

К РЕШЕНИЮ ЛИНЕЙНОЙ ЗАДАЧИ ВЕКТОРНОЙ ОПТИМИЗАЦИИ

1. **Постановка задачи.** Рассмотрим задачу векторной оптимизации

$$L_i(x) = \sum_{j=1}^n c_{ij} x_j \rightarrow \max, \quad i = \overline{1, k};$$

$$\begin{cases} \sum_{j=1}^n a_{ij} x_j = b_i, & i = \overline{1, m}; \\ d_j^1 \leq x_j \leq d_j^2, & j = \overline{1, n} \end{cases} \quad (1)$$

Здесь c_{ij} , $i = \overline{1, k}$, $j = \overline{1, n}$, a_{ij} , $i = \overline{1, m}$, $j = \overline{1, n}$, b_i , $i = \overline{1, m}$ – известные параметры, x_j , $j = \overline{1, n}$ – искомые переменные.

2. **Метод решения.** В [1] описан метод решения задачи (1), суть которого в следующем.

1) Последовательно решаем k обычных (с одним критерием) задач линейного программирования

$$L_r(z) = \sum_{j=1}^n c_{rj} z_j \rightarrow \max;$$

$$\begin{cases} \sum_{j=1}^n a_{ij} z_j = b_i, & i = \overline{1, m}; \\ d_j^1 \leq z_j \leq d_j^2, & j = \overline{1, n} \end{cases} \quad (2)$$

Пусть $z^{*r} = (z_1^{*r}, z_2^{*r}, \dots, z_n^{*r})'$ – решение задачи номер r ,

$L_r^* = L_r(z^{*r})$ – оптимальное значение r -го критерия.

2) Решаем задачу квадратичного программирования

$$R(x) = \sum_{i=1}^k \left[\sum_{j=1}^n \frac{c_{ij}}{L_i^*} x_j - 1 \right]^2 \rightarrow \min;$$

$$\begin{cases} \sum_{j=1}^n a_{ij} x_j = b_i, & i = \overline{1, m}; \\ d_j^1 \leq x_j \leq d_j^2, & j = \overline{1, n} \end{cases} \quad (3)$$

Ее решение $x^* = (x_1^*, x_2^*, \dots, x_n^*)'$ принимаем в качестве векторно-оптимального плана.

3. **Пример.** В качестве наглядного примера рассмотрим задачу типа (1) с двумя переменными и четырьмя критериями:

$$\begin{aligned} L_1(x) &= x_1 + x_2 \rightarrow \min, \\ L_2(x) &= x_1 - x_2 \rightarrow \min, \\ L_3(x) &= -x_1 + x_2 \rightarrow \min, \\ L_4(x) &= -x_1 - x_2 \rightarrow \min, \end{aligned} \quad (4)$$

$$\begin{cases} -\frac{1}{2} x_1 + x_2 \leq 2, \\ 2 x_1 - x_2 \leq 4, \\ 1 \leq x_{1,2} \leq 3. \end{cases}$$

Множество планов задачи (4) изображено на рисунке 2 и представляет собой многоугольник ABCDEF. Благодаря геометрической интерпретации, нетрудно видеть, что решения задач линейного программирования вида (2) с критериями номер 1, 2, 3, 4 находятся в точках D, B, F, A соответственно (для наглядности в этих точках изображены градиенты и линии уровня соответствующих критериев).

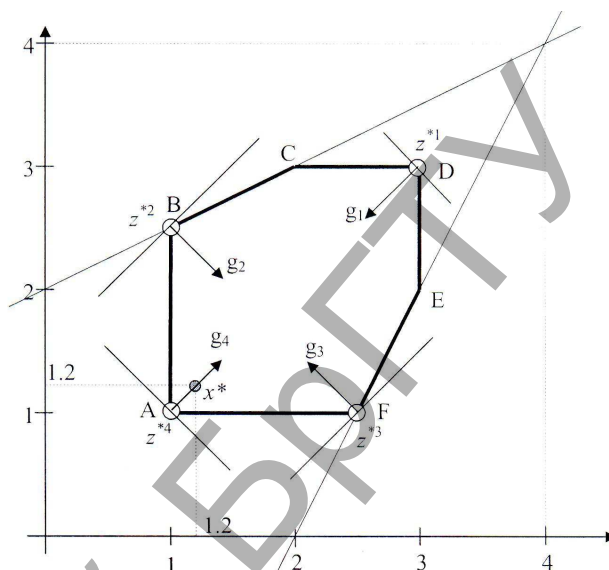


Рис. 1. Геометрическая интерпретация задачи (4)

