Integer Programming

Cathy Wu

1.041/1.200 Transportation: Foundations and Methods

Readings

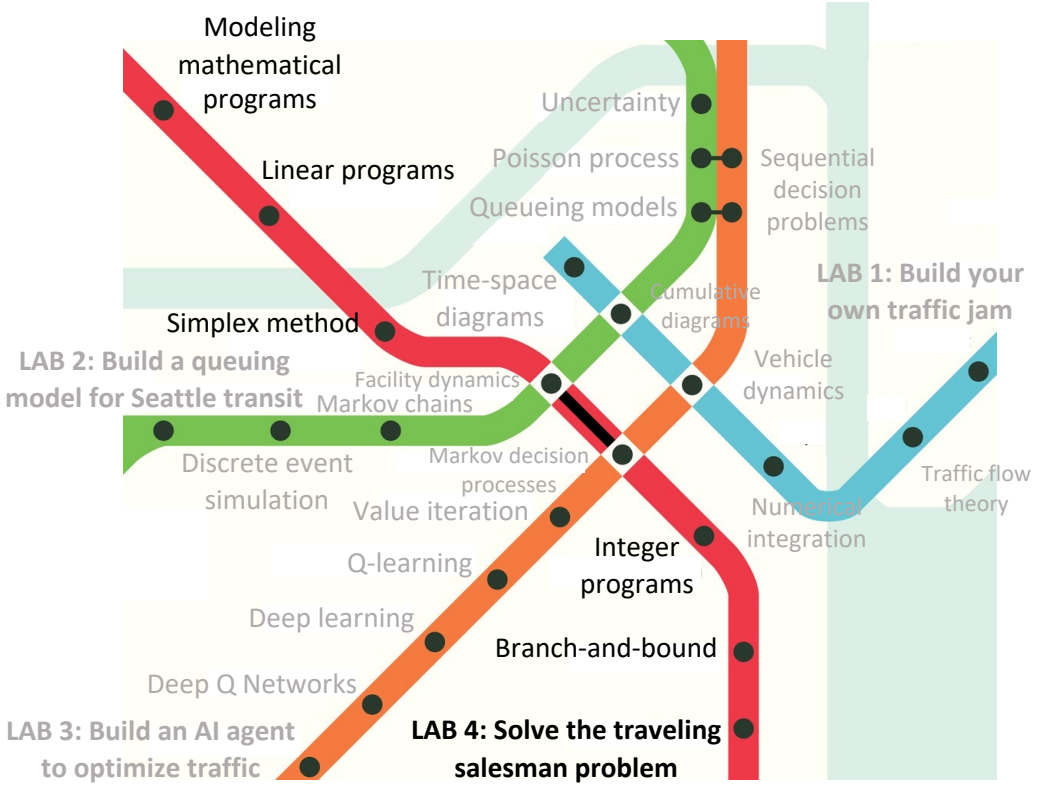
- Bradley, Stephen P., Arnoldo C. Hax, and Thomas L. Magnanti. **Applied mathematical programming**. Addison-Wesley (1977). Chapter 9 Integer Programming [URL]

Unit 4: Optimizing transportation resources

Unit 4

Optimizing

Single-stage



Outline

1. Integer programming
2. Example uses of integer variables
   - Knapsack problem
   - Scheduling
   - Traveling salesperson
3. Solving integer programs: branch and bound

What are integer programming problems?

- Linear programming problems in which fractional solutions are not realistic.
- Typically OK to assume as fractional
  - Highway traffic volumes (2000 vehicles is not so different from 2001, so neither is 2000.5)
  - Passengers on a train
- Typically not OK to assume as fractional
  - Number of trains, nuclear aircraft carriers, cruise ships, etc. (large expensive vehicles, 2 is quite different from 3)
  - Number of bus drivers
- Types of integer programs:
  - Mixed integer programs: when some, but not all, variables are restricted to be integer.
  - Pure integer programs: when all decision variables must be integers.
  - Binary programs: when all decision variables must be either 0 or 1.

Some Integer-Programming Models

Warehouse Location

- In modeling distribution systems, decisions must be made about tradeoffs between transportation costs and costs for operating distribution centers.
- As an example, suppose that a manager must decide which of m warehouses to use for meeting the demands of n customers for a good.
- The decisions to be made are which warehouses to operate and how much to ship from any warehouse to any customer.

Decision variables and relevant data

$$y_i = \begin{cases} 1 & \text{if warehouse } i \text{ is opened} \\ 0 & \text{if warehouse } i \text{ is not opened} \end{cases}$$

x_{ij} = Amount to be sent from warehouse i to customer j

f_i = Fixed operating cost for warehouse i, if opened (for example, a cost to lease the warehouse)

c_{ij} = Per-unit operating cost at warehouse i plus the transportation cost for shipping from warehouse i to customer j

- There are two types of constraints for the model:
  - the demand d_j of each customer must be filled from the warehouses,
  - goods can be shipped from a warehouse only if it is opened.

Model

$$\text{Minimize} \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij} + \sum_{i=1}^{m} f_i y_i$$

subject to:

$$\sum_{i=1}^{m} x_{ij} = d_j \quad (j = 1, 2, \ldots, n)$$

$$\sum_{j=1}^{n} x_{ij} - y_i \left(\sum_{j=1}^{n} d_j \right) \leq 0 \quad (i = 1, 2, \ldots, m)$$

$$x_{ij} \geq 0 \quad (i = 1, 2, \ldots, m;\ j = 1, 2, \ldots, n)$$

$$y_i = 0 \text{ or } 1 \quad (i = 1, 2, \ldots, m)$$

The model can be made much richer by including logical considerations

- The opening of one warehouse contingent upon opening of another warehouse (contingency constraints). Example: opening of pets warehouse is contingent on pet food & supplies warehouse.

$$y_i \leq y_j$$

- Conflicting warehouses (multiple-choice constraints)

$$y_1 + y_2 + y_3 + y_4 \leq 1$$

The 0-1 knapsack problem

- Resource constraints: warehouses may have costs to open / operate / construct (capital costs)

$$\text{Maximize} \sum_{i=1}^{m} f_i y_i$$

subject to:

$$\sum_{j=1}^{n} a_j y_j \leq b$$

$$y_j \in \{0, 1\} \quad (j = 1, 2, \ldots, m)$$

Scheduling

- Consider the scheduling of airline flight personnel.
- The airline has a number of routing ‘‘legs’’ to be flown, such as 10 A.M. New York to Chicago, or 6 P.M. Chicago to Los Angeles.
- The airline must schedule its personnel crews on routes to cover these flights. One crew, for example, might be scheduled to fly a route containing the two legs just mentioned.

Decision variables and relevant data

$$x_j = \begin{cases} 1 & \text{if a crew is assigned to route } j \\ 0 & \text{otherwise} \end{cases}$$

$$a_{ij} = \begin{cases} 1 & \text{if leg } i \text{ is included on route } j \\ 0 & \text{otherwise} \end{cases}$$

$$c_{ij} = \text{Cost for assigning a crew to route } j$$

- The coefficients a_{ij} define the acceptable combinations of legs and routes, taking into account such characteristics as sequencing of legs for making connections between flights and for including in the routes ground time for maintenance.

Model

$$\text{Minimize} \sum_{j=1}^{n} c_j x_j$$

subject to:

$$\sum_{j=1}^{n} a_{ij} x_j = 1 \quad (i = 1, 2, \ldots, m)$$

$$x_j = 0 \text{ or } 1 \quad (j = 1, 2, \ldots, n)$$

An alternative formulation that permits a crew to ride as passengers on a leg

$$\sum_{j=1}^{n} a_{ij} x_j \geq 1 \quad (i = 1, 2, \ldots, m)$$

- Set-partitioning: $\sum_{j=1}^{n} a_{ij} x_j = 1$
- Set-covering: $\sum_{j=1}^{n} a_{ij} x_j \geq 1$

Traveling salesperson problem

- Starting from home, a salesperson wishes to visit each of $(n - 1)$ other cities and return home at minimal cost.
- The salesperson must visit each city exactly once and it costs c_{ij} to travel from city i to city j.
- What route should the salesperson select?

