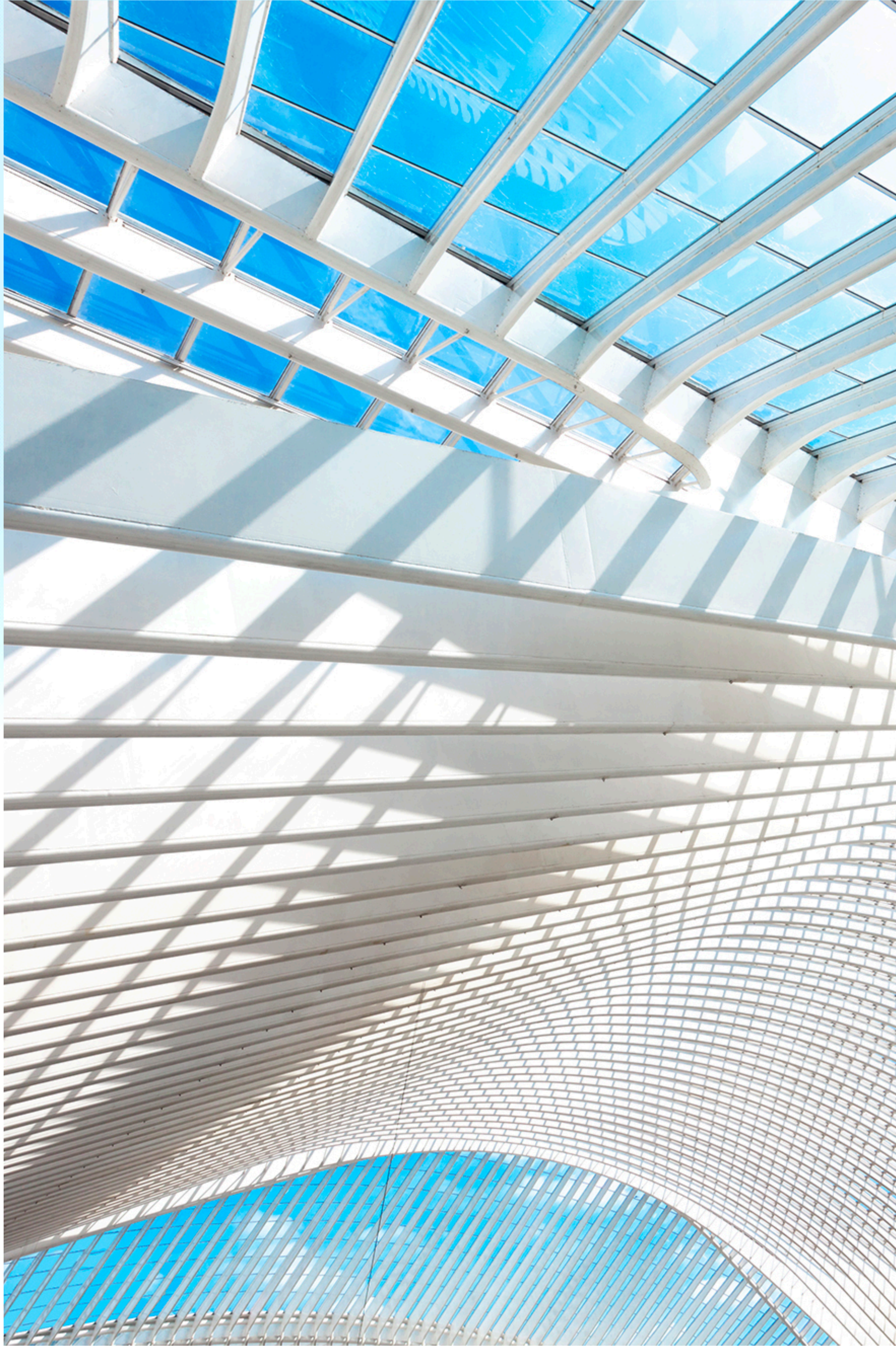
Mathematical Modeling

Chapter 6

Optimization of Discrete Model



Content

1. An Overview of Optimization Modeling

- We consider the following **optimization problem**:

$$\text{Optimize} \quad f_j(\mathbf{X}) \quad \text{for} \quad j \in J$$

subject to

$$g_i(\mathbf{X}) \left\{ \begin{array}{c} \geq \\ = \\ \leq \end{array} \right\} b_i \quad \text{for all} \quad i \in I,$$

where I and J are some finite sets. Here:

There are J objective functions $f_1, f_2, \ldots, f_J$ and each has I **constraints**, $g_1(\mathbf{X}) \left\{ \begin{array}{c} \geq \\ = \\ \leq \end{array} \right\} b_1$, $g_2(\mathbf{X}) \left\{ \begin{array}{c} \geq \\ = \\ \leq \end{array} \right\} b_2, \ldots,$ $g_I(\mathbf{X}) \left\{ \begin{array}{c} \geq \\ = \\ \leq \end{array} \right\} b_I$

The vector variable $\mathbf{X} = \left(x_1, x_2, \ldots, x_k\right)$ are called **decision variables.**

1. An Overview of Optimization Modeling

- The word **“optimize”** can be **“minimize”** or **“maximize” .**

- The goal is to find the decision vector variable $\mathbf{X}_0$ giving the optimal value for the set of functions $f_j(\mathbf{X})$, yet satisfying the constraints

$$g_i(\mathbf{X}) \begin{Bmatrix} \geq \\ = \\ \leq \end{Bmatrix} b_i \quad \text{for all} \quad i \in I$$

- In this Chapter, we focus the case where $\mathbf{X}$ is a vector of discrete values.