Таким образом:

$$z^*1 = \begin{pmatrix} 3 \\ 3 \end{pmatrix}, \quad z^*2 = \begin{pmatrix} 1 \\ 2.5 \end{pmatrix}, \quad z^*3 = \begin{pmatrix} 2.5 \\ 1 \end{pmatrix}, \quad z^*4 = \begin{pmatrix} 1 \\ 1 \end{pmatrix},$$

$$L_1^* = -6, \quad L_2^* = -1.5, \quad L_3^* = -1.5, \quad L_4^* = -2.$$

Сформируем целевую функцию задачи (3):

$$\begin{aligned} R(x) &= \left(\frac{1}{6} x_1 + \frac{1}{6} x_2 - 1 \right)^2 + \left(\frac{2}{3} x_1 - \frac{2}{3} x_2 - 1 \right)^2 + \\ &+ \left(-\frac{2}{3} x_1 + \frac{2}{3} x_2 - 1 \right)^2 + \left(\frac{1}{2} x_1 + \frac{1}{2} x_2 - 1 \right)^2 = \\ &= \frac{7}{6} x_1^2 - \frac{11}{9} x_1 x_2 + \frac{7}{6} x_2^2 - \frac{4}{3} x_1 - \frac{4}{3} x_2 + 4 \end{aligned}$$

Следовательно, задача для отыскания векторно-оптимальной точки принимает вид:

$$R(x) = \frac{7}{6} x_1^2 - \frac{11}{9} x_1 x_2 + \frac{7}{6} x_2^2 - \frac{4}{3} x_1 - \frac{4}{3} x_2 + 4 \rightarrow \min,$$

$$\begin{cases} -\frac{1}{2} x_1 + x_2 \leq 2, \\ 2 x_1 - x_2 \leq 4, \\ 1 \leq x_{1,2} \leq 3. \end{cases} \quad (5)$$

По расположению точек z^*1 , z^*2 , z^*3 , z^*4 и способу формирования функции $R(x)$, можно предположить, что векторно-оптимальное решение x^* является внутренней точкой многоугольника ABCDEF. Решение задачи (5) подтверждает это предположение:

$$x^* = \begin{pmatrix} 1.2 \\ 1.2 \end{pmatrix}, \quad R(x^*) = 2.4.$$

4. **Алгоритм.** Понятно, что конкретный алгоритм метода, изложенного в п.2, напрямую зависит от выбора алгоритмов решения задач (2), (3). Поэтому остановимся на этих алгоритмах подробнее.

Для решения серии задач (2) использован алгоритм симплекс-метода [2], в котором

1) для организации вычислений используется обратная базисная матрица;

2) для улучшения базисного плана используется первая неудовлетворяющая условиям оптимальности небазисная компонента.

3) начальный базисный план строится методом искусственного базиса. Реализован специальный алгоритм этого метода, который не требует хранения в матрице основных ограничений дополнительных единичных столбцов.

Особенностью серии задач (2) является то, что в них меняется только целевая функция. Система ограничений остаётся неизменной. Это позволяет при решении каждой следующей задачи использовать в качестве начального базисный оптимальный план, полученный при решении предыдущей. Поэтому метод искусственного базиса, упомянутый в п. 3), используется только при решении первой из серии задач (2).

Для решения задачи квадратичного программирования (3) использован алгоритм прямого метода [3-5]. Для применения этого алгоритма целевая функция должна быть приведена к виду

$$R(x) = \frac{1}{2} x' D x + p' x, \quad (6)$$

где D – $n \times n$ -матрица, p – n -вектор. Получим формулы, связывающие элементы целевой функции (3) с элементами представления (6). Для этого введем в рассмотрение $k \times n$ -матрицу

$$\tilde{C} = \left(\tilde{c}_{ij} = \frac{C_{ij}}{L_i}, i = \overline{1, k}, j = \overline{1, n} \right) \quad (7)$$

и n -вектор

$$\bar{1} = (1, 1, \dots, 1)'. \quad (8)$$

Используя (7), (8), получаем:

$$\begin{aligned} R(x) &= \sum_{i=1}^k \left[\sum_{j=1}^n \frac{C_{ij}}{L_i} x_j - 1 \right]^2 = (\tilde{C}x - \bar{1})' (\tilde{C}x - \bar{1}) = \\ &= x' \tilde{C}' \tilde{C} x - 2 \cdot \bar{1}' \tilde{C} x + \bar{1}' \bar{1} = \\ &= \frac{1}{2} x' (2 \tilde{C}' \tilde{C}) x + (-2 \tilde{C}' \bar{1})' x + n \end{aligned} \quad (9)$$

