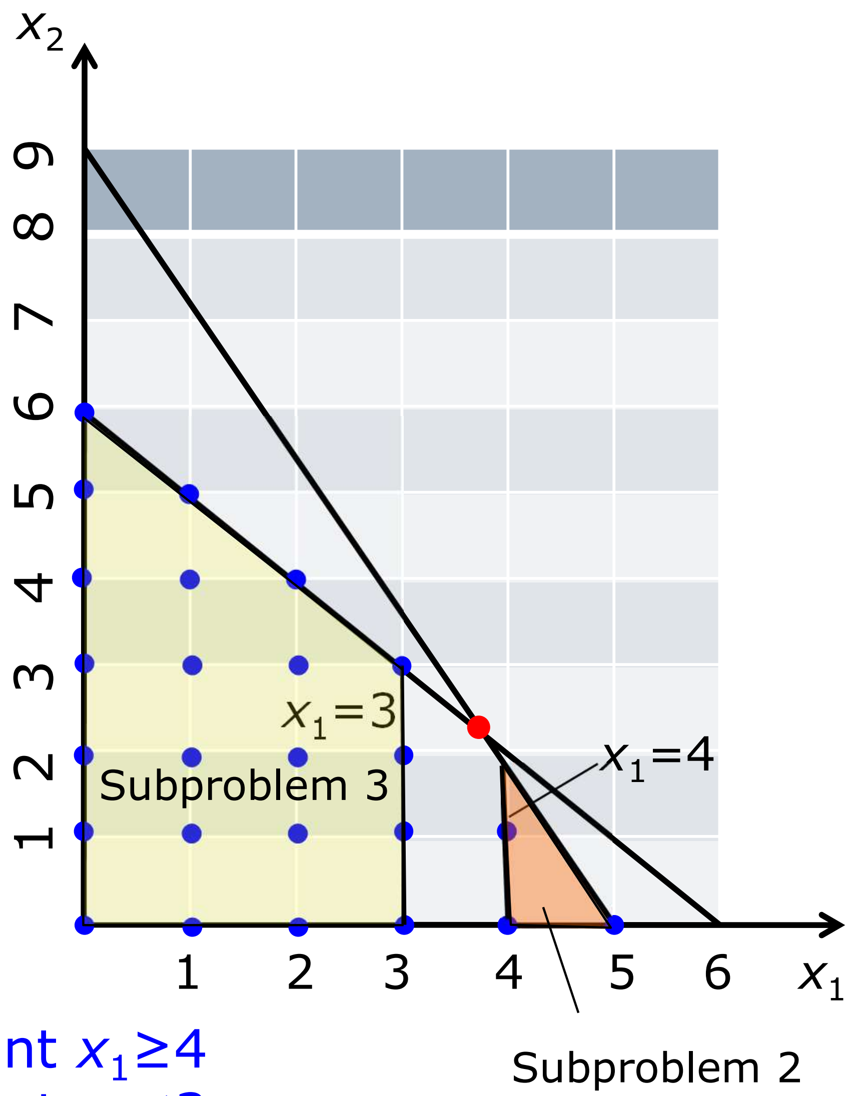


Optimal solution to subproblem 1:
x_1=3.75,x_2=2.25,z=41.25
Upper bound for z value of the IP is 41.25.

Example 1: Cont'd

- Subproblems 2 and 3:
  Branch on x_1.

Our next step is to partition the feasible region for the LP relaxation in an attempt to find out more about the location of the IP's optimal solution. We arbitrarily choose a variable that is fractional in the optimal solution to subproblem 1 – say x_1.



Subproblem 2: Subproblem 1 + constraint $x_1 \geq 4$
Subproblem 3: Subproblem 1 + constraint $x_1 \leq 3$

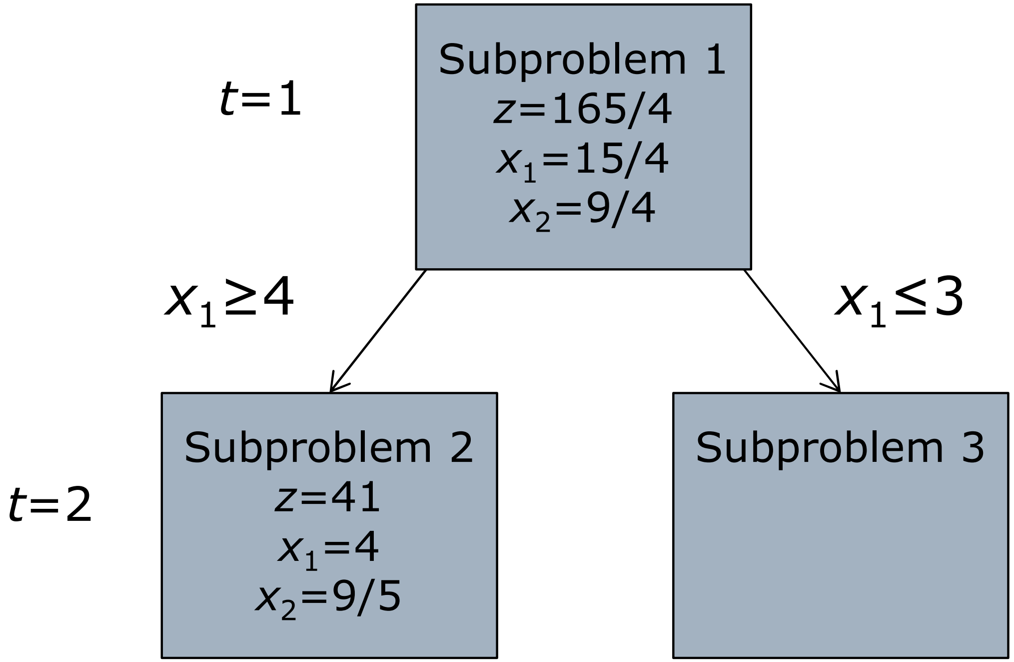
Example 1: Cont'd

Subproblem 2:

$$\begin{aligned}
&\max z = 8x_1 + 5x_2 \\
&\text{s.t.} \quad x_1 + x_2 \le 6 \\
&\qquad 9x_1 + 5x_2 \le 45 \\
&\qquad x_1 \ge 4 \\
&\qquad x_1, x_2 \ge 0
\end{aligned}$$

Subproblem 3:

$$\begin{aligned}
&\max z = 8x_1 + 5x_2 \\
&\text{s.t.} \quad x_1 + x_2 \le 6 \\
&\qquad 9x_1 + 5x_2 \le 45 \\
&\qquad x_1 \le 3 \\
&\qquad x_1, x_2 \ge 0
\end{aligned}$$