1. An Overview of Optimization Modeling

Example 1.1. *Determining a Production Schedule.*

Problem. A carpenter makes tables and bookcases. He is trying to determine how many of each type of furniture he should make each week. **The carpenter wishes to determine a weekly production schedule for tables and bookcases** that **maximizes his profits**. It costs \$5 and \$7 to produce tables and bookcases, respectively. The revenues are estimated by the expressions:

$50x_1 - 0.2x_1^2$, where x_1 = number of tables produced per week and

$65x_2 - 0.3x_2^2$, where x_2 = number of bookcases produced per week

- What is the goal?
- What is (are) your objective function(s)?
- What is the decision vector variable?
- Are there any constraints?

1. An Overview of Optimization Modeling

Example 1.1. *Determining a Production Schedule.*

Solution.

- What is the goal? Maximizing total profit
- What is the decision vector variable? $\mathbf{X} = (x_1, x_2)$
- Objective function = $f(x_1, x_2)$ = Total profit = Total revenue — Total cost

$$\text{Total revenue } = (50x_1 - 0.2x_1^2) + (65x_2 - 0.3x_2^2)$$

$$\text{Total cost } = 5x_1 + 7x_2$$

$$\text{Total profit } = (50x_1 - 0.2x_1^2) + (65x_2 - 0.3x_2^2) - 5x_1 - 7x_2$$

$$= 45x_1 - 58x_2 - 0.2x_1^2 - 0.3x_2^2$$

- There is no constraints except $x_1, x_2 \in \mathbb{N}$.

1. An Overview of Optimization Modeling

Example 1.2. *Determining a Production Schedule.*

Problem. The carpenter realizes **a net unit profit of $25 per table** and **$30 per bookcase**. He is trying to determine how many of each piece of furniture he should make each week. **He has up to 690 board-feet of lumber to devote weekly to the project and up to 120 hr of labor.** He can use lumber and labor productively elsewhere if they are not used in the production of tables and bookcases. He estimates that **it requires 20 board-feet of lumber and 5 hr of labor to complete a table and 30 board-feet of lumber and 4 hr of labor for a bookcase.** Moreover, he has signed contracts to **deliver at least four tables and two bookcases every week**. The carpenter **wishes to determine a weekly production schedule for tables and bookcases** that **maximizes his profits.**

1. An Overview of Optimization Modeling

Example 1.2. *Determining a Production Schedule.*

Question. Formulate the above problem.

- What is goal of the problem?
- What is (are) objective function(s)?
- What is the decision vector variable?
- Are there any constraints?

1. An Overview of Optimization Modeling

Example 1.2. *Determining a Production Schedule.*

Solution.

- What is the goal? Maximizing total profit
- What is the decision vector variable? $\mathbf{X} = (x_1, x_2)$
- Objective function = $f(x_1, x_2) = 25x_1 + 30x_2$
- There are constraints:

$$20x_1 + 30x_2 \leq 690 \qquad \text{(lumber)}$$

$$5x_1 + 4x_2 \leq 120 \qquad \text{(labor)}$$

$$x_1 \geq 4 \qquad \text{(contract)}$$

$$x_2 \geq 2 \qquad \text{(contract)}$$

1. An Overview of Optimization Modeling

Classifying Some Optimization Problems

- ***Constrained*** (if there there no constraint) or ***Unconstrained*** (if there is (are) constraint(s))
- ***Linear*** or ***nonlinear***
- ***Single-objective*** or ***multiobjective***
- ***Stochastic*** if the problem is probabilistic
- ***Dynamic*** if the problem is time-dependent
- ***Discrete (integer)*** or ***continuous*** or ***mixed-integer***

1. An Overview of Optimization Modeling

Unconstrained Optimization Problems.

Find the parameters of the model $y = f(x)$ to

$$\text{minimize} \sum_{i=1}^{m} \left| y_i - f(x_i) \right|,$$

where

$(x_i, y_i),\ i = 1, 2, \ldots, m$ are given.

Constrained Optimization Problems, see Example 1.2 above.

1. An Overview of Optimization Modeling

Example 2. Space Shuttle Cargo.

- There are various items to be taken on a space shuttle. Unfortunately, there are restrictions on the allowable weight and volume capacities.

- Suppose there are m different items, each given some numerical value c_j and having weight w_j and volume v_j. (How might you determine c_j in an actual problem?)

- Suppose the goal is to maximize the value of the items that are to be taken without exceeding the weight limitation W or the volume limitation V. We can formulate this model

1. An Overview of Optimization Modeling

Example 2. Space Shuttle Cargo.

$$\text{Let } y_j = \begin{cases} 1, & \text{if item } j \text{ is taken (yes)} \\ 0, & \text{if item } j \text{ is not taken (no)} \end{cases}$$

Then the problem is

$$\text{Maximize } \sum_{j=1}^{m} c_j y_j$$

subject to

$$\sum_{j=1}^{m} v_j y_j \le V$$

$$\sum_{j=1}^{m} w_j y_j \le W$$

1. An Overview of Optimization Modeling

Example 3. Approximation by a Piecewise Linear Function.

- Suppose the nonlinear function in Figure 7.1a represents a cost function and we want to find its minimum value over the interval $0 \leq x \leq a_3$.

If the function is particularly complicated, it could be approximated by a piecewise linear function such as that shown in Figure 7.1b. (The piecewise linear function might occur naturally in a problem, such as when different rates are charged for electrical use based on the amount of consumption.)