Decision variables

$$x_{ij} = \begin{cases} 1 & \text{if the salesperson goes from city } i \text{ to city } j \\ 0 & \text{otherwise} \end{cases}$$

Model

$$\text{Minimize} \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}x_{ij}$$

subject to:

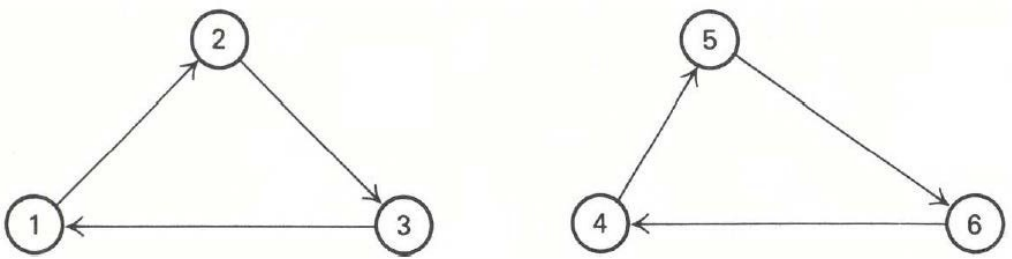
$$\sum_{i=1}^{n} x_{ij} = 1 \quad (j = 1, 2, \dots, n)$$

$$\sum_{j=1}^{n} x_{ij} = 1 \quad (i = 1, 2, \dots, m)$$

$$x_{ij} \in \{0, 1\} \quad (i = 1, 2, \dots, m;\ \ j = 1, 2, \dots, n)$$

- Sub-tours are not prevented.

Sub-tours elimination



+

$$x_{14} + x_{15} + x_{16} + x_{24} + x_{25} + x_{26} + x_{34} + x_{35} + x_{36} \geq 1$$

- With n cities, $(2^n - 1)$ constraints of this nature must be added…
  - True for this strategy and many other strategies used to eliminate sub-tours in TSP models.

Formulating Integer Programs

Binary 0-1 variables

- Suppose that we are to determine whether or not to engage in the following activities:
  1. to build a new warehouse,
  2. to undertake a public service/awareness campaign, or
  3. to develop/reform a new service.
- In each case, we must make a yes–no or so-called go–no–go decision. These choices are modeled easily by letting $x_j = 1$ if we engage in the jth activity and $x_j = 0$ otherwise.
- Variables that are restricted to 0 or 1 in this way are termed binary, bivalent, logical, or 0–1 variables.
- Binary variables are of great importance because they occur regularly in many model formulations.
- But can be also used as auxiliary variables.

Constraint Feasibility, a.k.a. Big M constraint

- Does a given choice of the decision variables satisfy the constraint below?

$$f(x_1, x_2, \ldots, x_n) \leq b$$

- Introduce a binary variable y with the interpretation:

$$y = \begin{cases} 1 & \text{if the constraint is known to be satisfied} \\ 0 & \text{otherwise} \end{cases}$$

and write:

$$f(x_1, x_2, \ldots, x_n) - By \leq b$$

B is chosen to be large enough so that the constraint always is satisfied if $y = 1$

Alternative constraints

$$f_1(x_1, x_2, \ldots, x_n) \leq b_1$$
$$f_2(x_1, x_2, \ldots, x_n) \leq b_2$$

- At least one, but not necessarily both, of these constraints must be satisfied.
  - This restriction can be modeled by combining the technique just introduced with a multiple-choice constraint as follows:

$$f_1(x_1, x_2, \ldots, x_n) - B_1 y_1 \leq b_1$$
$$f_2(x_1, x_2, \ldots, x_n) - B_2 y_2 \leq b_2$$
$$y_1 + y_2 \leq 1$$
$$y_1, y_2 \text{ are binary}$$

  - We can save one integer variable in this formulation:

$$f_1(x_1, x_2, \ldots, x_n) - B_1 y_1 \leq b_1$$
$$f_2(x_1, x_2, \ldots, x_n) - B_2 (1 - y_1) \leq b_2$$
$$y_1 = 0 \; or \; 1$$

Conditional Constraints

- These constraints have the form:
  $f_1(x_1, x_2, \ldots, x_n) > b_1$ implies that $f_2(x_1, x_2, \ldots, x_n) \leq b_2$
- If we note that:

$$(p \Longrightarrow q) \Leftrightarrow (\sim p \vee q)$$

we end up with the following alternative constraints:

$$f_1(x_1, x_2, \ldots, x_n) \leq b_1$$
$$f_2(x_1, x_2, \ldots, x_n) \leq b_2$$

k-Fold Alternatives

- We must satisfy at least k of the constraints:

$$f_j(x_1, x_2, \dots, x_n) \le b_j \quad (j = 1, 2, \dots, p)$$

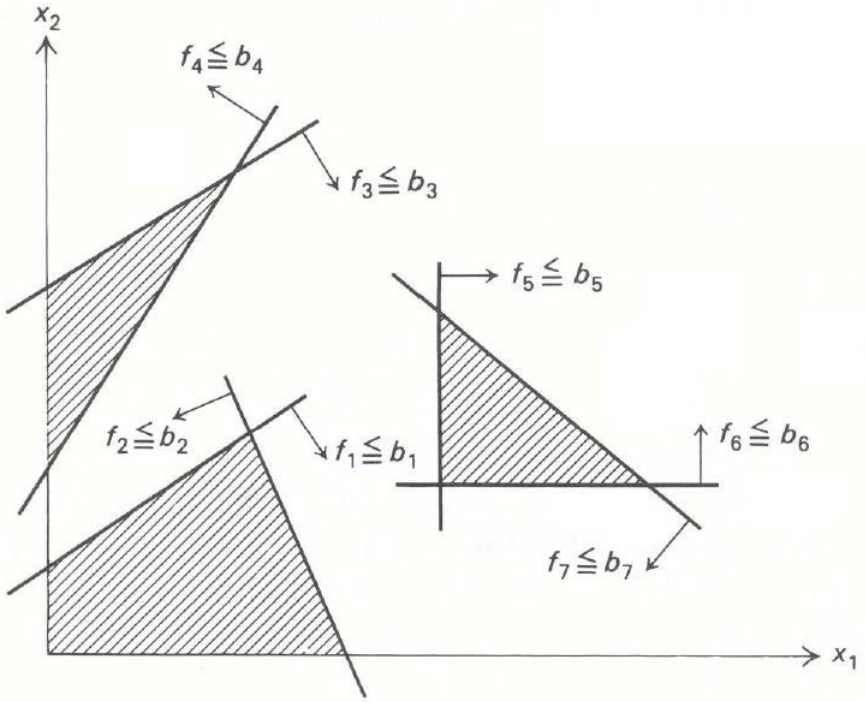
Assuming that B_j for $j = 1, 2, \dots, p$ are chosen so that the ignored constraints will not be binding, the general problem can be formulated as follows:

$$f_j(x_1, x_2, \dots, x_n) - B_j(1 - y_j) \le b_j \quad (j = 1, 2, \dots, p)$$

$$\sum_{j=1}^{p} y_j \ge k$$

$$y_j = 0 \text{ or } 1 \quad (j = 1, 2, \dots, p)$$

Compound Alternatives



$$\left.\begin{aligned} f_1(x_1, x_2) - B_1 y_1 &\leq b_1 \\ f_2(x_1, x_2) - B_2 y_1 &\leq b_2 \end{aligned}\right\} \quad \text{Region 1 constraints}$$

$$\left.\begin{aligned} f_3(x_1, x_2) - B_3 y_2 &\leq b_3 \\ f_4(x_1, x_2) - B_4 y_2 &\leq b_4 \end{aligned}\right\} \quad \text{Region 2 constraints}$$

$$\left.\begin{aligned} f_5(x_1, x_2) - B_5 y_3 &\leq b_5 \\ f_6(x_1, x_2) - B_6 y_3 &\leq b_6 \\ f_7(x_1, x_2) - B_7 y_3 &\leq b_7 \end{aligned}\right\} \quad \text{Region 3 constraints}$$

