|  |  |
| --- | --- |
| **Gerb-BMSTU_01** | **Министерство науки и высшего образования Российской Федерации**  Калужский филиал  федерального государственного бюджетного  образовательного учреждения высшего образования  ***«Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)»***  ***(КФ МГТУ им. Н.Э. Баумана)*** |

|  |  |
| --- | --- |
| **ФАКУЛЬТЕТ** | **ИУК «Информатика и управление»** |
| **КАФЕДРА** | **ИУК4 «Программное обеспечение ЭВМ,** |
| **информационные технологии»** | |

**Лабораторная работа №2**

**«Графический метод решения задачи математического программирования»**

**ДИСЦИПЛИНА: «Моделирование»**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Выполнил: студент гр. ИУК4-72Б | |  |  | ( | Сафронов Н.С. | ) |
|  |  |  | (подпись) |  | (Ф.И.О.) |  |
| Проверил: | |  |  | ( | Никитенко У.В. | ) |
|  |  |  | (подпись) |  | (Ф.И.О.) |  |

|  |  |
| --- | --- |
| Дата сдачи (защиты):  Результаты сдачи (защиты): | |
|  | - Балльная оценка:  - Оценка: |

Калуга, 2023

**Цель работы:** изучение математического аппарата математического программирования на примере задач небольшой размерности, допускающих графическое решение.

**Постановка задачи**

**Вариант 14**

Для функционирования завода необходимо пополнять его склад расходными материалами. Ежедневно на склад должно быть доставлено не менее 9 ед. расходного материала №1, 8 ед. расходного материала №2 и 11 ед. расходного материала №3. Для пополнения склада были заключены договоры с двумя автопредприятиями. Их возможности представлены в таблице:

|  |  |  |
| --- | --- | --- |
| Расходные материалы | Количество доставленных материалов | |
| Предприятие №1 | Предприятие №2 |
| Расходный материал №1 | 3 | 1 |
| Расходный материал №2 | 1 | 2 |
| Расходный материал №3 | 1 | 6 |

Стоимость перевозки по договору с Предприятием №1 - 4 д.е., с Предприятием №2 - 6 д.е. Составьте план перевозок, имеющий минимальную стоимость.

С помощью графического анализа чувствительности определите, как изменится значение целевой функции при изменении минимального уровня перевозок Предприятием №1.

**Результаты выполнения работы**

Составим математическую модель.

Введём переменные:

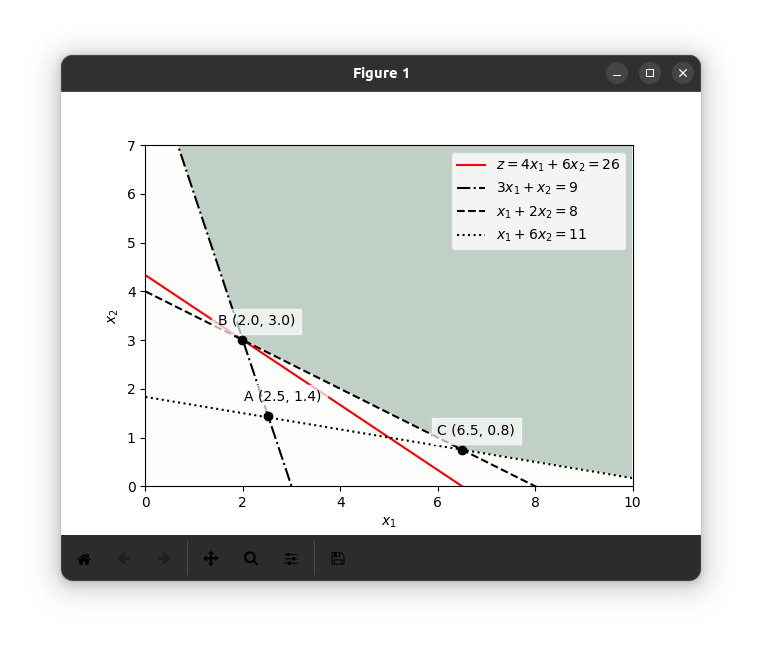
– уровень перевозок предприятием №1,

– уровень перевозок предприятием №2,

– стоимость плана перевозок.