When we use the approximation suggested in Figure 7.1, our problem is to find the minimum of the function:

1. An Overview of Optimization Modeling

Example 3. Approximation by a Piecewise Linear Function.

$$c(x) = \begin{cases} b_1 + k_1(x - 0) & \text{if } 0 \le x \le a_1 \\ b_2 + k_2(x - a_1) & \text{if } a_1 \le x \le a_2 \\ b_3 + k_3(x - a_2) & \text{if } a_2 \le x \le a_3 \end{cases}$$

Define the three new variables: $x_1 = (x - 0)$, $x_2 = (x - a_1)$, and $x_3 = (x - a_2)$, for each of the three intervals, and use the binary variablesy_1,y_2, and y_3 to restrict the x_i to the appropriate interval:

1. An Overview of Optimization Modeling

Example 3. Approximation by a Piecewise Linear Function.

$$0 \le x_1 \le y_1 a_1$$
$$0 \le x_2 \le y_2(a_2 - a_1)$$
$$0 \le x_3 \le y_3(a_3 - a_2)$$

where y_1,y_2, and y_3 are equal to 0 or 1. Because we want exactly one x_i to be active at any time, we impose the following constraint:

$$y_1 + y_2 + y_3 = 1.$$

Now that only one of the x_i is active at any one time, our objective function becomes

1. An Overview of Optimization Modeling

Example 3. Approximation by a Piecewise Linear Function.

$$c(x) = y_1(b_1 + k_1 x_1) + y_2(b_2 + k_2 x_2) + y_3(b_3 + k_3 x_3)$$

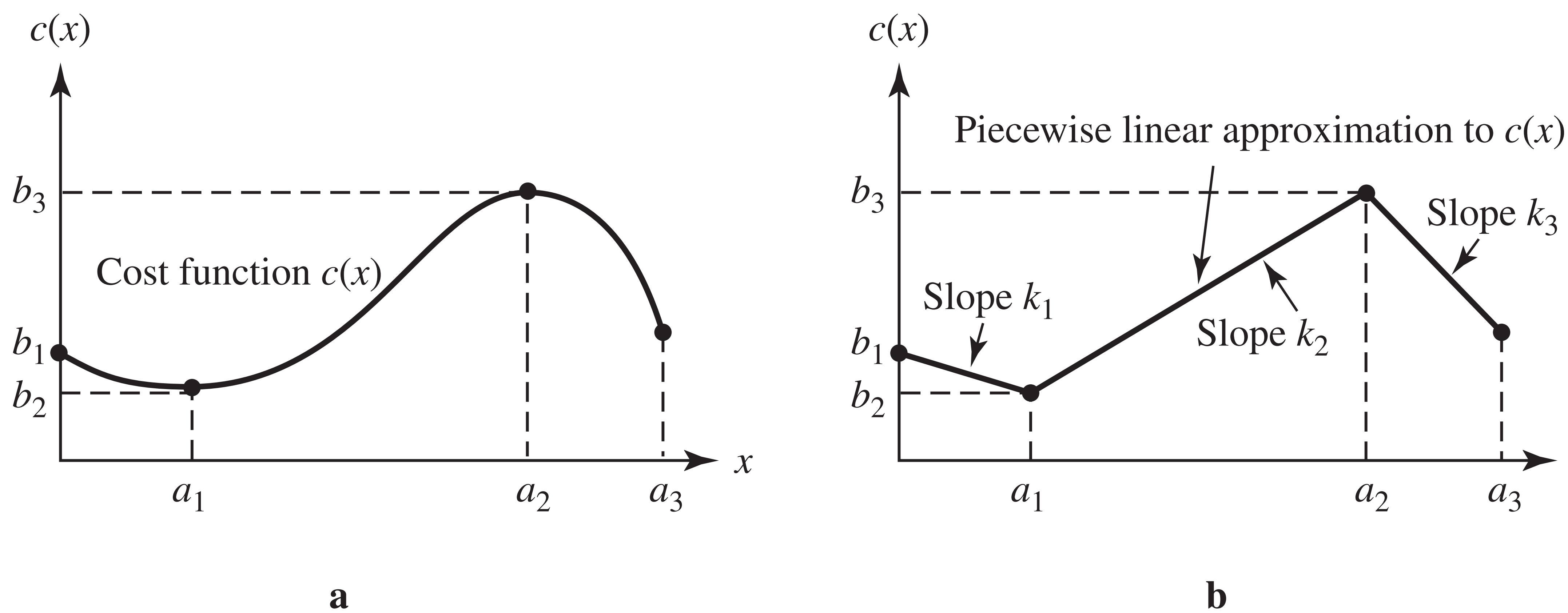


Figure 7.1

Using a piecewise linear function to approximate a nonlinear function

1. An Overview of Optimization Modeling

Example 3. Approximation by a Piecewise Linear Function.

Observe that whenever $y_i = 0$, the variable $x_i = 0$ as well. Thus, products of the form $x_i y_i$ are redundant, and the objective function can be simplified to give the following model:

$$\text{Minimize } k_1x_1 + k_2x_2 + k_3x_3 + y_1b_1 + y_2b_2 + y_3b_3$$

subject to

$$\begin{aligned} &0 \le x_1 \le y_1a_1 \\ &0 \le x_2 \le y_2(a_2 - a_1) \\ &0 \le x_3 \le y_3(a_3 - a_2) \\ &y_1 + y_2 + y_3 = 1 \end{aligned}$$

where y_1, y_2, and y_3 equal 0 or 1.