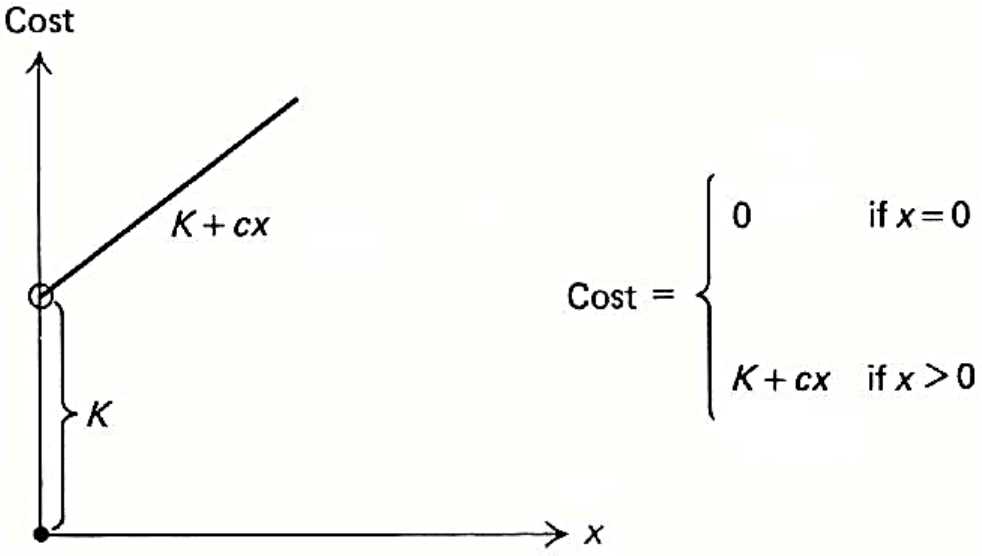
$$y_1 + y_2 + y_3 \leq 2,$$
$$x_1 \geq 0, \quad x_2 \geq 0,$$
$$y_1, y_2, y_3 \quad \text{binary}.$$

Representing Nonlinear Functions

- Nonlinear functions can be represented by integer-programming formulations…

Fixed Costs

- Frequently, the objective function for a minimization problem contains fixed costs (preliminary design costs, fixed investment costs, fixed contracts, and so forth).



Fixed Costs

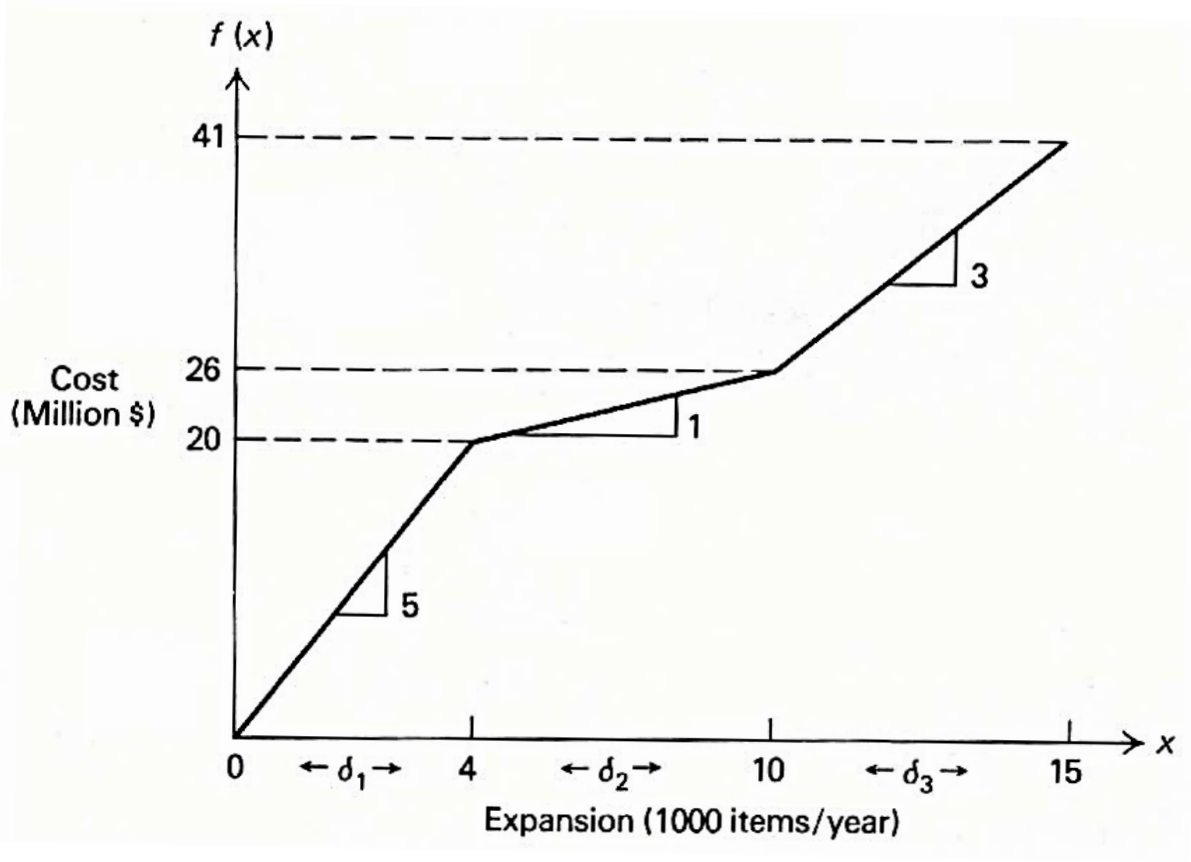
- Assume that the activity x has a limit of B units.
- Define y to be a binary variable that indicates when the fixed cost is incurred, so that $y = 1$ when $x > 0$ and $y = 0$ when $x = 0$.
- Then the contribution to cost due to x may be written as

$$Ky + cx$$

with the constraints

$$x \leq By$$
$$x \geq 0$$
$$y = 0 \; or \; 1$$

Piecewise Linear Representation



Piecewise Linear Representation

- To model the cost curve, we express any value of x as the sum of three variables $\delta_1, \delta_2, \delta_3$, so that the cost for each of these variables is linear.
- Hence,

$$x = \delta_1 + \delta_2 + \delta_3$$

where

$$0 \leq \delta_1 \leq 4$$
$$0 \leq \delta_2 \leq 6$$
$$0 \leq \delta_3 \leq 5$$

and the total variable cost is given by:

$$\text{Cost} = 5\delta_1 + \delta_2 + 3\delta_3$$

Piecewise Linear Representation

- Note that we have defined the variables so that:
  - δ_1 is the amount by which x exceeds 0, but is less than or equal to 4;
  - δ_2 is the amount by which x exceeds 4, but is less than or equal to 10
  - δ_3 is the amount by which x exceeds 10, but is less than or equal to 15.
- If this interpretation is to be valid, we must also require that $\delta_1 = 4$ whenever $\delta_2 > 0$ and that $\delta_2 = 6$ whenever $\delta_3 > 0$.
- However, these restrictions on the variables are simply conditional constraints and can be modeled by introducing (more) binary variables:

$$w_1 = \begin{cases} 1 & \text{if } \delta_1 \text{ is at its upper bound} \\ 0 & \text{otherwise} \end{cases}$$

$$w_2 = \begin{cases} 1 & \text{if } \delta_2 \text{ is at its upper bound} \\ 0 & \text{otherwise} \end{cases}$$

