**Лабораторная работа № 3**

**Решение задач линейного программирования**

1. Цель работы

Целью работы является исследование метода решения задач линейного программирования, построив алгоритм и отладив программу реализации для применения функции *linprog* математического пакета *MatLab*.

1. Методические указания

Рассматривается задача линейного программирования общего вида:



Здесь *х* – вектор длины *n*, с – вектор линейной формы, *А*1 – (*m* x *n*)-матрица неравенств, *b*1 – их вектор свободных членов, *А*2 – (*k* x *n*)-матрица равенств, причем *k* < *n,* *b*2 –вектор свободных членов равенств, *xm*, *xM* – минимальное и максимальное значения вектора *х* соответственно. Длину этого вектора будем называть порядком задачи линейного программирования.

Смысл задачи линейного программирования состоит в поиске вектора *хо*, который является решением линейной системы уравнений, определяемых парой (*А*2, *b*2), удовлетворяет системе линейных неравенств, определяемых парой (*А*1, *b*1), и находящемся в заданном интервале [*xm*, *xM*].

Пример 1. Простейшая задача линейного программирования.



Допустимая область значений – треугольник с множеством трех вершин

{(x1, x2)} ={(0,0), (0,1),(1,0)}.

Известно, что решение задачи линейного программирования находится в этих вершинах. Поиск решения осуществим перебором

J(0,0) = 0, J(0,1) = 3, J(1,0) = 2.

Следовательно, искомое решение имеет вид



Если критерий стремится не к минимуму, а к максимуму, то задача линейного программирования будет иметь вид



Поиск такого решения осуществляется аналогично предыдущему поиску

J(0,0) = 0, J(0,1) = -3, J(1,0) =- 2.

Следовательно, искомое решение новой задачи имеет вид



Пример 2. Усложним задачу линейного программирования



Допустимая область значений – трапеция с множеством четырех вершин

{(x1, x2)} ={(1,0), (0,1),(0.5,0),(0,0.5)}.

Известно, что решение задачи линейного программирования находится в этих вершинах. Поиск решения осуществим перебором

J(1,0) = 2, J(0,1) = 3, J(0.5,0) = 1, J(0,0.5) = 1.5

Следовательно, искомое решение имеет вид



Если критерий стремится не к минимуму, а к максимуму, то задача линейного программирования будет иметь вид



Поиск такого решения осуществляется аналогично предыдущему поиску и приводит к следующему результату



Для решения задачи линейного программирования общего вида используется m-функция из *Optimization Toolbox* математического пакета *MatLab*. Синтаксис этой функции имеет вид:

[*x,C*]=*linprog*(*c*, *A*1, *b*1, *A*2, *b*2, *xm*, *xM*)

Передаваемые в функцию переменные определяются задачей линейного программирования, а возвращаемые переменные содержат искомое решение задачи *х*=*хо* и минимальное значение линейной формы *С*.

При решении частной задачи 1 линейного программирования вида:



синтаксис функции *linprog* имеет вид:

[*x,C*]=*linprog*(*c*, *A*1, *b*1, [ ], [ ], *zeros(n,1)*)

При решении частной задачи 2 линейного программирования вида:



синтаксис функции *linprog* имеет вид:

[*x,C*]=*linprog*(*c*, [ ], [ ], *A*2, *b*2, xm, xM)

Рассмотренную раньше в Примере 1 задачу следует считать тестом с исходными данными n =2, m=1, k=0. Задачу из Примера 2 следует считать тестом с исходными данными n =2, m=1, k=1.

Примеры решения указанных задач линейного программирования с помощью функции *linprog.*

Пример 3. Задача линейного программирования общего вида имеет порядок *n*=4, число неравенств *m*=6, число равенств *k*=2, параметры задачи:



Результаты решения задачи:



Пример 4. Частная задача 1 линейного программирования имеет порядок *n*=4, число неравенств *m*=6, параметры задачи:



Результаты решения задачи:



Пример 3. Частная задача 2 линейного программирования имеет порядок *n*=4, число число равенств *k*=2, параметры задачи:



Результаты решения задачи:



Решение задач осуществлялось с помощью функции *linprog*.

1. Порядок выполнения работы
   1. С помощью шаблона сценария *sc.m* написать сценарий *sc5.m* исследования функции *linprog.*
   2. Предусмотреть исследование задачи линейного программирования общего вида, а также частных задач 1,2.
   3. При исследовании функции *linprog* задавать различные значения параметров рассматриваемых задач.
   4. Используя сценарий-платформу *scр.m,* выполнить исследование функции *linprog.*
   5. Зафиксировать в Отчете вид исследуемых задач и полученные результаты.
2. Варианты заданий

|  |  |  |  |
| --- | --- | --- | --- |
| Вариант | Порядок задачи линейного программирования | Число неравенств | Число равенств |
| 1 | 4 | 5 | 3 |
| 2 | 6 | 7 | 4 |
| 3 | 5 | 6 | 2 |
| 4 | 6 | 7 | 5 |
| 5 | 7 | 8 | 4 |
| 6 | 4 | 6 | 2 |
| 7 | 6 | 8 | 4 |
| 8 | 7 | 9 | 3 |
| 9 | 6 | 7 | 3 |
| 10 | 4 | 5 | 2 |