Сравнивая (6) и (9), получаем:

$$D = 2 \tilde{C}' \tilde{C}, \quad p = -2 \tilde{C}' \bar{1} \quad (10)$$

Покомпонентно соотношения (10) означают:

$$\begin{aligned} d_{ij} &= 2 \sum_{r=1}^k \tilde{c}_{ri} \tilde{c}_{rj}, \quad i = \overline{1, n}, j = \overline{1, n}, \\ p_j &= -2 \sum_{r=1}^k \tilde{c}_{rj}, \quad j = \overline{1, n}. \end{aligned} \quad (11)$$

Соотношения (10), (11) позволяют не использовать матрицу D в алгоритме прямого метода в явном виде, а вычислять её необходимые элементы по мере необходимости.

Одним из фрагментов итерации прямого метода является вычисление градиента целевой функции. В нашем случае

$$\begin{aligned} f &= \text{grad } R(x) = D x + p = \\ &= 2 \tilde{C}' \tilde{C} x - 2 \tilde{C}' \bar{1} = 2 \tilde{C}' (\tilde{C} x - \bar{1}). \end{aligned} \quad (12)$$

В силу (12) градиент целевой функции вычисляется также без непосредственного использования матрицы D по 2-этапной формуле:

$$q = \tilde{C} x - \bar{1}, \quad f = 2 \tilde{C}' q.$$

Как и в симплекс-методе, в прямом методе существует проблема построения начального опорного плана. Поскольку ограничения в задаче (3) такие же, как и у задач (2), базисный оптимальный план, полученный при решении последней из серии задач (2), используется в качестве начального опорного плана при решении задачи (3).

5. Программная реализация. Для решения задач векторной оптимизации (1) описанным в п.4 алгоритмом в системе программирования DELPHI 7 на алгоритмическом языке Object Pascal разработано WINDOWS-приложение *VOpt*. Его основу составляют 2 модуля: *KPSolve* и *VOSolve*. Модуль *KPSolve* реализует алгоритм прямого метода для решения задачи квадратичного программирования (симплекс-метода для решения задачи линейного программирования). Модуль *VOSolve* является внешним по отношению к *KPSolve* и предназначен для:

- формирования последовательности задач линейного программирования (2);
- формирования задачи квадратичного программирования (3);
- вызова соответствующей версии модуля *KPSolve*;
- обработки результатов решения.

6. Численный эксперимент. Численный эксперимент проведен на основе генератора задач вида (1), который входит в структуру приложения *VOpt*. Для проведения эксперимента использован персональный компьютер с процессором AMD Sempron(tm) Processor 2800+ с тактовой частотой 1.61 ГГц и оперативной памятью объемом 512 Мбайт. С помощью генератора формировались серии по 10 задач с одинаковыми размерами. Для каждой серии рассчитывалось среднее время решения. Результаты эксперимента приведены в таблице 1.

Таблица 1. Результаты численного эксперимента

Размеры задачи (1)			Время решения (сек)
k	m	n	
4	5	10	0.0063
12	10	20	0.0437
12	20	50	0.2375
20	40	100	2.836
30	80	200	62.23
40	100	300	298.35

Результаты эксперимента свидетельствуют о возможности использования разработанной программы для решения задач с сотнями переменных, десятками ограничений и критериев.

СПИСОК ЦИТИРОВАННЫХ ИСТОЧНИКОВ

- Иоселиани, А.Н. Методы оптимизации параметров теплообменных аппаратов АЭС / А.Н. Иоселиани, А.А. Михалевич, В.Б. Нестеренко, М.Е. Салуквадзе – Мн.: Наука и техника, 1981. – 144 с.
- Данциг, Д.Б. Линейное программирование, его применения и обобщения. – М.: Прогресс, 1966. – 600 с.
- Габасов, Р. К методам решения общей задачи выпуклого квадратичного программирования / Р. Габасов, Ф.М. Кириллова, В.М. Ракецкий // Докл. АН СССР. – 1981. – Т. 258. – № 6. – С. 1289–1293.
- Ракецкий, В.М. Прямой опорный метод квадратичного программирования. // Проблемы оптимального управления: сб. научн. тр. – Минск: Наука и техника, 1981. – С. 318–335.
- Габасов, Р. Конструктивные методы оптимизации. / Р. Габасов, Ф.М. Кириллова, О.И. Костюкова, В.М. Ракецкий – Мн.: Изд-во "Университетское", 1987. – Ч. 4: Выпуклые задачи – 223 с.

Материал поступил в редакцию 17.02.12

RAKETSKI V.M. To a solution of a linear problem of a multicriteria optimization

The algorithm of a solution of a linear problem of a multicriteria optimization is described.