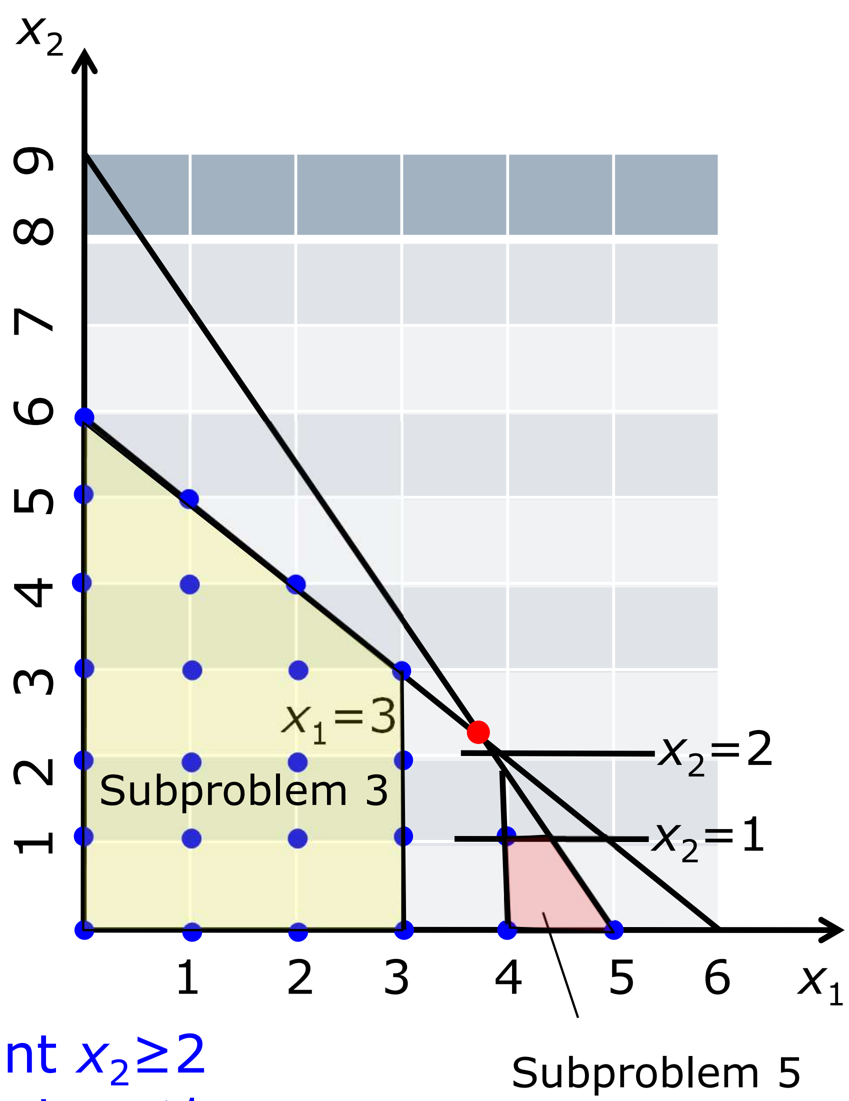
We now choose any subproblem that has not been solved yet. We arbitrarily choose to solve subproblem 2.

Optimal solution to subproblem 2:
$x_1=4, x_2=9/5, z=41$

Example 1: Cont'd

- Subproblems 4 and 5:

The optimal solution to subproblem 2 did not yield an all-integer solution, so we choose to use subproblem 2 to create two new subproblems. We choose a fractional-valued variable in the optimal solution to subproblem 2 and then branch on that variable – x_2.



Subproblem 4: Subproblem 2 + constraint $x_2 \geq 2$
Subproblem 5: Subproblem 2 + constraint $x_2 \leq 1$

Example 1: Cont'd

Subproblem 4:

$$\begin{aligned} \max\ z = 8x_1 + 5x_2 & \\ \text{s.t.}\quad x_1 + x_2 &\le 6 \\ 9x_1 + 5x_2 &\le 45 \\ x_1 &\ge 4 \\ x_2 &\ge 2 \\ x_1, x_2 &\ge 0 \end{aligned}$$

Subproblem 5:

$$\begin{aligned} \max\ z = 8x_1 + 5x_2 & \\ \text{s.t.}\quad x_1 + x_2 &\le 6 \\ 9x_1 + 5x_2 &\le 45 \\ x_1 &\ge 4 \\ x_2 &\le 1 \\ x_1, x_2 &\ge 0 \end{aligned}$$

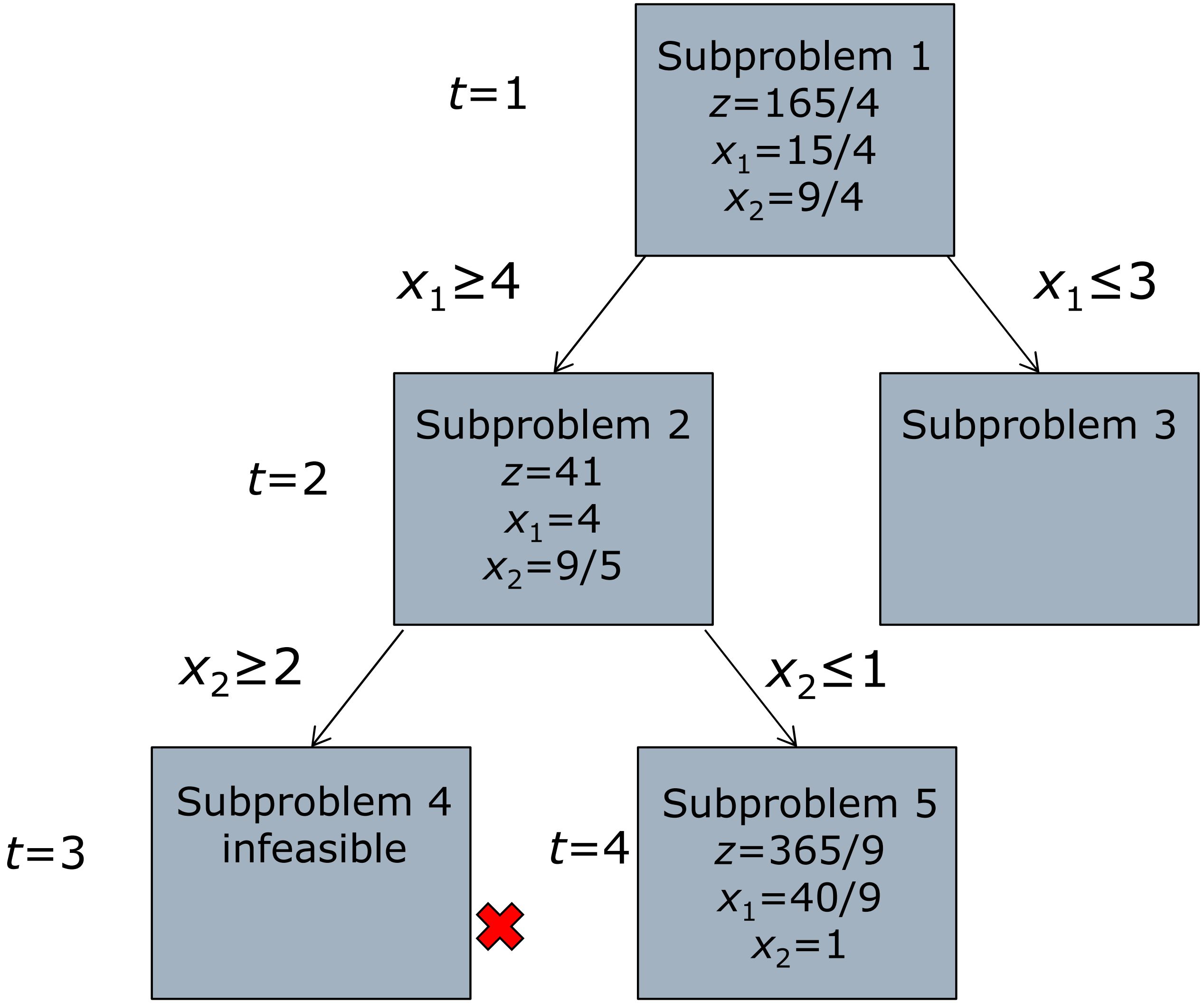
The set of unsolved subproblems consists of subproblems 3,4, and 5.

