Федеральное государственное образовательное бюджетное

учреждение высшего образования

**«Финансовый университет при Правительстве Российской Федерации»**

**(Финансовый университет)**

Колледж информатики и программирования

**ОТЧЁТ**

**По лабораторной работе №4**

Студент: Горланов Владимир Владимирович

Дисциплина/Профессиональный модуль: Математическое моделирование

Группы: 4ИСИП-619

Преподаватель

Сибирев И.В.

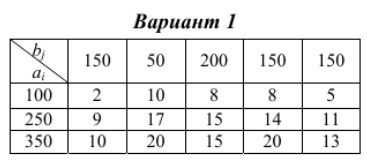
Оценка за работу :\_\_\_\_\_\_\_

**Москва – 2023 г.**

**Тема:** «Транспортная задача».

**Цель работы:** приобретение навыков решения транспортной задачи с составлением первоначального плана распределения поставок различными методами.

**Ход работы:**

Некоторый однородный продукт, сосредоточенный у трех поставщиков А1, А2, А3 в количестве а1, а2, а3 т соответственно, необходимо доставить потребителям В1, В2, В3, В4, В5 в количестве b1, b2, b3, b4, b5 т. Стоимость Cij перевозки тонны груза от i-го поставщика j-му потребителю задана матрицей D. Составить план перевозок, имеющий минимальную стоимость и позволяющий вывести все грузы и полностью удовлетворить потребности. 

Эта задача может быть сформулирована как задача линейного программирования. Нам нужно найти план перевозок, который минимизирует общую стоимость перевозки груза, при условии, что каждый из потребителей получает требуемое количество груза, а каждый из поставщиков поставляет только свое доступное количество груза.

Для решения этой задачи можно использовать библиотеку PuLP в Python.

Вот такой код:

(Перед этим надо отдельно выполнить pip install pulp)

import pulp

import numpy as np

# Матрица

D = np.array([

[2, 10, 8, 8, 5],

[9, 17, 15, 14, 11],

[10, 20, 15, 20, 13]

])

a = [100, 250, 350]

b = [150, 50, 200, 150, 150]

# Создаем новую задачу линейного программирования

prob = pulp.LpProblem("Transportation Problem", pulp.LpMinimize)

# Создаем переменные решения для каждого возможного маршрута

routes = [(i, j) for i in range(3) for j in range(5)]

vars\_dict = pulp.LpVariable.dicts("Route", routes, lowBound=0, cat=pulp.LpInteger)

# Определяем целевую функцию, минимизирующую стоимость перевозки груза

prob += pulp.lpSum([vars\_dict[i, j]\*D[i][j] for (i, j) in routes])

# Добавляем ограничения, гарантирующие, что каждый потребитель получает требуемое количество груза

for j in range(5):

prob += pulp.lpSum([vars\_dict[i, j] for i in range(3)]) == b[j]

# Добавляем ограничения, гарантирующие, что каждый поставщик поставляет только свое доступное количество груза

for i in range(3):

prob += pulp.lpSum([vars\_dict[i, j] for j in range(5)]) <= a[i]

# Решаем задачу линейного программирования

prob.solve()

# Выводим результаты

print("Total Cost = ", pulp.value(prob.objective))

for v in prob.variables():

print(v.name, "=", v.varValue)

Ответ:

Total Cost = 8450.0

Route\_(0,\_0) = 0.0

Route\_(0,\_1) = 50.0

Route\_(0,\_2) = 0.0

Route\_(0,\_3) = 0.0

Route\_(0,\_4) = 50.0

Route\_(1,\_0) = 0.0

Route\_(1,\_1) = 0.0

Route\_(1,\_2) = 0.0

Route\_(1,\_3) = 150.0

Route\_(1,\_4) = 100.0

Route\_(2,\_0) = 150.0

Route\_(2,\_1) = 0.0

Route\_(2,\_2) = 200.0

Route\_(2,\_3) = 0.0

Route\_(2,\_4) = 0.0