$$4w_1 \leq \delta_1 \leq 4$$

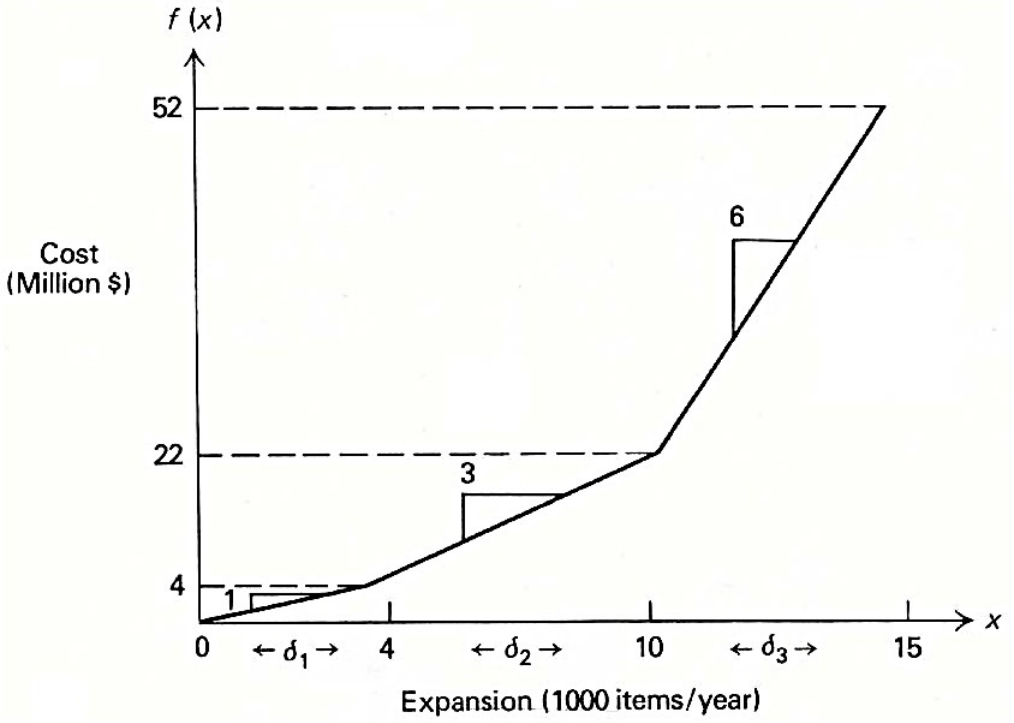
$$6w_2 \leq \delta_2 \leq 6w_1$$

$$0 \leq \delta_3 \leq 5w_2$$

$$w_1 \text{ and } w_2 \text{ are binary}$$

Diseconomies of Scale

- When marginal costs are increasing for a minimization problem or marginal returns are decreasing for a maximization problem.



Diseconomies of Scale

$$\text{Cost} = \delta_1 + 3\delta_2 + 6\delta_3$$

subject to:

$$0 \leq \delta_1 \leq 4$$
$$0 \leq \delta_2 \leq 6$$
$$0 \leq \delta_3 \leq 5$$

- The conditional constraints involving binary variables in the previous formulation can be ignored if the cost curve appears in a minimization objective function, since the coefficients of $\delta_1, \delta_2, \delta_3$ imply that it is always best to set $\delta_1 = 4$ before taking $\delta_2 > 0$, and to set $\delta_2 = 6$ before taking $\delta_3 > 0$.
- This representation without integer variables is not valid, however, if economies of scale are present.

Some Characteristics of Integer Programs

A sample problem

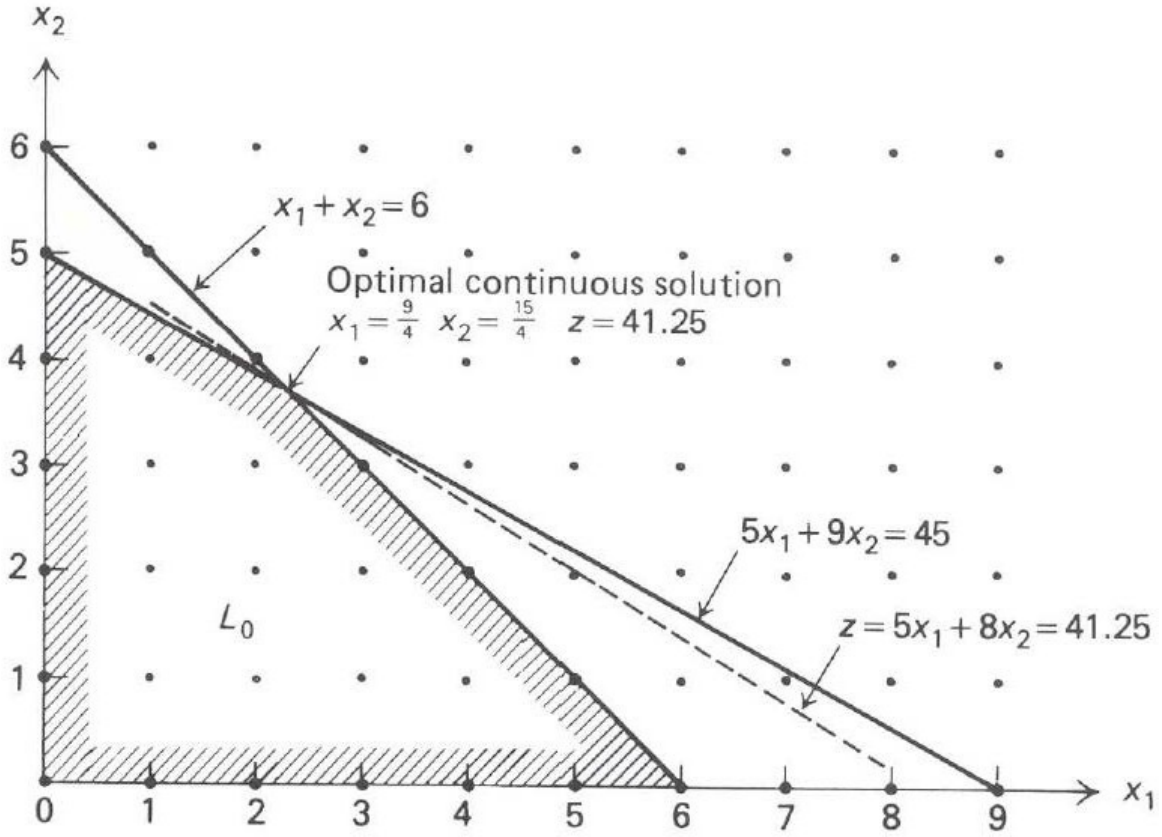
Determine:

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

subject to:

$$x_1 + x_2 \leq 6$$
$$5x_1 + 9x_2 \leq 45$$
$$x_1, x_2 \geq 0 \text{ and are integers}$$

Feasible region



Dots in the shaded region are feasible integer points.