Получаем следующую систему уравнений:

Найдём оптимальный план перевозок:



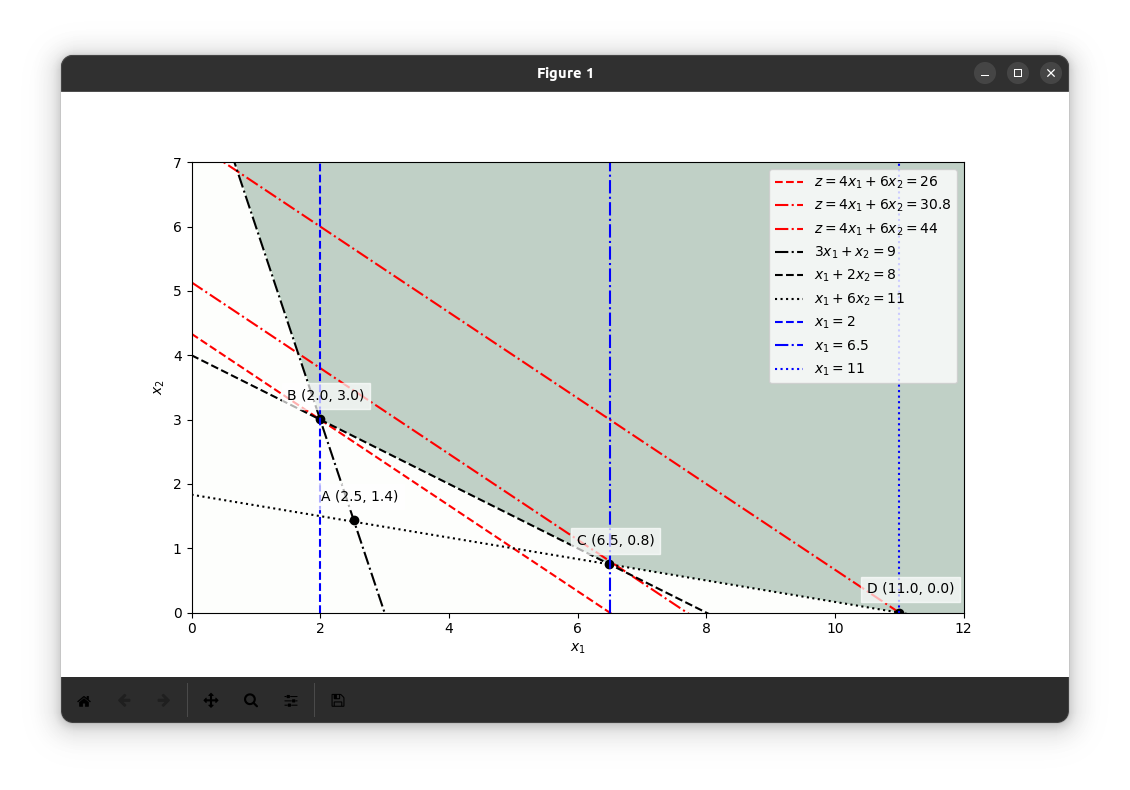
**Рисунок 1 –** Найденный оптимальный план перевозок

Получаем, оптимальную точку , где – план перевозок, имеющий минимальную стоимость – 26 д.е.

Увеличим минимальный уровень перевозок Предприятия №1 (см. рис 2). При его изменении, начиная с , точка будет передвигаться вдоль отрезка , а затем, начиная с , вдоль .

Вычислим стоимость перевозки Предприятием №1 на отрезке :

Вычислим стоимость перевозки Предприятием №1 на отрезке :



**Рисунок 2 –** Изменение плана при увеличении минимального уровня перевозок предприятия №1

**Вывод:** в ходе выполнения лабораторной работы был изучен математический аппарат математического программирования на примере задач небольшой размерности, допускающих графическое решение.