2. Linear Programming I: Geometric Solutions

Linear Programming.

An optimization problem is said to be a linear program if it satisfies the following properties:

1. There is a unique objective function.

2. Whenever a decision variable appears in either the objective function or one of the constraint functions, it must appear only as a power term with an exponent of 1, possibly multiplied by a constant.

3. No term in the objective function or in any of the constraints can contain products of the decision variables.

4. The coefficients of the decision variables in the objective function and each constraint are constant.

5. The decision variables are permitted to assume fractional as well as integer values.

2. Linear Programming I: Geometric Solutions

Terms in linear programming:

- *Nonnegative constraints*
- *Feasible set*
- *Feasible vs infeasible solution*
- *Convex set*

2. Linear Programming I: Geometric Solutions

Illustration:

Figure 7.2

The feasible region for the constraints $x_1 + 2x_2 \leq 4$, $x_1, x_2 \geq 0$

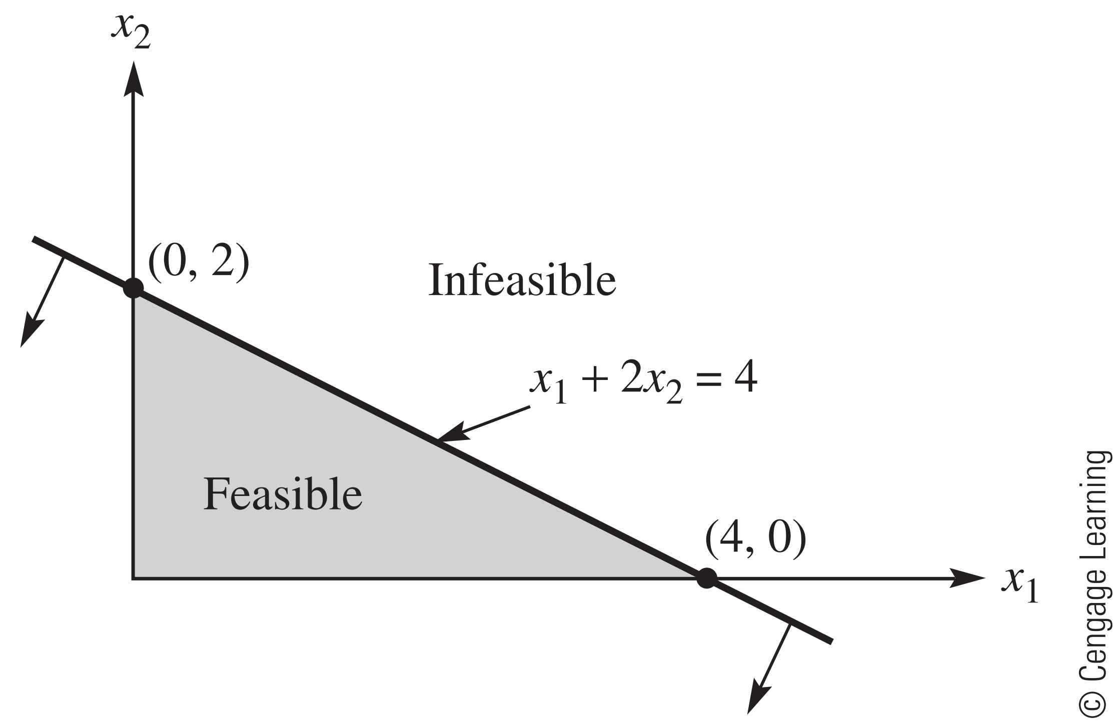
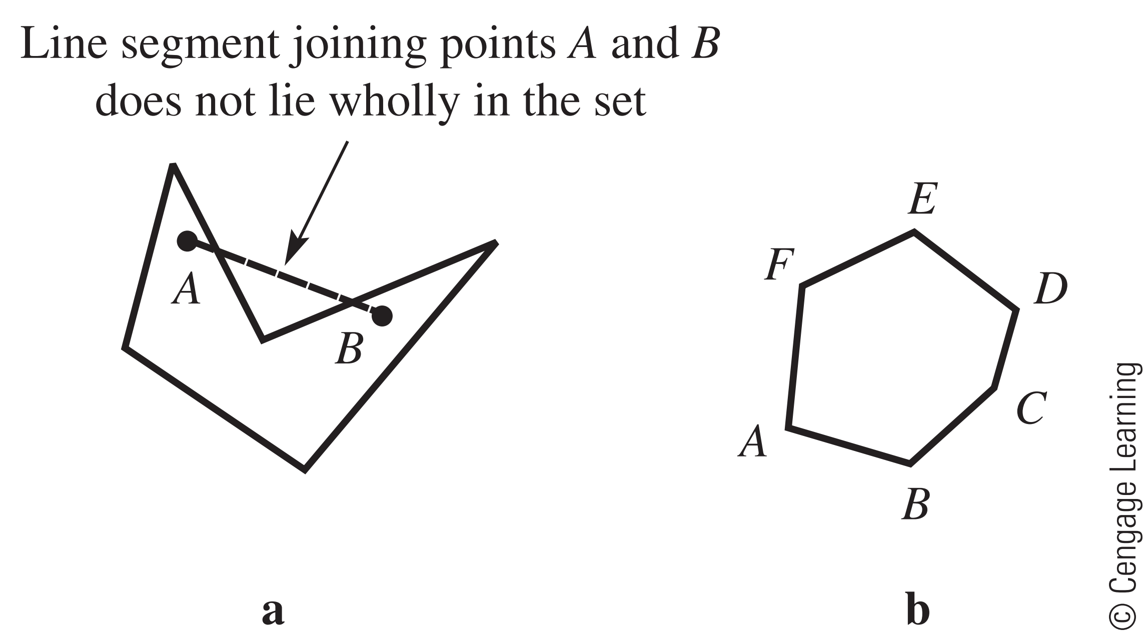


Figure 7.3

The set shown in **a** is not convex, whereas the set shown in **b** is convex.