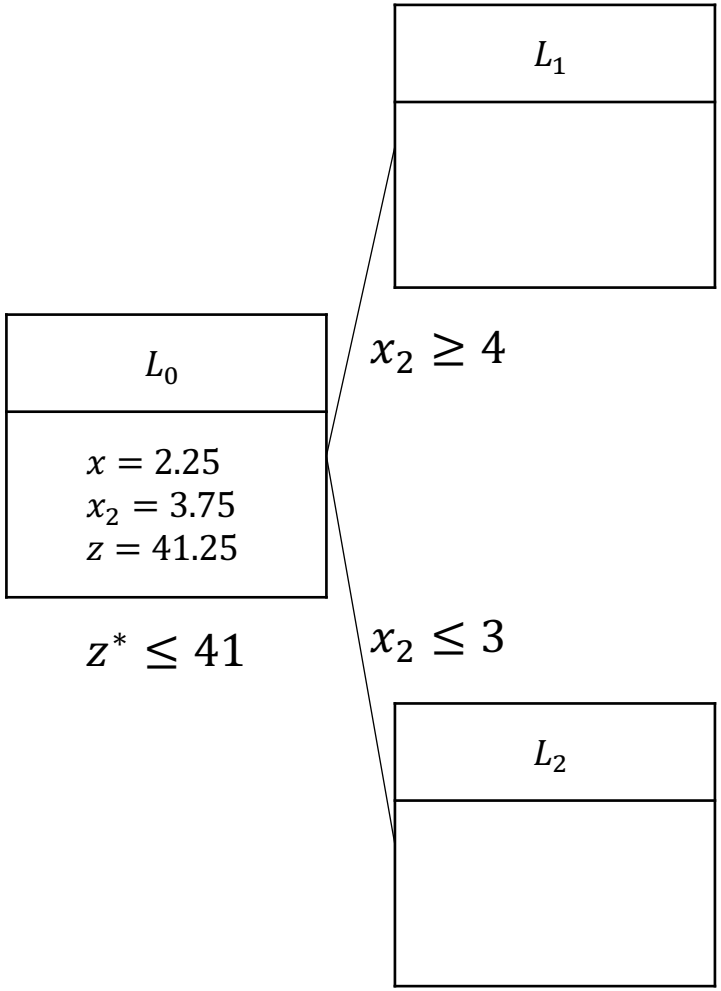
Some notes

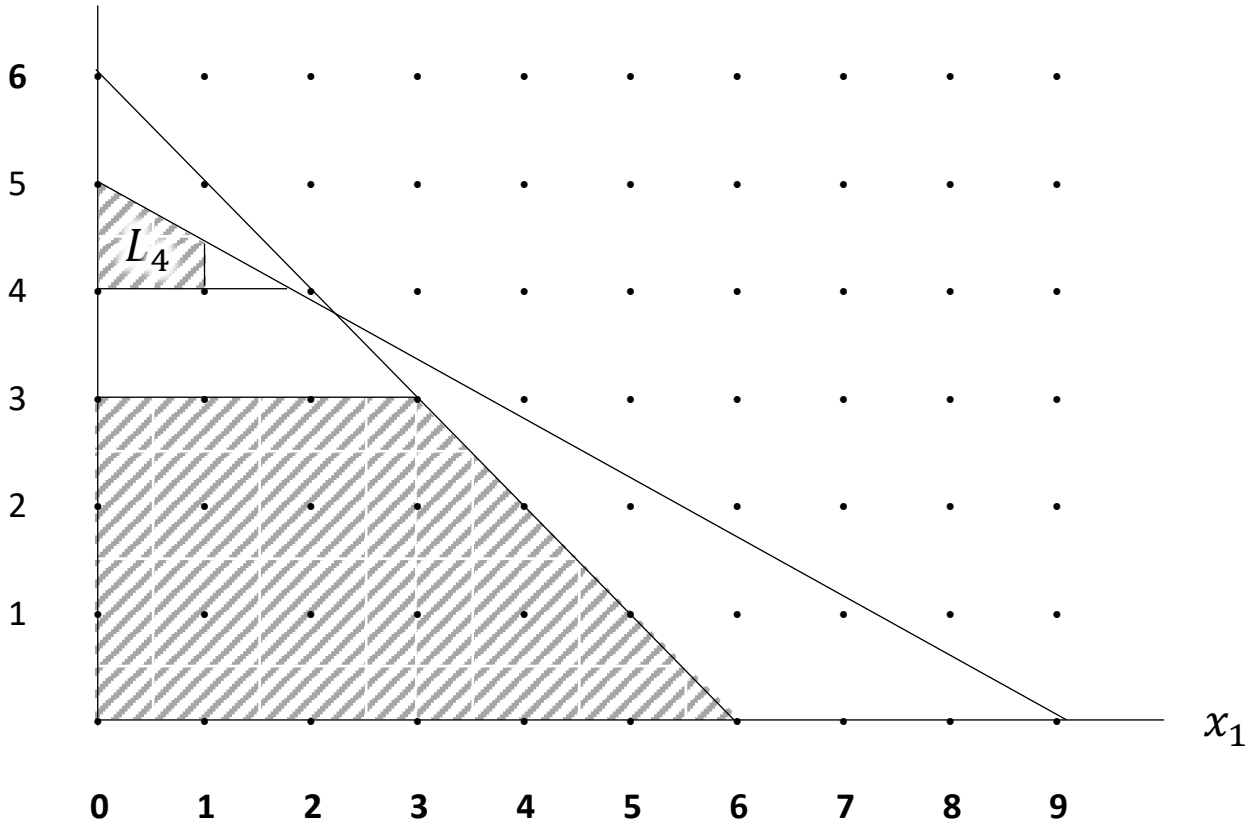
- If the integrality restrictions on variables are dropped, the resulting problem is a linear program. We will call it the associated linear program.
- The optimal integer-programming solution is not obtained by rounding the linear-programming solution.
- The closest point to the optimal linear-program solution is not even feasible.
- The nearest feasible integer point to the linear-program solution is far removed from the optimal integer point.
- It is not sufficient simply to round linear-programming solutions.
- Systematic and explicit enumeration does not work even for small problems.