We choose to solve the most recently created subproblem (LIFO rule), i.e., subproblems 4 and 5 should be solved next.

We arbitrarily choose to solve subproblem 4, which is infeasible. Thus, subproblem 4 cannot yield the optimal solution to the IP. Futher branching on subproblem 4 cannot yield any useful information. We say that subproblem 4 is fathomed.

Next we should solve subproblem 5 according to LIFO rule. The optimal solution to subproblem 5 is x_1=40/9,x_2=1,z=365/9

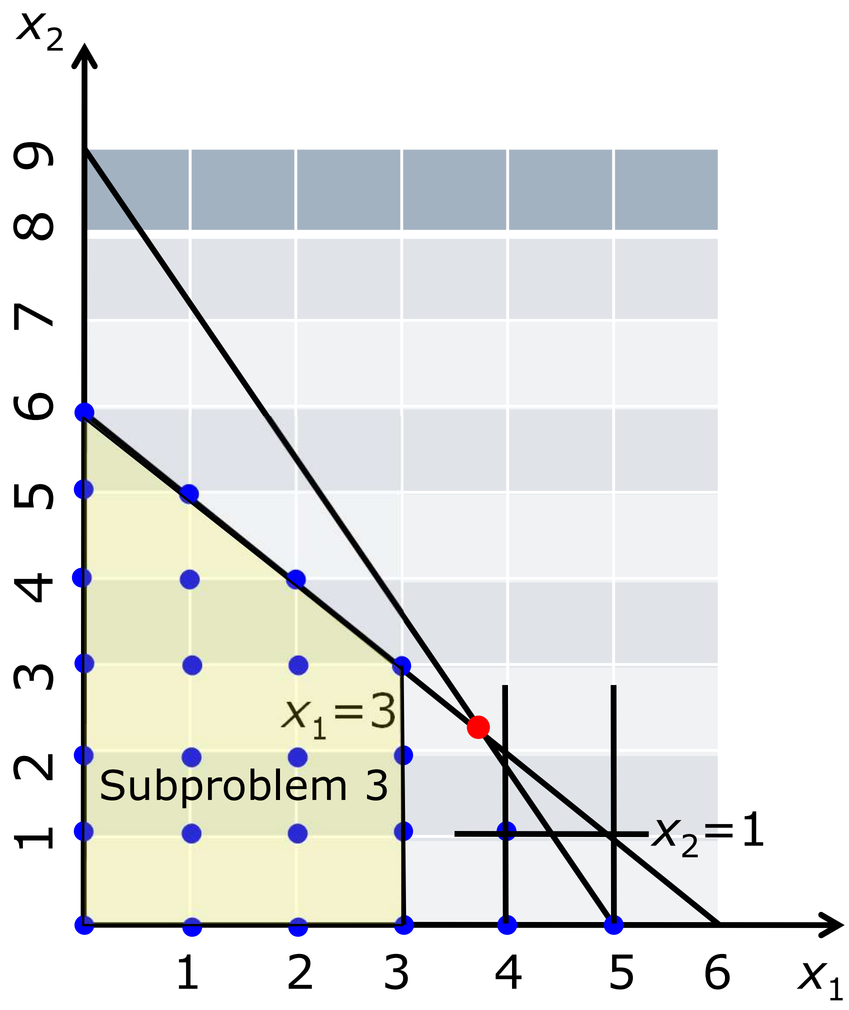
Example 1: Cont'd



Example 1: Cont'd

- **Subproblems 6 and 7**:

The optimal solution to subproblem 5 did not yield an all-integer solution, so we choose to use partition subproblem 5's feasible region by branching on the fractional-valued variable x_1. This yields two new subproblems.



Subproblem 6: Subproblem 5 + constraint $x_1 \geq 5$
Subproblem 7: Subproblem 5 + constraint $x_1 \leq 4$

Example 1: Cont'd

Subproblem 6:

$$
\begin{aligned}
\max\ & z = 8x_1 + 5x_2 \\
\text{s.t.}\ & x_1 + x_2 \le 6 \\
& 9x_1 + 5x_2 \le 45 \\
& x_1 \ge 4 \\
& x_2 \le 1 \\
& x_1 \ge 5 \\
& x_1, x_2 \ge 0
\end{aligned}
$$

Subproblem7:

$$
\begin{aligned}
\max\ & z = 8x_1 + 5x_2 \\
\text{s.t.}\ & x_1 + x_2 \le 6 \\
& 9x_1 + 5x_2 \le 45 \\
& x_1 \ge 4 \\
& x_2 \le 1 \\
& x_1 \le 4 \\
& x_1, x_2 \ge 0
\end{aligned}
$$

The set of unsolved subproblems consists of subproblems 3,6, and 7.

LIFO rule implies that we next solve subproblems 6 and 7.

We arbitrarily choose to solve subproblem 7, which has optimal solution $z=37$, $x_1=4$, $x_2=1$. Both x_1 and x_2 assume integer values, so this solution is feasible for the original IP. Subproblem 7 has been fathomed.

