ЛАБОРАТОРНАЯ РАБОТА № 6
ИССЛЕДОВАНИЕ АЛГОРИТМА РЕШЕНИЯ ЗАДАЧ ПАРАМЕТРИЧЕСКОГО ПРОГРАММИРОВАНИЯ

Не следует смешивать того, что нам кажется невероятным и неестественным, с абсолютно невозможным

Карл Фридрих Гаусс

1. ЦЕЛЬ РАБОТЫ

1. Освоить алгоритмы решения задач параметрического программирования.
2. Приобрести навыки исследования линейных моделей математического программирования при наличии параметрических зависимостей в её компонентах.
3. Научиться оценивать зависимости изменения оптимального решения при вариациях вектора коэффициентов целевой функции или вектора свободных членов системы ограничений.

2. ТЕОРЕТИЧЕСКИЕ ПОЛОЖЕНИЯ

В практической деятельности возникают случаи, когда компоненты модели задачи линейного программирования зависят, в той или иной степени от какого-либо фактора(параметра), например, времени. При этом требуется отыскать оптимальное решение при изменениях (вариациях) этого параметра в некоторых конечных или бесконечных пределах.

Так, для примера, в задачах экономического толка [11], возможны сезонные колебания объёмов продаж (элементы вектора свободных членов системы ограничений) либо величины прибыли / себестоимости (коэффициенты целевой функции).

Хотя зависимость элементов векторов может быть самой разнообразной, в приложениях методов параметрического программирования к задачам линейного программирования чаще всего применяются линейные зависимости указанных элементов от параметров.

Для вектора ограничений

$$b_i^{\nabla} = b_i + tp_i,\ i = \overline{1,m},\ t \in [a,b], \tag{6.1}$$

где b_i — значение элемента i-ого ограничения в отсутствие параметра, t — собственно параметр, p_i — коэффициент параметрического изменения правой части i-ого ограничения; a и b — границы интервала изменения самого параметра.

Для коэффициентов функции цели

$$c_j^{\nabla} = c_j + tp_j,\ j = \overline{1,n},\ t \in [a,b], \tag{6.2}$$

где c_j — значение коэффициента при j-ой переменной целевой функции.

Используя свойство неотрицательности для компонентов вектора оптимальных переменных, в случае (6.1), или условие достижение оптимума, для (6.2), на основании оптимального решения получают условие для уточнения границ [a, b].

Задача параметрического программирования будет считаться решённой, когда интервалы изменения параметров a и b выйдут за пределы заданных в математической модели, то есть $a_0 \to -\infty$, $b_0 \to +\infty$.

$$\begin{aligned} t_0 = a_0 &= \max_{\tilde{p}_i > 0} \left\{ -\frac{\tilde{b}_i}{\tilde{p}_i} \right\}, \quad a), \\ t_0 = b_0 &= \min_{\tilde{p}_i < 0} \left\{ -\frac{\tilde{b}_i}{\tilde{p}_i} \right\}, \quad б), \end{aligned} \tag{6.3}$$

где $\tilde{p}_i$ и $\tilde{b}_i$ — элементы разложения вектора $B' = A_0'$ по базисным векторам оптимального решения.

$$\begin{aligned} t_0 = a_0 &= \max_{\tilde{p}_j > 0} \left\{ -\frac{\tilde{\delta}_j}{\tilde{p}_j} \right\}, \quad a), \\ t_0 = b_0 &= \min_{\tilde{p}_j < 0} \left\{ -\frac{\tilde{\delta}_j}{\tilde{p}_j} \right\}, \quad б), \end{aligned} \tag{6.4}$$

где $\tilde{\delta}_j = \delta_j + t_0 \tilde{p}_j$ — симплекс-разность j-ой переменной в таблице оптимального решения.

Формулы (6.3) применяются при решении ЗЛП с параметрическими зависимостями вектора свободных членов системы ограничений. Формула (6.3, б) будет определять при этом номер строки, соответствующей вектору, выводимому из базиса, после чего последует несколько итераций двойственного симплекс- метода.

Формулы (6.4) используются при параметрических изменениях вектора коэффициентов функции цели, при этом (6.4, б) указывает на направляющий столбец.