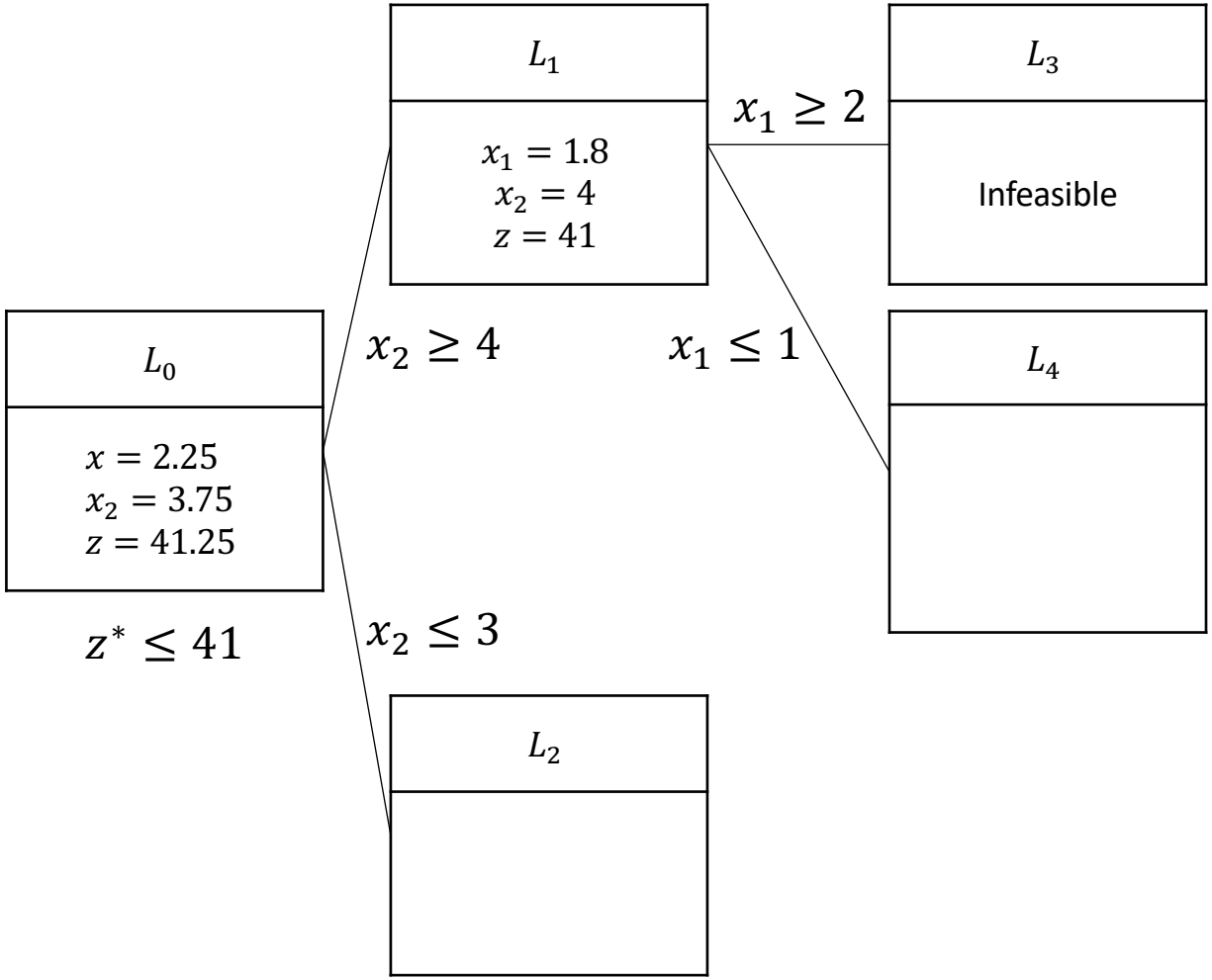
Branch-and-Bound

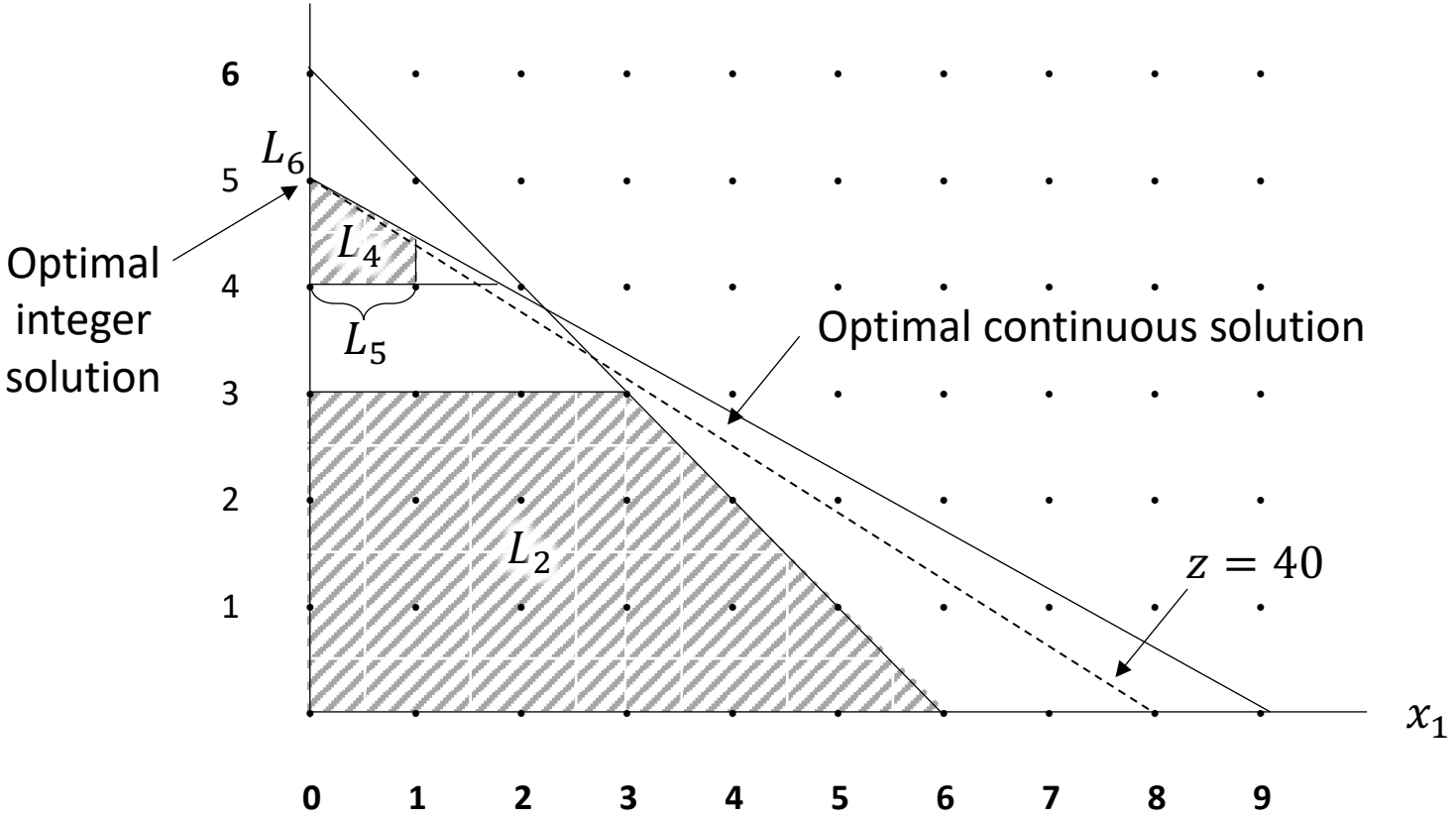
Strategy of ‘‘divide and conquer’’

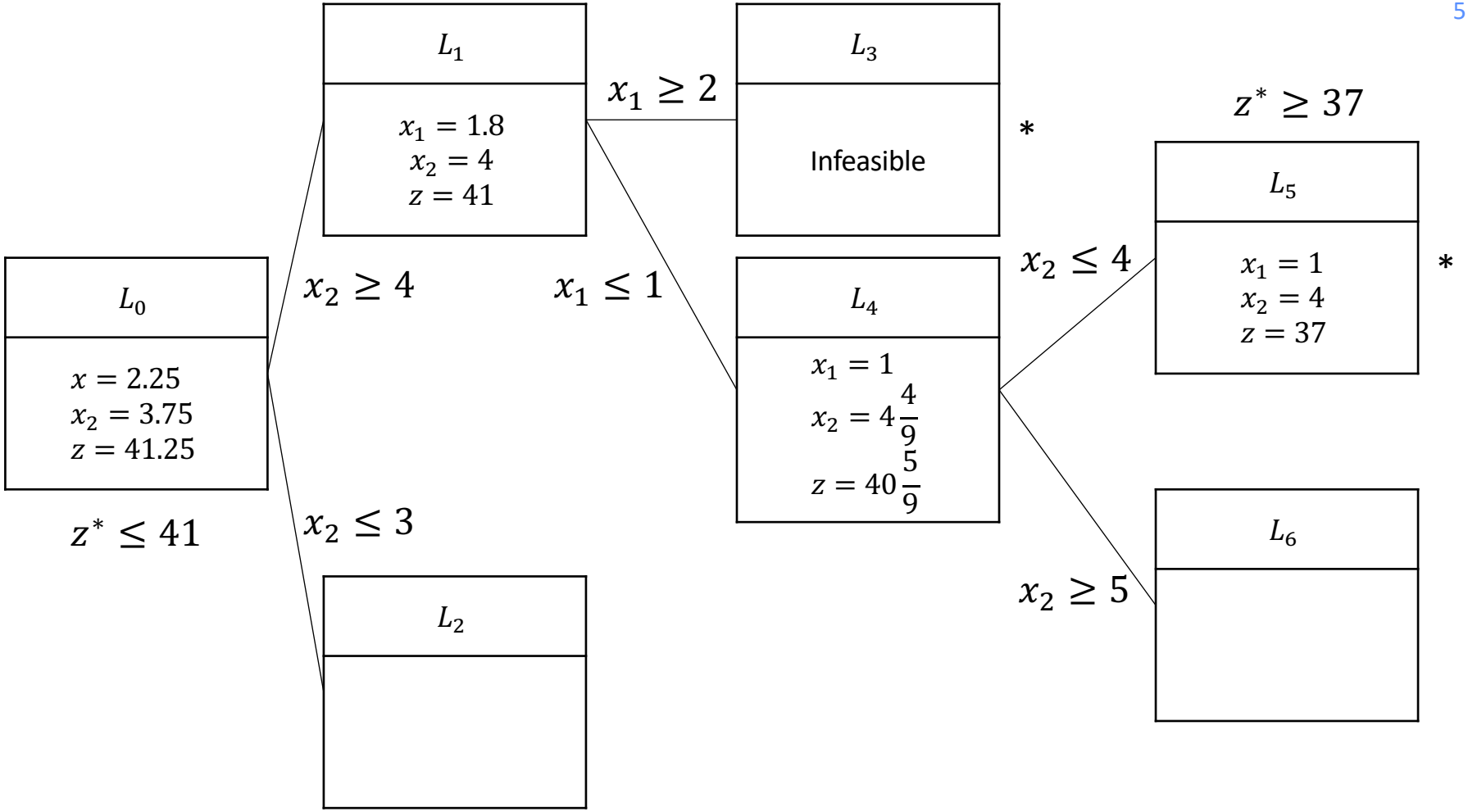
- An integer linear program is a linear program further constrained by the integrality restrictions.
- Thus, in a **maximization problem**, the value of the objective function, at the linear-program optimum, will always be an upper bound on the optimal integer-programming objective.
- In addition, any integer feasible point is always a lower bound on the optimal linear-program objective value.
- The idea of branch-and-bound is to utilize these observations to systematically subdivide the linear programming feasible region and make assessments of the integer-programming problem based upon these subdivisions.