Example 1: Cont'd

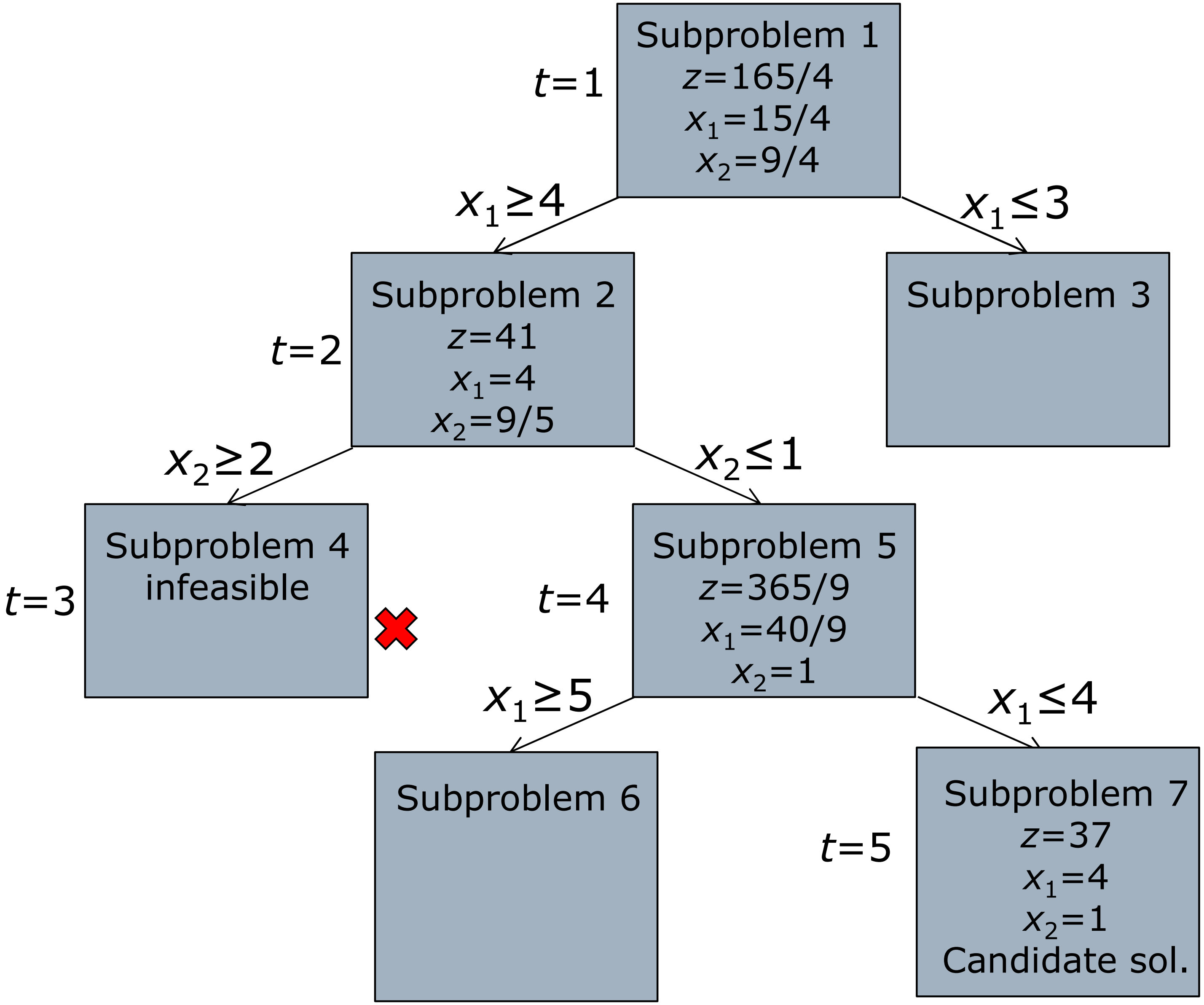
Subproblem7:

$$\max z=8x_1 + 5x_2$$
$$\text{s.t.}\quad \begin{aligned} x_1 + x_2 &\le 6 \\ 9x_1 + 5x_2 &\le 45 \\ x_1 &\ge 4 \\ x_2 &\le 1 \\ x_1 &\le 4 \\ x_1, x_2 &\ge 0 \end{aligned}$$

A solution obtained by solving a subproblem in which all variables have integer values is a candidate solution. Because the candidate solution may be optimal, we must keep a candidate solution until a better feasible solution to the IP (if any exists) is found.

Subproblem 7 has a feasible solution to the original IP with z=37, so we may conclude that the optimal z-value for the IP≥37. The z-value for the candidate solution is a lower bound on the optimal z-value for the IP.

Example 1: Cont'd



Example 1: Cont'd

Subproblem 6:

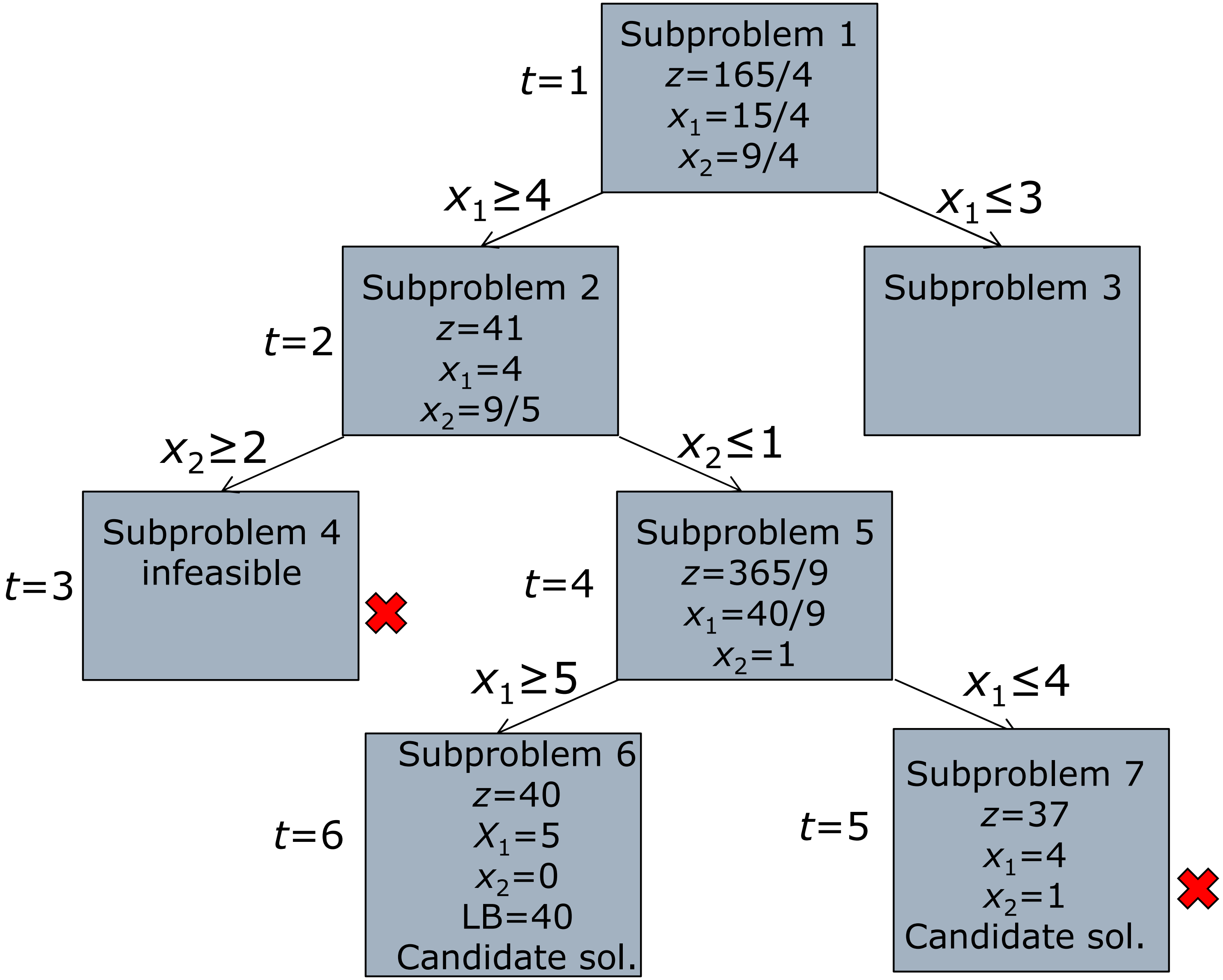
$$
\begin{aligned}
\max\ z = 8x_1 + 5x_2 & \\
\text{s.t.} \quad x_1 + x_2 &\le 6 \\
9x_1 + 5x_2 &\le 45 \\
x_1 &\ge 4 \\
x_2 &\le 1 \\
x_1 &\ge 5 \\
x_1, x_2 &\ge 0
\end{aligned}
$$

The only remaining unsolved subproblems are 6 and 3. Following the LIFO rule, we next solve subproblem 6.

The optimal solution to subproblem 6 is z=40, x_1=5, x_2=0. This is a candidate solution. Its z-value of 40 is larger than the z-value of the best previous candidate (candidate 7 with z=37). Thus subproblem 7 cannot yield the optimal solution to the original IP.

We update the LB = 40.

Example 1: Cont'd



Example 1: Cont'd

Subproblem3:

$$\max z = 8x_1 + 5x_2$$

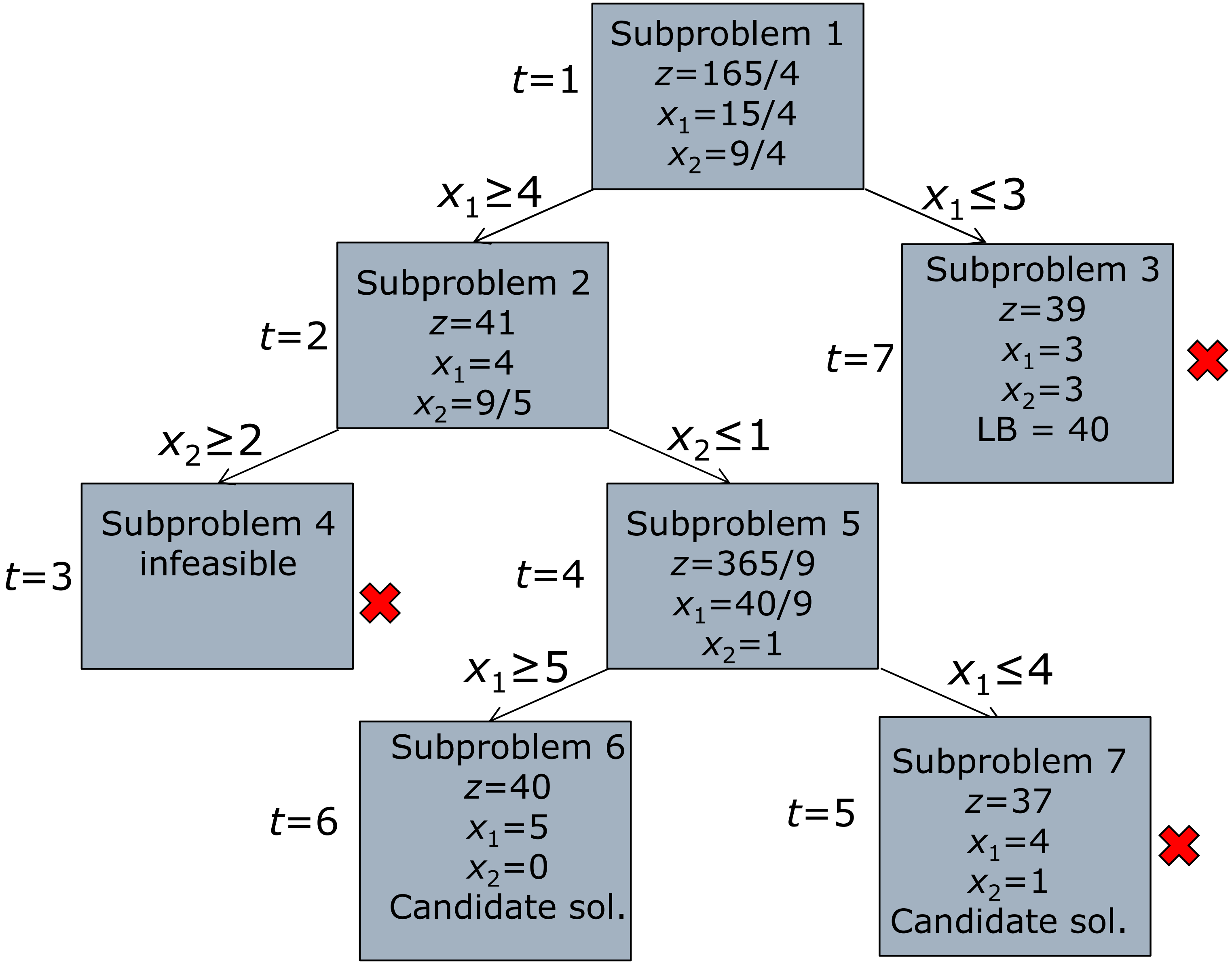
$$\text{s.t.} \quad \begin{aligned} x_1 + x_2 &\le 6 \\ 9x_1 + 5x_2 &\le 45 \\ x_1 &\le 3 \\ x_1, x_2 &\ge 0 \end{aligned}$$

The only unsolved subproblem is 3. The optimal solution for subproblem 3 is $z=39$, $x_1=x_2=3$.

