- 教材讨论
  - TC第29章

General Linear Programming

- Linear function

$$f(x_1, x_2, \ldots, x_n) = a_1x_1 + a_2x_2 + \cdots + a_nx_n = \sum_{j=1}^{n} a_jx_j \ .$$

- Linear equality

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

- Linear inequality

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

Food with Nutrients

- There are m different types of food, $F_1,\ldots,F_m$, that supply varying quantities of the n nutrients, $N_1,\ldots,N_n$, that are essential to good health. Let c_j be the minimum daily requirement of nutrient, N_j . Let b_i be the price per unit of food, F_i . Let a_{ij} be the amount of nutrient N_j contained in one unit of food F_i . The problem is to supply the required nutrients at minimum cost.

The Model

- Let y_i be the number of units of food F_i to be purchased per day. The cost per day of such a diet is

$$b_1 * y_1 + b_2 * y_2 + \cdots + b_m * y_m. \ (1)$$

- The amount of nutrient N_j contained in this diet is

$$a_{1j} * y_1 + a_{2j} * y_2 + \cdots + a_{mj} * y_m$$

- for $j = 1,\ldots,n$. We do not consider such a diet unless all the minimum daily requirements are met, that is, unless

$$a_{1j} * y_1 + a_{2j} * y_2 + \cdots + a_{mj} * y_m \geq c_j \quad \text{for } j = 1,\ldots,n. \ (2)$$

- Of course, we cannot purchase a negative amount of food, so we automatically have the Constraints

$$y_1 \geq 0, y_2 \geq 0, \ldots, y_m \geq 0. \ (3)$$

- Our problem is: minimize (1) subject to (2) and (3). This is exactly the standard minimum problem.

The Job Assignment

- There are I persons available for J jobs. The value of person i working 1 day at job j is a_{ij} , for $i = 1,...,I$, and $j = 1,...,J$. The problem is to choose an assignment of persons to jobs to maximize the total value.

The Model

- An assignment is a choice of numbers, x_{ij} , for $i = 1,\ldots,I$, and $j = 1,\ldots,J$, where *xij* represents the proportion of person *i*'s time that is to be spent on job *j* . Thus,

$$\sum_{j=1}^{J} x_{ij} \leq 1 \ \ for\ i = 1, \ldots I \quad (1)$$

$$\sum_{i=1}^{I} x_{ij} \leq 1 \ \ for\ j = 1, \ldots I \quad (2)$$

$$x_{ij} \geq 0 \ \ for\ i = 1, \ldots, I\ and\ j = 1, \ldots, J \quad (3)$$

- Equation (1) reflects the fact that a person cannot spend more than 100% of his time working, (2) means that only one person is allowed on a job at a time, and (3) says that no one can work a negative amount of time on any job. Subject to (1), (2) and (3), we wish to maximize the total value,

$$\sum_{i=1}^{I} \sum_{j=1}^{J} a_{ij} x_{ij}$$

Production Scheduling

- A company is involved in the production of two items (X and Y). The resources need to produce X and Y are twofold, namely machine time for automatic processing and craftsman time for hand finishing. The table below gives the number of minutes required for each item:

	Machine time	Craftsman time
X	13	20
Y	19	29

- The company has 40 hours of machine time available in the next working week but only 35 hours of craftsman time. Machine time is costed at £10 per hour worked and craftsman time is costed at £2 per hour worked. Both machine and craftsman idle times incur no costs. The revenue received for each item produced (all production is sold) is £20 for X and £30 for Y. The company has a specific contract to produce 10 items of X per week for a particular customer.
- Formulate the problem of deciding how much to produce per week as a linear program.

The Model

- Let

  x be the number of items of X

  y be the number of items of Y
- then the LP is:
- maximize

  $20x + 30y$ - 10(machine time worked) - 2(craftsman time worked)
- subject to:

$$13x + 19y <= 40*60 \text{ machine time}$$
$$20x + 29y <= 35*60 \text{ craftsman time}$$
$$x >= 10 \text{ contract}$$
$$x, y >= 0$$

- so that the objective function becomes maximize

$$20x + 30y - 10(13x + 19y)/60 - 2(20x + 29y)/60$$

Standard Form

In ***standard form***, we are given n real numbers $c_1, c_2, \ldots, c_n$; m real numbers $b_1, b_2, \ldots, b_m$; and mn real numbers a_{ij} for $i = 1, 2, \ldots, m$ and $j = 1, 2, \ldots, n$. We wish to find n real numbers $x_1, x_2, \ldots, x_n$ that

$$\begin{array}{lrcll}
\text{maximize} & \displaystyle\sum_{j=1}^{n} c_j x_j & & & \\
\text{subject to} & & & & \\
& \displaystyle\sum_{j=1}^{n} a_{ij} x_j & \leq & b_i & \text{for } i = 1, 2, \ldots, m \\
& x_j & \geq & 0 & \text{for } j = 1, 2, \ldots, n\,.
\end{array}$$

To Standard Form

- 4 possible reasons that we need to convert a linear program to a standard form.

1. The objective function might be a minimization rather than a maximization.
2. There might be variables without nonnegativity constraints.
3. There might be ***equality constraints***, which have an equal sign rather than a less-than-or-equal-to sign.
4. There might be ***inequality constraints***, but instead of having a less-than-or-equal-to sign, they have a greater-than-or-equal-to sign.