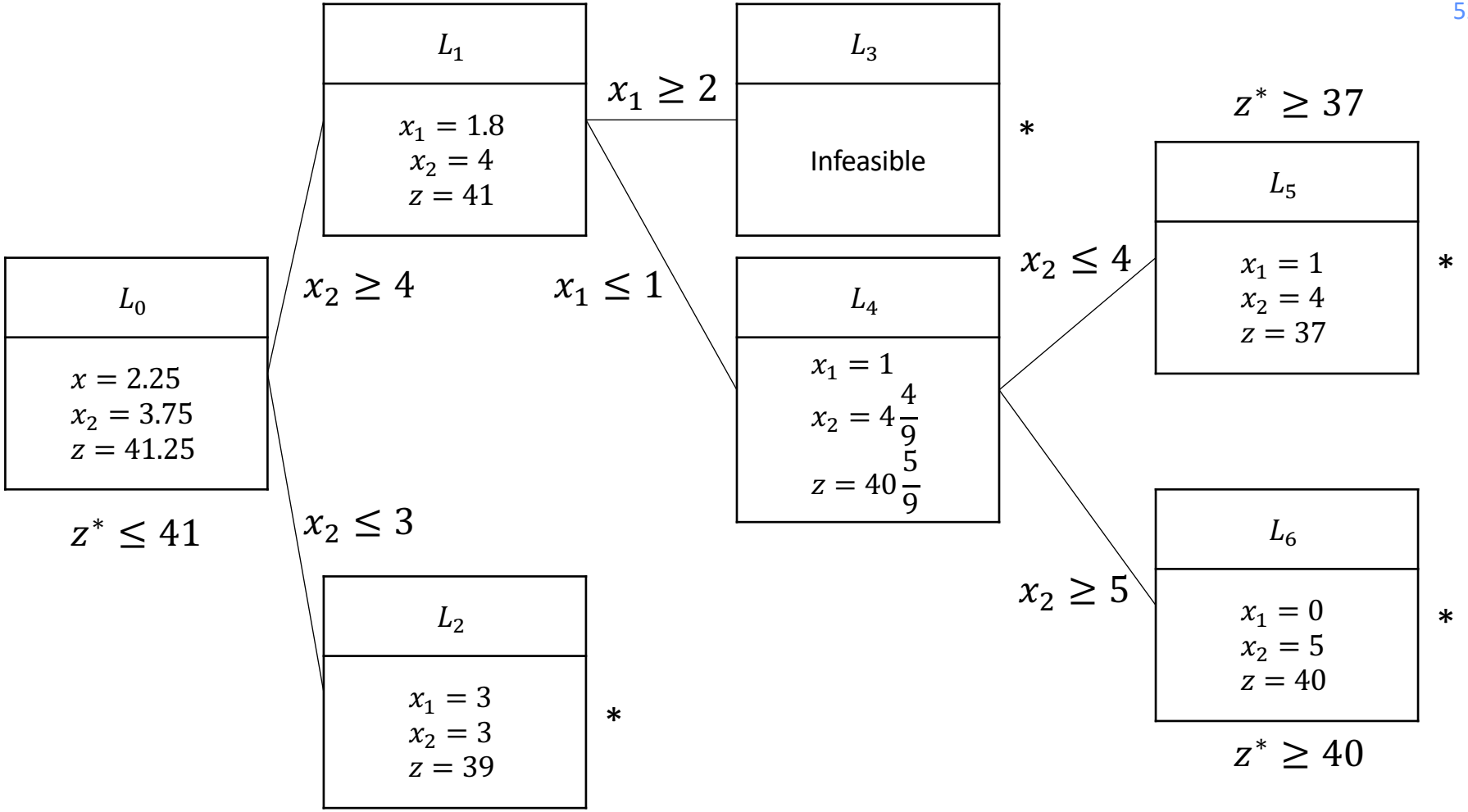

L_1
L_0
$x = 2.25$
$x_2 = 3.75$
$z = 41.25$
$x_2 \geq 4$
$x_2 \leq 3$
$z^* \leq 41$
L_2


6
5
4
3
2
1
L_4
x_1
0
1
2
3
4
5
6
7
8
9


L_0
$x = 2.25$
$x_2 = 3.75$
$z = 41.25$
$z^* \leq 41$
$x_2 \geq 4$
$x_2 \leq 3$
L_1
$x_1 = 1.8$
$x_2 = 4$
$z = 41$
$x_1 \geq 2$
$x_1 \leq 1$
L_3
Infeasible
L_4
L_2


6
5
4
3
2
1
0
1
2
3
4
5
6
7
8
9
x_1
L_6
L_4
L_5
L_2
Optimal integer solution
Optimal continuous solution
$z = 40$


L_0
$x = 2.25$
$x_2 = 3.75$
$z = 41.25$
$z^* \leq 41$
L_1
$x_1 = 1.8$
$x_2 = 4$
$z = 41$
$x_2 \geq 4$
L_2
$x_2 \leq 3$
L_3
Infeasible
$x_1 \geq 2$
*
L_4
$x_1 = 1$
$x_2 = 4\frac{4}{9}$
$z = 40\frac{5}{9}$
$x_1 \leq 1$
L_5
$x_1 = 1$
$x_2 = 4$
$z = 37$
$z^* \geq 37$
$x_2 \leq 4$
*
L_6
$x_2 \geq 5$


L_0
$x = 2.25$
$x_2 = 3.75$
$z = 41.25$
$z^* \leq 41$
$x_2 \geq 4$
$x_2 \leq 3$
L_1
$x_1 = 1.8$
$x_2 = 4$
$z = 41$
L_2
$x_1 = 3$
$x_2 = 3$
$z = 39$
*
$x_1 \geq 2$
$x_1 \leq 1$
L_3
Infeasible
*
L_4
$x_1 = 1$
$x_2 = 4\frac{4}{9}$
$z = 40\frac{5}{9}$
$x_2 \leq 4$
$x_2 \geq 5$
$z^* \geq 37$
L_5
$x_1 = 1$
$x_2 = 4$
$z = 37$
*
L_6
$x_1 = 0$
$x_2 = 5$
$z = 40$
*
$z^* \geq 40$