Если отрицательные значения $\tilde{p}_i$, $i = 1, m$ или $\tilde{p}_j$ $j=1, n$ отсутствуют, то это означает, что $b_0 \to +\infty$ и достигнут оптимум. Оптимальное значение параметра a_0 устанавливается, обычно, на первой итерации, как только он становится отрицательным.

Параметрические изменения элементов матрицы ограничений не рассматриваются, поскольку:

указанные элементы считаются «технологическими» коэффициентами (расходы сырья, нормы выхода продукта и т.д.), не зависящими от параметров;

параметрический анализ, в этом случае весьма громоздок и вырождается в последовательное решение всех ЗЛП, возникающих при фиксированных значениях параметра.

Подробно ход решения представлен в [5, с. 5 – 23]

3 ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

1. В дополнение к модели ЗЛП, использовавшейся в лабораторных работах № 1 – 5, получить у преподавателя вектор параметрического изменения P коэффициентов целевой функции или вектора свободных членов системы ограничений и диапазон изменения параметра. Типовые варианты приводятся в приложении Б настоящего методического указания.

2. Зафиксировать значение параметра $t = t_0$, решить ЗЛП и получить оптимальное решение для этого случая.

3. Используя оценки (6.3) и/или (6.4) решить задачу параметрического программирования: получить набор интервалов изменения параметра и соответствующие им наборы оптимальных решений.

4. Построить чертёж, на котором представить:

исходную область ограничений;

прямоугольную область (области), в пределах которой будут находиться координаты конца нормали в оптимальных интервалах параметра;

отобразить изменения границ исходной области при изменениях параметра.

5. Оформить отчет, сделать выводы и защитить результаты выполнения лабораторной работы.

4. КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Когда возникают задачи параметрического программирования?

2. Почему границей изменения параметра t при вариации вектора свободных членов служит отношение $\frac{\tilde{b}_i}{\tilde{p}_i}$?

3. Почему границей изменения параметра t при вариации коэффициентов функции цели служит отношение $\frac{\tilde{\delta}_j}{\tilde{p}_j}$?

4. Что означает физически или геометрически параметрические изменения элементов вектора свободных членов системы ограничений?

5. Что означает физически или геометрически параметрические изменения коэффициента целевой функции?

6. Всегда ли разрешима задача параметрического программирования?

7. Почему задача параметрического изменения элементов матрицы системы ограничений рассматривается и решается весьма редко?

8. В чем сущность и особенности метода решения задачи параметрического программирования по отношению к классической ЗЛП?

9. Когда задача параметрического программирования не будет иметь решения?

10. Почему на каждом шаге работы алгоритма возникают сразу две границы изменения параметра (верхнее b и нижнее a)

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Деордица Ю. Ф. Исследование операций в планировании управления / Ю. Ф. Деордица, Ю. М., Нефедов. – Киев : Вища школа, 1991. – 196 с.
2. Зайченко Ю. П. Исследование операций : учебное пособие / Ю. П. Зайченко. – Киев : Вища школа, 1979. – 392 с.
3. Зайченко Ю. П. Исследование операций: сборник задач / Ю. П. Зайченко, С. А. Шумилова. – Киев : Вища школа, 1990. – 239 с.
4. Карлусов В. Ю. Исследование операций и методы оптимизации : учебное пособие / В. Ю. Карлусов ; Севастопольский государственный университет. – Севастополь : СевГУ, 2018. – 315 с.
5. Методическое пособие к решению задач линейного программирования по дисциплине «Методы исследования операций» для студентов направлений подготовки 09.03.02 – «Информационные системы и технологии» и 09.03.03 – «Прикладная информатика» всех форм

обучения / Севастопольский государственный университет ; сост.: В. Ю. Карлусов, Е. Н. Заикина. – Севастополь : СевГУ, 2021. – 59 с.

6. Методическое пособие к выполнению лабораторно - вычислительного практикума по дисциплине «Методы исследования операций». Часть 3: «Параметрическое программирование», «Квадратичное программирование», «Линейное целочисленное программирование» для студентов профилей 09.03.02 – «Информационные системы и технологии» и 09.03.03 – «Прикладная информатика» всех форм обучения / Севастопольский государственный университет ; сост.: Е. Н. Заикина, В. Ю. Карлусов – Севастополь : СевГУ, 2016. – 46 с.