Subproblem 3 cannot yield a z-value exceeding the current LB of 40, so it cannot yield the optimal solution to the original IP.

There is no remaining unsolved subproblems, and subproblem 6 yields the optimal solution to the original IP.

Example 1: Cont'd



Summary B&B: branching

- Suppose that in a given subproblem (call it old subproblem), x_j assumes a fractional value between the integers k and k+1. Then the two newly generated subproblems are

New Subproblem 1 Old subproblem + Constraint $x_j \leq k$

New Subproblem 2 Old subproblem + Constraint $x_j \geq k+1$

Summary (cont'd): fathoming (pruning) rules

- If it is unnecessary to branch on a subproblem, we say that it is **fathomed.** These three situations (for a max problem) result in a subproblem being fathomed
  - The subproblem is infeasible; thus it cannot yield the optimal solution to the IP.
  - The subproblem yields an optimal solution in which all variables have **integer values**. If this optimal solution has a better *z*-value than any previously obtained solution that is feasible in the IP, then it becomes a **candidate solution**, and its *z*-value becomes the current lower bound (LB) on the optimal *z*-value for the IP.
  - The optimal *z*-value for the subproblem does not exceed (in a max problem) the current LB, so it may be eliminated from consideration.

Summary (cont'd): node selection rules

- Two general approaches are used to determine which subproblem should be solved next.
  - The most widely used is LIFO (late in first out).
    - LIFO leads us down one side of the branch-and-bound tree and quickly find a candidate solution and then we backtrack our way up to the top of the other side
    - The LIFO approach is often called **backtracking**.
  - The second commonly used approach is **jumptracking**.
    - When branching on a node, the jumptracking method solves all the problems created by branching.

Tolerance Setting for Excel Solver

- When solving IP problems using Excel Solver you can adjust a Solver tolerance setting.
- The setting is found under the Options.
- For example a tolerance value of .20 causes the Solver to stop when a feasible solution is found that has an objective function value within 20% of the optimal *z*-value for the problem's LP relaxation.

4. B&B Method for Solving Mixed IP Problems

- In mixed IP, some variables are required to be integers and others are allowed to be either integer or nonintegers.
- To solve a mixed IP by the B&B method, modify the method by branching only on variables that are required to be integers.
- For a solution to a subproblem to be a candidate solution, it need only assign integer values to those variables that are required to be integers

Example 2

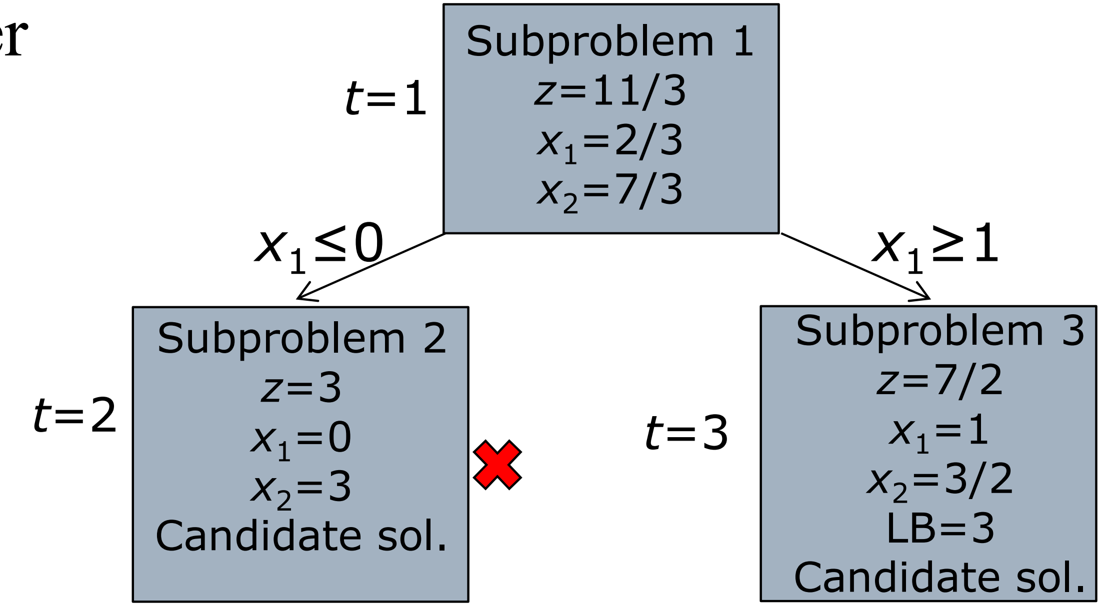
- Using B&B method to solve the following mixed IP:

$$\max z = 2x_1 + x_2$$

$$\text{s.t.} \quad 5x_1 + 2x_2 \leq 8$$

$$x_1 + x_2 \leq 3$$

$$x_1, x_2 \geq 0;\ x_1 \text{ integer}$$



5. Solving Combinatorial Optimization Problems by B&B Method

- Loosely speaking, a **combinatorial optimization problem** is any optimization problem that has a finite number of feasible solutions.
- A B&B approach is often the most efficient way to solve them.

Machine-Scheduling Problem

- Four jobs must be processed on a single machine. The time required to process each job and the date the job is due are shown in Table 63. The delay of a job is the number of days after the due date that a job is completed (if a job is completed on time or early, the job's delay is zero). In what order should the jobs be processed to minimize the total delay of the four jobs?