**ПРИЛОЖЕНИЯ**

**Листинг программы**

# Вариант 14  
from functools import reduce  
  
import matplotlib.lines  
import matplotlib.pyplot as plt  
import numpy as np  
  
if \_\_name\_\_ == "\_\_main\_\_":  
  
 conditions = [  
 lambda x, y: 3 \* x + y >= 9,  
 lambda x, y: x + 2 \* y >= 8,  
 lambda x, y: x + 6 \* y >= 11  
 ]  
  
 equalities = [  
 lambda x, y: 3 \* x + y - 9,  
 lambda x, y: x + 2 \* y - 8,  
 lambda x, y: x + 6 \* y - 11  
 ]  
  
 explicit\_equalities = [  
 lambda x: 9 - 3 \* x,  
 lambda x: 4 - x / 2,  
 lambda x: (11 - x) / 6,  
 ]  
  
 labels = [  
 '$3x\_1 + x\_2 = 9$',  
 '$x\_1 + 2x\_2 = 8$',  
 '$x\_1 + 6x\_2 = 11$'  
 ]  
  
 colors = [  
 "k-.",  
 "k--",  
 "k:"  
 ]  
  
 figure, axis = plt.subplots()  
  
 x = np.arange(0, 12, 0.01)  
  
 plan = (26 - 4 \* x) / 6  
 plt.plot(x, plan, "r--", label=f'$z = 4x\_1 + 6x\_2 = 26$')  
 plan = (30.8 - 4 \* x) / 6  
 plt.plot(x, plan, "r-.", label=f'$z = 4x\_1 + 6x\_2 = 30.8$')  
 plan = (44 - 4 \* x) / 6  
 plt.plot(x, plan, "r-.", label=f'$z = 4x\_1 + 6x\_2 = 44$')  
  
 point\_name = ord('A')  
 for i in range(len(explicit\_equalities)):  
 previous\_i = i - 1 if i != 0 else len(explicit\_equalities) - 1  
 previous\_f = explicit\_equalities[previous\_i](x)  
 f = explicit\_equalities[i](x)  
 idx = np.argwhere(np.diff(np.sign(previous\_f - f))).flatten()[0]  
 plt.plot(x, f, colors[i], label=labels[i])  
 plt.plot(x[idx], f[idx], 'ko')  
 axis.annotate(  
 f'{chr(point\_name)} ({x[idx]:.1f}, {f[idx]:.1f})',  
 (x[idx] - 0.5, f[idx] + 0.3),  
 backgroundcolor='#ffffffB0'  
 )  
 point\_name += 1  
  
 for i in range(len(explicit\_equalities)):  
 if i != 2:  
 continue  
 f = explicit\_equalities[i](x)  
 idx = np.argwhere(np.diff(np.sign(f))).flatten()[0]  
 plt.plot(x[idx], f[idx], 'ko')  
 axis.annotate(  
 f'{chr(point\_name)} ({x[idx]:.1f}, {f[idx]:.1f})',  
 (x[idx] - 0.5, f[idx] + 0.3),  
 backgroundcolor='#ffffffB0'  
 )  
 point\_name += 1  
  
 plt.axvline(2, color='b', linestyle='--', label='$x\_1=2$')  
 plt.axvline(6.5, color='b', linestyle='-.', label='$x\_1=6.5$')  
 plt.axvline(11, color='b', linestyle=':', label='$x\_1=11$')  
 axis.set\_ylim(0, 7)  
 axis.set\_xlim(0, 12)  
  
 plt.xlabel("$x\_1$")  
 plt.ylabel("$x\_2$")  
  
 xs, ys = np.meshgrid(x, x)  
 regions = [condition(xs, ys) for condition in conditions]  
 intersection = np.array(reduce(lambda \_x, \_y: \_x & \_y, regions))  
 extent = (x.min(), x.max(), x.min(), x.max())  
 plt.imshow(  
 intersection.astype(int),  
 extent=extent,  
 origin="lower",  
 cmap="Greens",  
 alpha=0.25  
 )  
  
 plt.legend()  
 plt.show()