ЭЛЕКТРОННЫЕ ИЗДАНИЯ, ДОСТУПНЫЕ ПО ПОДПИСКЕ СЕВГУ

7. Горлач, Б. А. Исследование операций [Электронный ресурс]: учебное пособие / Б. А. Горлач. — Электрон. дан. — Санкт-Петербург: Лань, 2013. — 448 с. — Режим доступа: https://e.lanbook.com/book/4865. — Загл. с экрана.
8. Ржевский, С. В. Исследование операций [Электронный ресурс] : учебное пособие / С. В. Ржевский. — Электрон. дан. — Санкт-Петербург: Лань, 2013. — 480 с. — Режим доступа: https://e.lanbook.com/book/32821. — Загл. с экрана.
9. Есипов, Б. А. Методы исследования операций [Электронный ресурс] : учебное пособие / Б. А. Есипов. — Электрон. дан. — Санкт-Петербург : Лань, 2013. — 304 с. — Режим доступа: https://e.lanbook.com/book/68467. — Загл. с экрана.
10. Акулич, И. Л. Математическое программирование в примерах и задачах [Электронный ресурс] : учебное пособие / И. Л. Акулич. — Электрон. дан. — Санкт-Петербург : Лань, 2011. — 352 с. — Режим доступа: https://e.lanbook.com/book/2027. — Загл. с экрана.
11. Балдин К. В. Математическое программирование / Балдин К. В., Брызгалов Н. А., Рукосуев А. В., — 2-е изд. — М.:Дашков и К, 2018. — 218 с. — Режим доступа : http://znanium.com/catalog/product/415097. — ISBN 978-5-394-01457-4

ВАРИАНТЫ К ВЫПОЛНЕНИЮ ЛАБОРАТОРНЫХ РАБОТ ПО ИССЛЕДОВАНИЮ ЗАДАЧ ПАРАМЕТРИЧЕСКОГО ПРОГРАММИРОВАНИЯ

Функции цели

0. $f(x_1, x_2) = (1 + 1\ t)\cdot x_1 + (2 - 2\ t)\cdot x_2 \to \max;\ t \in [-1, 1]$.
1. $f(x_1, x_2) = (2 + 3\ t)\cdot x_1 + (1\cdot + 3\ t)\ x_2 \to \max;\ t \in [-1, 1]$.
2. $f(x_1, x_2) = (2 + 4\ t)\cdot x_1 + (4 - 3\ t)\cdot x_2 \to \max;\ t \in [-1, 1]$.

3. $f(x_1, x_2) = (3 + 2\ t)\cdot x_1 + (6 - 5\ t)\cdot x_2 \to \max;\ t \in [-1, 1]$.
4. $f(x_1, x_2) = (7 - 6\ t)\cdot x_1 + (3 + 2\ t)\cdot x_2 \to \max;\ t \in [-1, 1]$.
5. $f(x_1, x_2) = (3 - 2\ t)\cdot x_1 + (8 - 5\ t)\cdot x_2 \to \max;\ t \in [-1, 1]$.
6. $f(x_1, x_2) = (4 - 3\ t)\cdot x_1 + (3 + 2\ t)\cdot x_2 \to \max;\ t \in [-1, 1]$.
7. $f(x_1, x_2) = (5 - 2\ t)\cdot x_1 + (5 - 3\ t)\cdot x_2 \to \max;\ t \in [-1, 1]$.
8. $f(x_1, x_2) = (4 + 2\ t)\cdot x_1 + (7 - 5\ t)\cdot x_2 \to \max;\ t \in [-1, 1]$.
9. $f(x_1, x_2) = (2 + 6\ t)\cdot x_1 + (5 - 3\ t)\cdot x_2 \to \max;\ t \in [-1, 1]$.

Правые части систем ограничений

$t \in [-1, 1]$

```
for (i = 1; i <= 3; i++)
{
    if (b[i] <= 10) a = 1;
        else if (b[i] <= 20) a = 5;
            else if (b[i] <= 30) a = 10;
                else a = 15;
    }
b[i] += a * t;
}
```