Линейное программирование

Основная задача линейного программирования

Стандартная форма

- Первая стандартная форма задачи линейного программирования имеет вид

$$
\begin{aligned}
&c_1x_1 + c_2x_2 + \ldots + c_nx_n \Rightarrow \max \\
&a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n \le b_1, \\
&a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n \le b_2, \\
&\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\
&a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n \le b_m, \\
&x_1 \ge 0; \quad x_2 \ge 0; \quad \ldots; \quad x_n \ge 0.
\end{aligned}
$$

Стандартная форма

- Вторая стандартная форма задачи линейного программирования имеет вид

$$
\begin{aligned}
&c_1x_1+c_2x_2+\ldots+c_nx_n \Rightarrow \min\\
&a_{11}x_1+a_{12}x_2+\ldots+a_{1n}x_n \geq b_1,\\
&a_{21}x_1+a_{22}x_2+\ldots+a_{2n}x_n \geq b_2,\\
&\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\\
&a_{m1}x_1+a_{m2}x_2+\ldots+a_{mn}x_n \geq b_m,\\
&x_1 \geq 0;\quad x_2 \geq 0;\quad \ldots;\quad x_n \geq 0.
\end{aligned}
$$

Каноническая форма

- Канонической формой задачи линейного программирования называется задача вида

$$
\begin{aligned}
&c_1x_1 + c_2x_2 + \ldots + c_nx_n \Rightarrow \min \\
&a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n = b_1, \\
&a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n = b_2, \\
&\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\
&a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n = b_m, \\
&x_1 \ge 0; \quad x_2 \ge 0; \quad \ldots; \quad x_n \ge 0.
\end{aligned}
$$

Правила приведения

Рассмотрим теперь те приёмы, которые позволяют произвольные формы задач линейного программирования приводить к указанным выше стандартным формам.

1. Превращение max в min и наоборот.

Если целевая функция в задаче линейного программирования задана в виде

$$c_1x_1 + c_2x_2 + \ldots + c_nx_n \Rightarrow \min$$

то, умножая ее на (- 1), приведем ее к виду

$$(-c_1)x_1 + (-c_2)x_2 + \ldots + (-c_n)x_n \Rightarrow \max$$

так как смена знака приводит к смене min на max.

Аналогично можно заменить max на min.

Правила приведения

2. Смена знака неравенства.

Если ограничение задано в виде

$$a_{i1}x_1 + a_{i2}x_2 + \ldots + a_{in}x_n \le b_i$$

то, умножая на (-1), получим:

$$(-a_{i1})x_1 + (-a_{i2})x_2 + \ldots + (-a_{in})x_n \ge -b_i$$

Аналогично, неравенство вида больше либо равно можно превратить в неравенство вида меньше либо равно .

Правила приведения

3. Превращение равенства в систему неравенств.
Если ограничение задано в виде

$$a_{i1}x_1 + a_{i2}x_2 + \ldots + a_{in}x_n = b_i$$

то его можно заменить эквивалентной системой двух неравенств

$$a_{i1}x_1 + a_{i2}x_2 + \ldots + a_{in}x_n \le b_i$$

$$(-a_{i1})x_1 + (-a_{i2})x_2 + \ldots + (-a_{in})x_n \le -b_i$$

или такой же системой неравенств со знаками больше либо равно.

Указанные выше приемы позволяют приводить задачи линейного программирования к **стандартной форме**.

Правила приведения

4. Превращение неравенств в равенства.

Для приведения задачи к канонической форме, где **все** ограничения имеют вид **равенств**, вводят **дополнительные переменные** $x_{n+1}, x_{n+2}, \ldots, x_{n+p}$, которые тоже считаются неотрицательными и записывают исходную задачу в виде

$$c_1x_1 + c_2x_2 + \ldots + c_nx_n + 0 \cdot x_{n+1} + 0 \cdot x_{n+2} + \ldots + 0 \cdot x_{n+p} \Rightarrow \min$$

$$\begin{cases} a_{r+1,1}x_1 + a_{r+1,2}x_2 + \ldots + a_{r+1,n}x_n - x_{n+r+1} = b_{r+1} \\ \ldots\ldots\ldots\ldots\ldots\ldots \\ a_{p1}x_1 + a_{p2}x_2 + \ldots + a_{pn}x_n - x_{n+p} = b_p \end{cases} \qquad \begin{cases} a_{p+1,1}x_1 + a_{p+1,2}x_2 + \ldots + a_{p+1,n}x_n = b_{p+1} \\ \ldots\ldots\ldots\ldots\ldots\ldots \\ a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n = b_m \end{cases}$$

$$x_1 \geq 0; \quad x_2 \geq 0; \quad \ldots \quad x_n \geq 0; \quad x_{n+1} \geq 0; \quad \ldots \quad x_{n+p} \geq 0$$

Правила приведения

То есть в неравенстве со знаком меньше либо равно **добавляют** дополнительную неотрицательную переменную, а из неравенства со знаком больше либо равно **вычитают** дополнительную переменную.

В целевую функцию эти дополнительные переменные включают с коэффициентом 0, т.е. фактически они в целевой функции отсутствуют.

Получив решение задачи в канонической форме, для получения решения исходной задачи надо просто выбросить из решения значения введенных дополнительных переменных.