2. Linear Programming I: Geometric Solutions

Theorem 1

Suppose the feasible region of a linear program is a nonempty and bounded convex set. Then the objective function must attain both a maximum and a minimum value occurring at extreme points of the region. If the feasible region is unbounded, the objective function need not assume its optimal values. If either a maximum or a minimum does exist, it must occur at one of the extreme points.

2. Linear Programming I: Geometric Solutions

Example. Consider a version of the carpenter's problem from Section 7.1.

The carpenter realizes a net unit profit of \$25 per table and \$30 per bookcase. He is trying to determine how many tables (x_1) and how many bookcases (x_2) he should make each week.

He has up to 690 board-feet of lumber and up to 120 hours of labor to devote weekly to the project. The lumber and labor can be used productively elsewhere if not used in the production of tables and bookcases.

He estimates that it requires 20 board-feet of lumber and 5 hr of labor to complete a table and 30 board-feet of lumber and 4 hr of labor for a bookcase.

2. Linear Programming I: Geometric Solutions

This formulation yields

$$\text{Maximize } 25x_1 + 30x_2$$

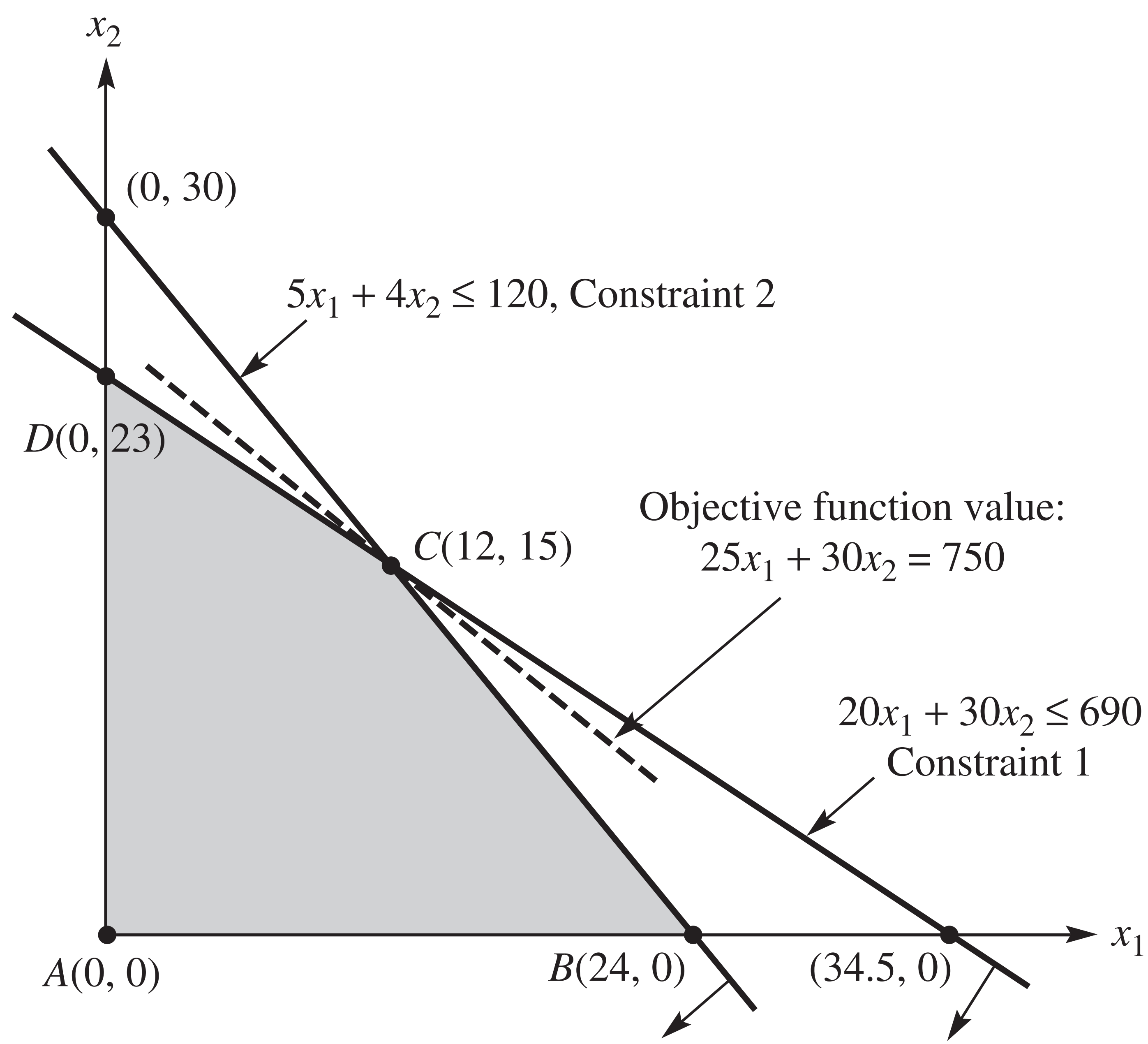
subject to

$$\begin{aligned} 20x_1 + 30x_2 &\leq 690 \qquad &\text{(lumber)} \\ 5x_1 + 4x_2 &\leq 120 \qquad &\text{(labor)} \\ x_1, x_2 &\geq 0 \qquad &\text{(nonnegativity)} \end{aligned}$$

2. Linear Programming I: Geometric Solutions

Figure 7.4

The set of points satisfying the constraints of the carpenter's problem form a convex set.