Durations and Due Date of Jobs

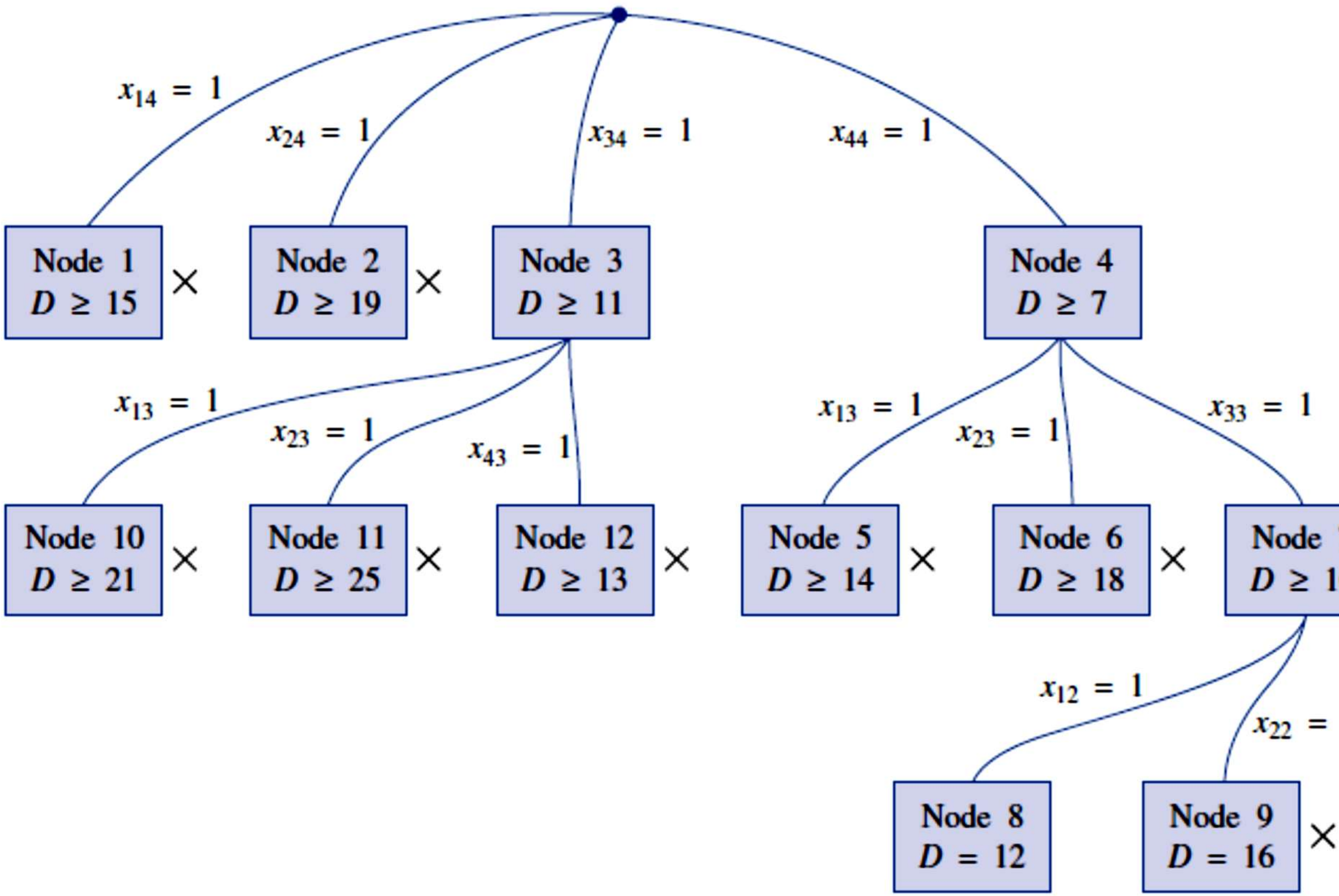
Job	Days Required to Complete Job	Due Date
1	6	End of day 8
2	4	End of day 4
3	5	End of day 12
4	8	End of day 16

Delays Incurred If Jobs Are Processed in the Order 1-2-3-4

Job	Completion Time of Job	Delay of Job
1	6	0
2	$6 + 4 = 10$	$10 - 4 = 6$
3	$6 + 4 + 5 = 15$	$15 - 12 = 3$
4	$6 + 4 + 5 + 8 = 23$	$23 - 16 = 7$

- $x_{ij} = 1$ if job i is the jth job to be processed; 0 otherwise.

Branch-and-Bound Tree for Machine-Scheduling Problem



Example 3: Traveling Salesperson Problem

- A salesperson must visit each of the 10 cities before returning to his home. What ordering of the cities minimizes the total distance the salesperson must travel before returning home? *This problem is called the traveling sales person problem (TSP).*
- Joe State lives in Gary, Indiana and owns insurance agencies in Gary, Fort Wayne, Evansville, Terre Haute and South Bend. Each December he visits each of his insurance agencies. The distance between each agency is known. What order of visiting his agencies will minimize the total distance traveled?

	Gary	**Fort Wayne**	**Evansville**	**Terre Haute**	**South Bend**
City 1 Gary	0	132	217	164	58
City 2 Fort Wayne	132	0	290	201	79
City 3 Evansville	217	290	0	113	303
City 4 Terre Haute	164	201	113	0	196
City 5 South Bend	58	79	303	196	0

Example 3: Solution

- Several B&B approaches have been developed for solving TSPs.
- Optimal solution: Gary->South Bend->Fort Wayne->Terre Haute->Evansville->Gary. Total travel distance: 668 miles

Heuristic Methods for Example 3

- When using B&B methods to solve TSPs with many cities, large amounts of computer time is needed.
- **Heuristic methods**, or **heuristics**, can be used to quickly lead to a good solution.
- Heuristics is a method used to solve a problem by trial and error when an exact algorithm approach is impractical.
- Two types of heuristic methods can be used to solve TSP: *nearest neighbor method* and *cheapest-insertion method*.

Nearest Neighbor Heuristic (NNH)

- Nearest Neighbor Method
  - Begin at any city and then "visit" the nearest city.
  - Then go to the unvisited city closest to the city we have most recently visited.
  - Continue in this fashion until a tour is obtained. After applying this procedure beginning at each city, take the best tour found.

Example, if we arbitrarily choose to start from City 1, the NNH yields the following tour: 1-5-2-4-3-1 with a total travel distance of 704 miles.

6. Cutting Plane Algorithm

- An alternative method to the branch-and-bound method is the **cutting plane algorithm**.
- Summary of the cutting plane algorithm

Step 1 Find the optional solution for the IP's programming relaxation. If all variables in the optimal solution assume integer values, we have found an optimal solution to the IP; otherwise, proceed to step2.

Step 2 Pick a constraint in the LP relaxation optimal solution whose right-hand side has the fractional part closest to 1/2. This constraint will be used to generate a cut.

Step 2a For the constraint identified in step 2, write its right-hand side and each variable's coefficient in the form $\lfloor x \rfloor + f$, where $0 <= f < 1$.

Step 2b Rewrite the constraint used to generate the cut as

All terms with integer coefficients = all terms with fractional coefficients

Then the cut is All terms with fractional coefficients <= 0

Step 3 Use the simplex to find the optimal solution to the LP relaxation, with the cut as an additional constraint.