Further Considerations

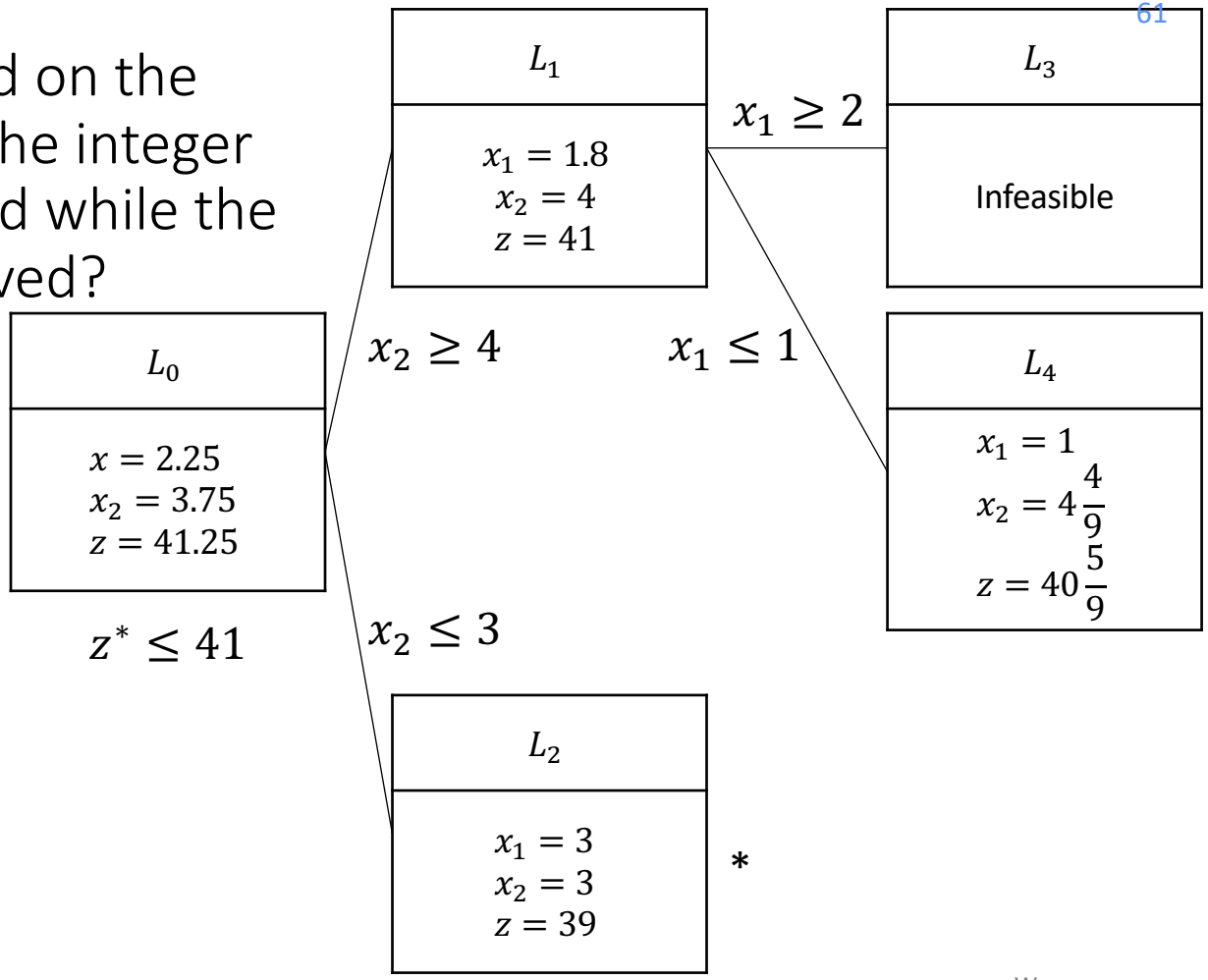
- There are three points that have yet to be considered with respect to the branch-and-bound procedure:
  - What is the best way to subdivide a given region, and which unanalyzed subdivision should be considered next?
  - Can the upper bound ($z = 41$, in the example) on the optimal value z of the integer program be improved while the problem is being solved?

What is the best way to subdivide a given region, and which unanalyzed subdivision should be considered next?

- If we can make our choice of subdivisions in such a way as to rapidly obtain a good (with luck, near-optimal) integer solution z', then we can eliminate many potential subdivisions immediately.
  - as if any region has its linear programming value $z < z'$, then the objective value of no integer point in that region can exceed z' and the region need not be subdivided.
- There is no universal method for making the required choice, although several heuristic procedures have been suggested, such as selecting the subdivision:
  - with the largest optimal linear-programming value.
  - on a last-generated–first-analyzed basis (depth-first).
- Rules for determining which fractional variables to use in constructing subdivisions are more subtle…

Can the upper bound on the optimal value z^* of the integer program be improved while the problem is being solved?

- Initial upper bound = 41.25
- Current upper bound = $40\frac{5}{9}$



Summary

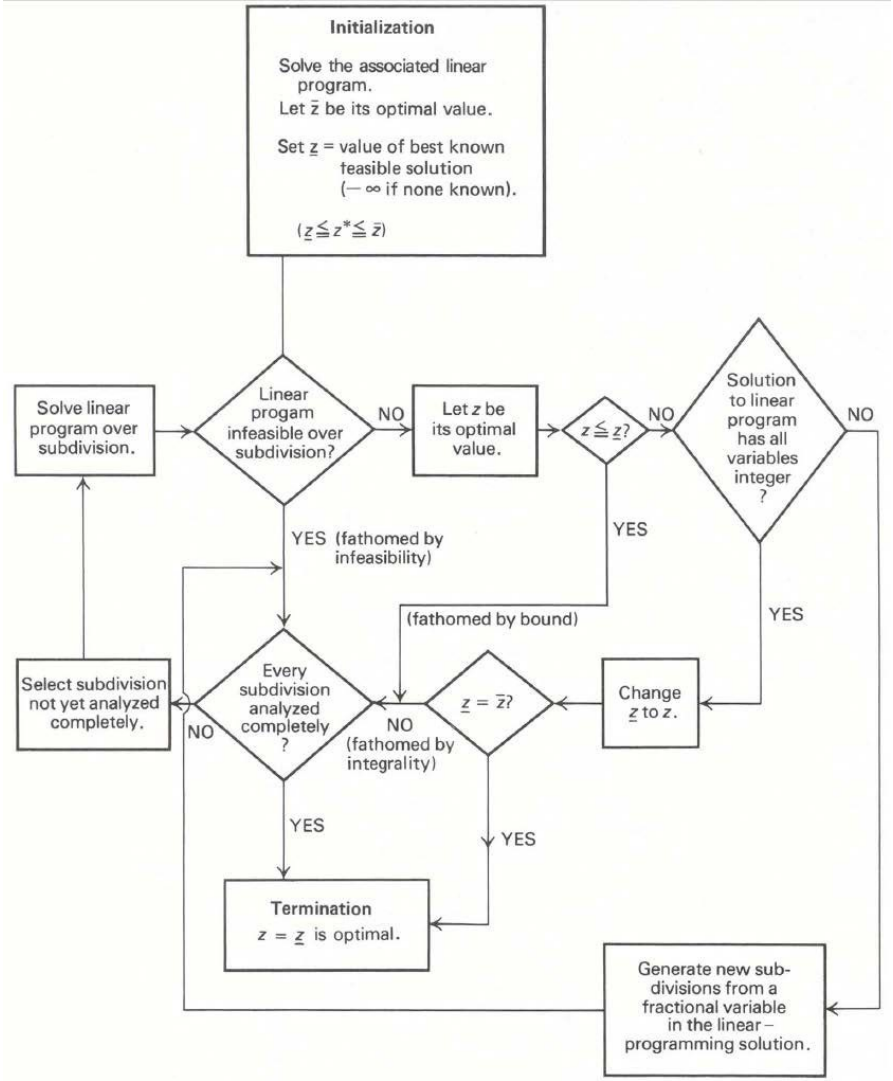
- The essential idea of branch-and-bound is to subdivide the feasible region to develop bounds $z_L < z^* < z^U$ on z^*.
- For a maximization problem, the lower bound z_L is the highest value of any feasible integer point encountered.
- The upper bound is given by the optimal value of the associated linear program or by the largest value for the objective function at any ‘‘hanging’’ box.
- After considering a subdivision, we must branch to (move to) another subdivision and analyze it.

Summary

- If either
  1. the linear program over L_j is infeasible;
  2. the optimal linear-programming solution over L_j is integer; or
  3. the value of the linear-programming solution z^j over L_j satisfies $z^j \leq z_L$ (if maximizing),

  then L_j need not be subdivided. In these cases, integer-programming terminology says that L_j has been fathomed.
- Case
  1. is termed fathoming by infeasibility,
  2. fathoming by integrality, and
  3. fathoming by bounds.

Branch-and-bound Flowchart



References

1. Bradley, Stephen P., Arnoldo C. Hax, and Thomas L. Magnanti. **Applied mathematical programming**. Addison-Wesley (1977). Chapter 9 Integer Programming

   - Companion slides of Applied Mathematical Programming by Bradley, Hax, and Magnanti (Addison-Wesley, 1977) prepared by José Fernando Oliveira Maria Antónia Carravilla