2. Linear Programming I: Geometric Solutions

If an optimal solution to a linear program exists, it must occur among the extreme points of the convex set formed by the set of constraints.

The values of the objective function (profit for the carpenter's problem) at the extreme points are

Extreme point	Objective function value
A (0, 0)	\$0
B (24, 0)	600
$\boxed{C\ (12, 15)}$	$\boxed{750}$
D (0, 23)	690

Maximum at $C(12, 15)$.
Maximum weekly profit = \$750

2. Linear Programming I: Geometric Solutions

Example. Data Fitting-Problem. Consider using the Chebyshev criterion to fit the model $y = cx$ to the following data set:

x	1	2	3
y	2	5	8

- The **optimization problem** that determines the parameter c to minimize the largest absolute deviation $r_i = \left| y_i - y(x_i) \right|$ (residual or error) is the linear program:

$$\text{Minimize } r$$

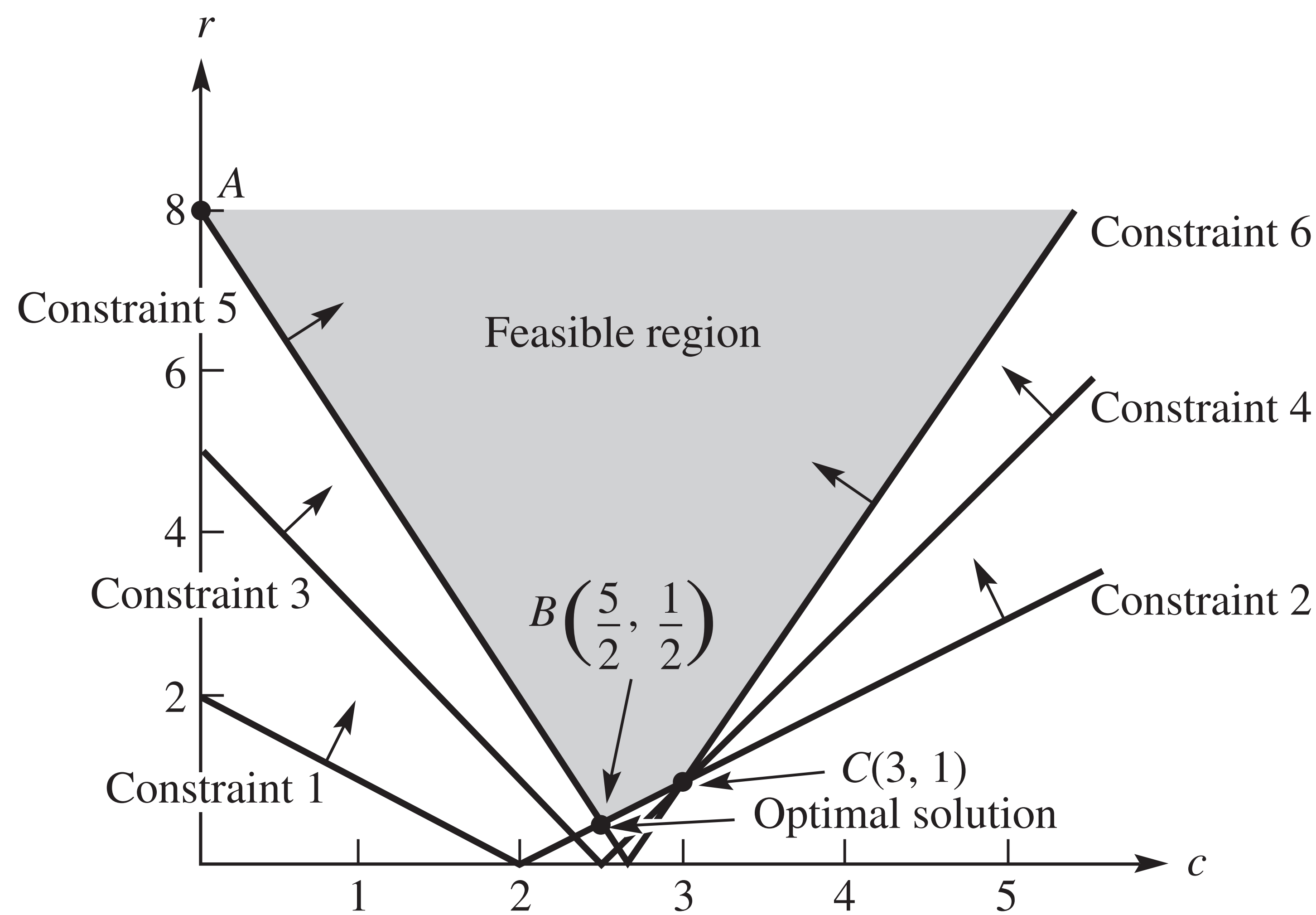
subject to

$$\left.\begin{array}{rl} r - (2 - c) \geq 0 & \text{(constraint 1)} \\ r + (2 - c) \geq 0 & \text{(constraint 2)} \\ r - (5 - 2c) \geq 0 & \text{(constraint 3)} \\ r + (5 - 2c) \geq 0 & \text{(constraint 4)} \\ r - (8 - 3c) \geq 0 & \text{(constraint 5)} \\ r + (8 - 3c) \geq 0 & \text{(constraint 6)} \end{array}\right\}$$