- If all variables assume integer values in the optimal solution, we have found an optimal solution to the IP.
- Otherwise, pick the constraint with the most fractional right-hand side and use it to generate another cut, which is added to the model.
- We continue this process until we obtain a solution in which all variables are integers. This will be an optimal solution to the IP.

Example

$$\begin{aligned} \max z &= 8x_1 + 5x_2 \\ \text{s.t.} \quad x_1 + x_2 &\le 6 \\ 9x_1 + 5x_2 &\le 45 \\ x_1, x_2 \ge 0;\ x_1, x_2 &\text{ integer} \end{aligned} \tag{51}$$

- Optimal solution of LP relaxation

$$z = 41.25 - 1.25s_1 - 0.75s_2$$
$$x_2 = 2.25 - 2.25s_1 + 0.25s_2$$
$$x_1 = 3.75 + 1.25s_1 - 0.25s_2$$

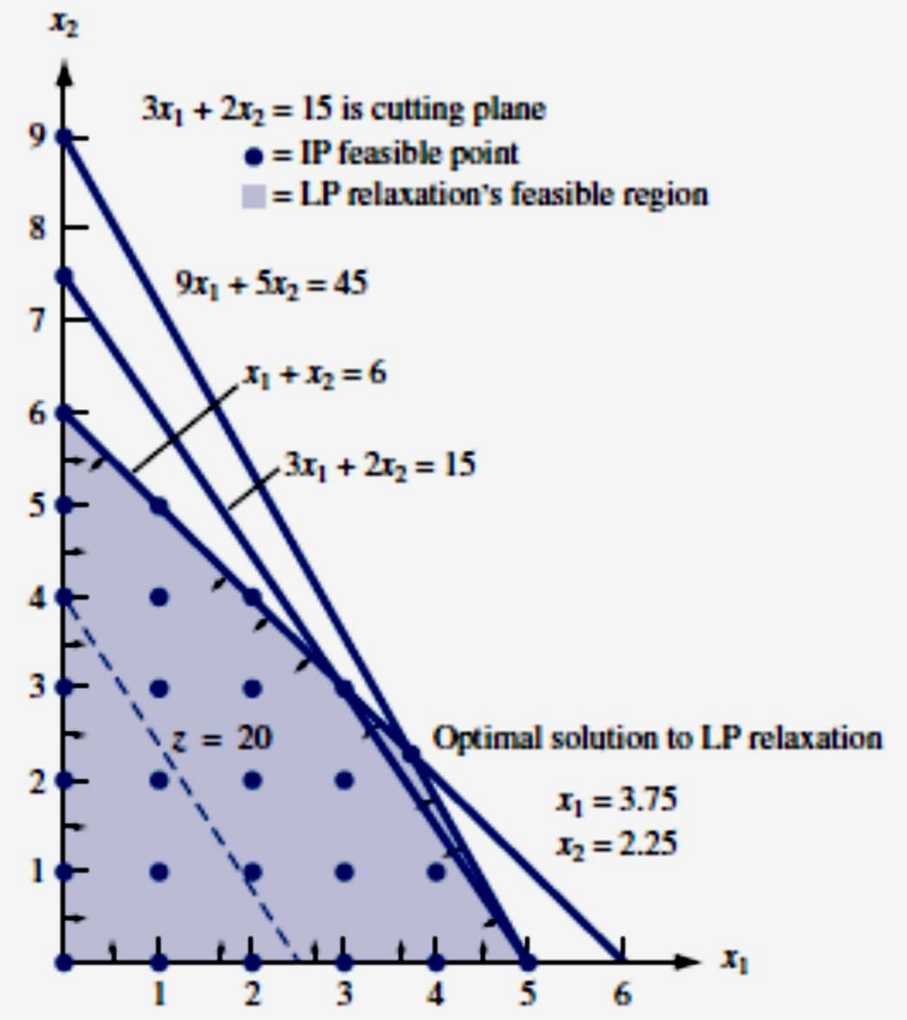


FIGURE 33
Example of Cutting Plane

Cut generation

- A cut generated using a constraint in which a BV is fractional:

$$(52)\quad x_1 - 1.25s_1 + 0.25s_2 = 3.75$$

$$(53)\quad x_1 - 2s_1 + 0.75s_1 + 0s_2 + 0.25s_2 = 3 + 0.75$$

$$(54)\quad x_1 - 2s_1 + 0s_2 - 3 = 0.75 - 0.75s_1 - 0.25s_2$$

$$(55)\quad 0.75 - 0.75s_1 - 0.25s_2 <= 0$$

- A constraint is called a cut that has two properties:
  1. Any feasible point for the IP will satisfy the cut.
  2. The current optimal solution to the LP relaxation will not satisfy the cut.
- The algorithm requires that all coefficients of variables in the constraints and all right-hand sides of constraints be integers. This is to ensure that if the original decision variables are integers, then the slack and excess variables will also be integers. Thus, a constraint such as $x_1 + 0.5x_2 <= 3.6$ must be replaced by $10x_1 + 5x_2 <= 36$.

To show Point 1

- Consider any point that is feasible for the IP. For such a point, $x1$ and $x2$ take on integer values, and the point must be feasible in the LP relaxation of (51). Because (54) is just a rearrangement of the optimal tableau's second constraint, any feasible point for the IP must satisfy (54). Any feasible solution to the IP must have $s1 >= 0$ and $s2 >= 0$. Because $0.75 < 1$, any feasible solution to the IP will make the right-hand side of (54) less than 1. Also note that for any point that is feasible for the IP, the left-hand side of (54) will be an integer. Thus, for any feasible point to the IP, the right-hand side must be an integer that is less than 1. This implies that any point that is feasible for the IP satisfies (55), so our cut does not eliminate any feasible integer points from consideration!

To show Point 2

- The current optimal solution to the LP relaxation has $s_1 = s_2 = 0$. Thus, it cannot satisfy (55). This argument works because 0.75 (the fractional part of the right-hand side of the second constraint) is greater than 0.
- Thus, if we choose any constraint whose right-hand side is fractional, we can cut off the LP relaxation's optimal solution.
- The effect of the cut (55) can be seen in Figure 33; all points feasible for the IP (51) satisfy the cut (55), but the current optimal solution to the LP relaxation ($x_1 = 3.75$ and $x_2 = 2.25$) does not. To obtain the graph of the cut, we replaced s_1 by $6 - x_1 - x_2$ and s_2 by $45 - 9x_1 - 5x_2$. This enabled us to rewrite the cut as $3x_1 + 2x_2 <= 15$.

Summary of IP

- Advantages:
  - More realistic: Indivisibility
  - More Flexibility
- Disadvantages: To ensure this, we include the following constraints:
  - More difficult to model
  - Can be much more difficult to solve