Slack Form

- (N, B, A, b, c, v)

$$
\begin{aligned}
z &= v + \sum_{j \in N} c_j x_j \\
x_i &= b_i - \sum_{j \in N} a_{ij} x_j \quad \text{for } i \in B ,
\end{aligned}
$$

- *basic variable* and *nonbasic variable*
- *tight*

$$
\begin{array}{lrcrcrcr}
\text{maximize} & 3x_1 & + & x_2 & + & 2x_3 & & \\
\text{subject to} & & & & & & & \\
& x_1 & + & x_2 & + & 3x_3 & \leq & 30 \\
& 2x_1 & + & 2x_2 & + & 5x_3 & \leq & 24 \\
& 4x_1 & + & x_2 & + & 2x_3 & \leq & 36 \\
& & x_1, x_2, x_3 & & & & \geq & 0 .
\end{array}
\Rightarrow
\begin{array}{rcrcrcrcr}
z & = & & & 3x_1 & + & x_2 & + & 2x_3 \\
x_4 & = & 30 & - & x_1 & - & x_2 & - & 3x_3 \\
x_5 & = & 24 & - & 2x_1 & - & 2x_2 & - & 5x_3 \\
x_6 & = & 36 & - & 4x_1 & - & x_2 & - & 2x_3
\end{array}
$$

Pivoting

- Wh
  - a ... d the
    e ... lled
    t ... es.

PIVOT(N, B, A, b, c, v, l, e)

```
// Compute the coefficients of the equation for new basic variable x_e.
let Â be a new m × n matrix
b̂_e = b_l / a_le
for each j ∈ N − {e}
    â_ej = a_lj / a_le
â_el = 1 / a_le
// Compute the coefficients of the remaining constraints.
for each i ∈ B − {l}
    b̂_i = b_i − a_ie b̂_e
    for each j ∈ N − {e}
        â_ij = a_ij − a_ie â_ej
    â_il = −a_ie â_el
// Compute the objective function.
v̂ = v + c_e b̂_e
for each j ∈ N − {e}
    ĉ_j = c_j − c_e â_ej
ĉ_l = −c_e â_el
// Compute new sets of basic and nonbasic variables.
N̂ = N − {e} ∪ {l}
B̂ = B − {l} ∪ {e}
return (N̂, B̂, Â, b̂, ĉ, v̂)
```

Simplex Algorithm

- How do we determine whether a linear program is feasible?
- What do we do if the linear program is feasible, but the initial basic solution is not feasible?
- How do we determine whether a linear program is unbounded?
- How do we choose the entering and leaving variables?

Simplex Procedure

```
SIMPLEX(A, b, c)
(N, B, A, b, c, v) = INITIALIZE-SIMPLEX(A, b, c)
let Δ be a new vector of length n
while some index j ∈ N has c_j > 0
    choose an index e ∈ N for which c_e > 0
    for each index i ∈ B
        if a_ie > 0
            Δ_i = b_i / a_ie
        else Δ_i = ∞
    choose an index l ∈ B that minimizes Δ_i
    if Δ_l == ∞
        return "unbounded"
    else (N, B, A, b, c, v) = PIVOT(N, B, A, b, c, v, l, e)
```

1. the slack form is equivalent to the slack form returned by the call of INITIALIZE-SIMPLEX,
2. for each $i \in B$, we have $b_i \geq 0$, and
3. the basic solution associated with the slack form is feasible.

Termination

Assuming that INITIALIZE-SIMPLEX returns a slack form for which the basic solution is feasible, SIMPLEX either reports that a linear program is unbounded, or it terminates with a feasible solution in at most $\binom{n+m}{m}$ iterations.

Linear Programming Duality

$$\begin{aligned}
\text{maximize} \quad & \sum_{j=1}^{n} c_j x_j \\
\text{subject to} \quad & \\
& \sum_{j=1}^{n} a_{ij} x_j \le b_i \quad \text{for } i = 1, 2, \ldots, m \\
& x_j \ge 0 \quad \text{for } j = 1, 2, \ldots, n \,.
\end{aligned}$$

Standard linear program

Its dual

$$\begin{aligned}
\text{minimize} \quad & \sum_{i=1}^{m} b_i y_i \\
\text{subject to} \quad & \\
& \sum_{i=1}^{m} a_{ij} y_i \ge c_j \quad \text{for } j = 1, 2, \ldots, n \,, \\
& y_i \ge 0 \quad \text{for } i = 1, 2, \ldots, m \,.
\end{aligned}$$

Weak Linear-Programming Duality

Lemma 29.8 (Weak linear-programming duality)

Let $\bar{x}$ be any feasible solution to the primal linear program in (29.16)–(29.18) and let $\bar{y}$ be any feasible solution to the dual linear program in (29.83)–(29.85). Then, we have

$$\sum_{j=1}^{n} c_j \bar{x}_j \leq \sum_{i=1}^{m} b_i \bar{y}_i \ .$$

Linear-Programming Duality

Theorem 29.10 (Linear-programming duality)
Suppose that SIMPLEX returns values $\bar{x} = (\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_n)$ for the primal linear program (A, b, c). Let N and B denote the nonbasic and basic variables for the final slack form, let c' denote the coefficients in the final slack form, and let $\bar{y} = (\bar{y}_1, \bar{y}_2, \ldots, \bar{y}_m)$ be defined by equation (29.91). Then $\bar{x}$ is an optimal solution to the primal linear program, $\bar{y}$ is an optimal solution to the dual linear program, and

$$\sum_{j=1}^{n} c_j \bar{x}_j = \sum_{i=1}^{m} b_i \bar{y}_i \ . \tag{29.92}$$

It works well!

Theorem 29.13 (Fundamental theorem of linear programming)
Any linear program L, given in standard form, either

1. has an optimal solution with a finite objective value,
2. is infeasible, or
3. is unbounded.

If L is infeasible, SIMPLEX returns "infeasible." If L is unbounded, SIMPLEX returns "unbounded." Otherwise, SIMPLEX returns an optimal solution with a finite objective value.

Integer Linear Programming

- It is really hard!

- A special case, 0-1 integer linear programming