2. Linear Programming I: Geometric Solutions

Figure 7.5

The feasible region for fitting $y = cx$ to a collection of data



Three points of the boundary:

$A(0, 8)$

$B(5/2, 1/2)$

$C(3, 1)$

2. Linear Programming I: Geometric Solutions

Extreme point	Objective function value
(c, r)	$f(r) = r$
A	8
$\boxed{B}$	$\boxed{\frac{1}{2}}$
C	1

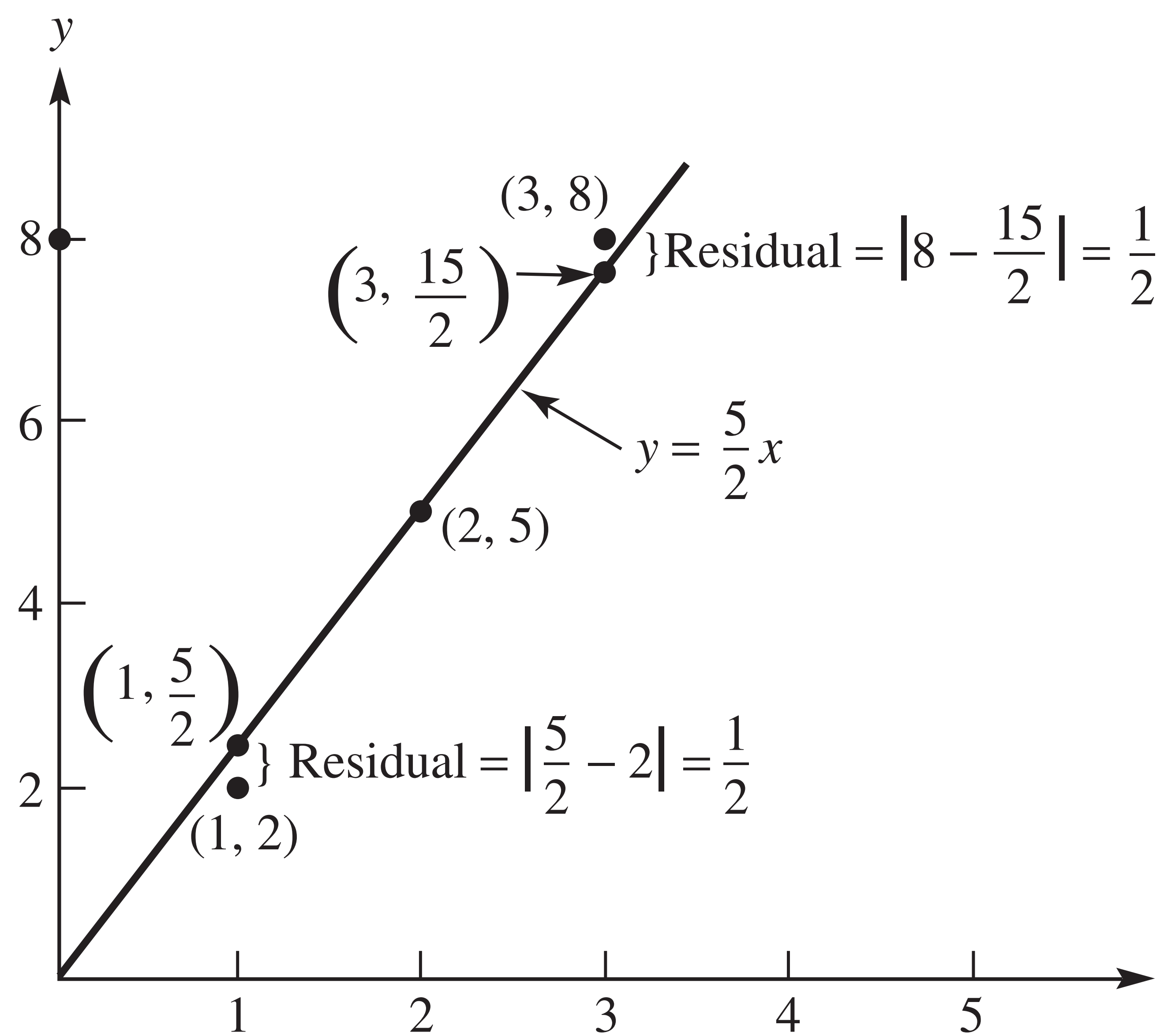
Minimum at $B(5/2, 1/2)$. Thus, $c = 5/2$

2. Linear Programming I: Geometric Solutions

Model Interpretation.

Figure 7.6

The line $y = (5/2)x$ results in a largest absolute deviation $r_{\max} = \frac{1}{2}$, the smallest possible $r_{\max}$.



3. Linear Programming II: Algebraic Solutions

Linear Programming.

The procedure for finding an optimal solution to a linear program with a nonempty and bounded feasible region:

1. Find all intersection points of the constraints.

2. Determine which intersection points, if any, are feasible to obtain the extreme points.

3. Evaluate the objective function at each extreme point.

4. Choose the extreme point(s) with the largest (or smallest) value for the objective function.

To implement this procedure algebraically, we must characterize the **intersection points** and the **extreme points**.

3. Linear Programming II: Algebraic Solutions

Consider a general linear programming problem:

$$\text{Optimize} \quad f\left(x_1, x_2, \ldots, x_m\right)$$

subject to

$$\begin{aligned} g_1\left(x_1, x_2, \ldots, x_m\right) &\le c_1 \\ g_2\left(x_1, x_2, \ldots, x_m\right) &\le c_2 \\ &\vdots \\ g_n\left(x_1, x_2, \ldots, x_m\right) &\le c_n \\ x_1, x_2, \ldots, x_m &\ge 0 \end{aligned}$$

If there is a constraint written in the form:
$g_i\left(x_1, x_2, \ldots, x_m\right) \ge c_i$,
we just convert it to
$-g_i\left(x_1, x_2, \ldots, x_m\right) \le -c_i$,
then it will be alright.

That is, we have m decision variables, m nonnegative constraints, and n constraints.

3. Linear Programming II: Algebraic Solutions

Solving general linear programming problem algebraically, do the following steps:

Step 1. Add "slack" variable $y_i \geq 0$ to each constraint $g_i\left(x_1, x_2, \ldots, x_m\right) \leq c_i$ so that we have equality constraint $g_i\left(x_1, x_2, \ldots, x_m\right) + y_i = c_i$, for all $i = 1, 2, \ldots, n$.

Step 2. Rewrite the constraints:

$$
\begin{aligned}
g_1\left(x_1, x_2, \ldots, x_m\right) + y_1 &= c_1 \\
g_2\left(x_1, x_2, \ldots, x_m\right) + y_2 &= c_2 \\
&\vdots \\
g_n\left(x_1, x_2, \ldots, x_m\right) + y_n &= c_n \\
x_1, x_2, \ldots, x_m, y_1, y_2, \ldots, y_n &\geq 0
\end{aligned}
$$

That is, we have $(m+n)$ decision variables, $(m+n)$ nonnegative constraints, and n **equality** constraints.

3. Linear Programming II: Algebraic Solutions

Step 3. Determine the intersection points.

To determine an intersection point, choose m of the variables (because **originally we have** m **decision variables**) and set them to zero.

Computational Complexity. There are $C_{m+n}^{m} = \dfrac{(m+n)!}{m!n!}$ choices of such combination!

Step 4. Decide which intersection points are feasible and infeasible.

Step 5. Compute the objective function at all feasible intersection points and choose the optimal one.

3. Linear Programming II: Algebraic Solutions

Example. Solving the Carpenter's Problem Algebraically

The carpenter's model is

$$\text{Maximize } 25x_1 + 30x_2$$

subject to

$$\begin{aligned} 20x_1 + 30x_2 &\le 690 \quad &\text{(lumber)} \\ 5x_1 + 4x_2 &\le 120 \quad &\text{(labor)} \\ x_1, x_2 &\ge 0 \quad &\text{(nonnegativity)} \end{aligned}$$

3. Linear Programming II: Algebraic Solutions

Following **Step 1** to **Step 2**, we get

$$\text{Maximize} \quad 25x_1 + 30x_2$$

subject to

$$\begin{aligned} 20x_1 + 30x_2 + y_1 &= 690 \\ 5x_1 + 4x_2 + y_2 &= 120 \\ x_1, x_2, y_1, y_2 &\geq 0 \end{aligned}$$

Step 3. Determine all possible intersection points. There are $C_4^2 = 6$ possible intersection points.

3. Linear Programming II: Algebraic Solutions

Steps 3 & 4. Determine all possible intersection points in the equality constraints. There are $C_4^2 = 6$ possible intersection points.

Case 1. $x_1 = 0, x_2 = 0 \implies y_1 = 690, y_2 = 120$

Case 2. $x_1 = 0, y_1 = 0 \implies x_2 = 23, y_2 = 28$

Case 3. $x_1 = 0, y_2 = 0 \implies x_2 = 30, y_1 = -210$ **(infeasible)**

Case 4. $x_2 = 0, y_1 = 0 \implies x_1 = 34.5, y_2 = -52.5$ **(infeasible)**

Case 5. $x_2 = 0, y_2 = 0 \implies x_1 = 24, y_1 = 210$

Case 6. $y_1 = 0, y_2 = 0 \implies x_1 = 12, x_2 = 15$

3. Linear Programming II: Algebraic Solutions

Step 5. Compute the objective function at all feasible intersection points and choose the optimal one.

Extreme point	Value of objective function
$A(0,0)$	\$0
$D(0,23)$	690
$C(12,15)$	750
$B(24,0)$	600

Our procedure determines that the optimal solution to maximize the profit is $x_1 = 12, x_2 = 15$ .That is, the carpenter should make 12 tables and 15 bookcases for a maximum profit of $750.

3. Linear Programming II: Algebraic Solutions

Computational Complexity: Intersection Point Enumeration

Obviously, as the size of the linear program increases (in terms of the numbers of decision variables and constraints), this technique of enumerating all possible intersection points becomes unwieldy, even for powerful computers.

How can we improve the procedure?

3. Linear Programming II: Algebraic Solutions

Computational Complexity: Solution?

- Note that we enumerated some intersection points in the carpenter example that turned out to be infeasible. Is there a way to quickly identify that a possible intersection point is infeasible?

- Moreover, if we have found an extreme point (i.e., a feasible intersection point) and know the corresponding value of the objective function, can we quickly determine if another proposed extreme point will improve the value of the objective function?

- We desire a procedure that does not enumerate infeasible intersection points and that enumerates only those extreme points that improve the value of the objective function for the best solution found so far in the search.