ГУАП

КАФЕДРА № 43

ОТЧЕТ
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

Профессор		С.И. Колесникова
должность, уч. степень, звание	подпись, дата	инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №1

Нелинейное программирование. Вариационный принцип.

по дисциплине: Компьютерное моделирование

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР.	4134к		Столяров Н.С.
		подпись, дата	инициалы, фамилия

Санкт-Петербург
2024

Цель работы.

Цель настоящей работы – освоить средства моделирования задач линейного программирования. Решение простейшей вариационной задачи

Постановка задачи

Вариант 17

При откорме животных каждое животное ежедневно должно получить не менее 60 ед. питательного вещества А, не менее 50 ед. вещества В и не менее 12 ед. вещества С. Указанные питательные вещества содержат три вида корма. Содержание единиц питательных веществ в 1 кг каждого из видов корма приведено в следующей таблице:

Питательные вещества	Количество единиц питательных веществ на 1 кг корма вида		
	I	II	III
A	1	3	4
B	2	4	2
C	1	4	3

Составить дневной рацион, обеспечивающий получение необходимого количества питательных веществ при минимальных денежных затратах, если цена 1 кг корма I вида составляет 9 коп., корма II вида – 12 коп. и корма III вида – 10 коп.

Формализованная постановка задачи

Обозначим переменные (xi) как количество килограммов корма iii-го вида, которое необходимо использовать в дневном рационе.

1. x1: количество килограммов корма I вида.
2. x2: количество килограммов корма II вида.
3. x3: количество килограммов корма III вида.

Целевая функция

Цель — минимизировать общую стоимость кормов, которая выражается как:

Minimize $Z=9x1+12x2+10x3$

где

- 9 — цена 1 кг корма I вида,
- 12 — цена 1 кг корма II вида,
- 10 — цена 1 кг корма III вида.

Ограничения

Рацион должен удовлетворять минимальным требованиям к количеству питательных веществ A,B,CA, B, CA,B,C:

1. Ограничение по веществу A: $x1+3x2+4x3 \geq 60$
2. Ограничение по веществу B: $2x1+4x2+2x3 \geq 50$
3. Ограничение по веществу C: $x1+4x2+3x3 \geq 12$
4. Ограничения на неотрицательность: $x1 \geq 0, x2 \geq 0, x3 \geq 0$

Итоговая формулировка задачи

Мы должны **минимизировать** функцию стоимости: $Z=9x1+12x2+10x3$
при выполнении ограничений:

- $x1+3x2+4x3 \geq 60$.
- $2x1+4x2+2x3 \geq 50$.
- $1x1+4x2+3x3 \geq 12$.
- $x1 \geq 0, x2 \geq 0, x3 \geq 0$.

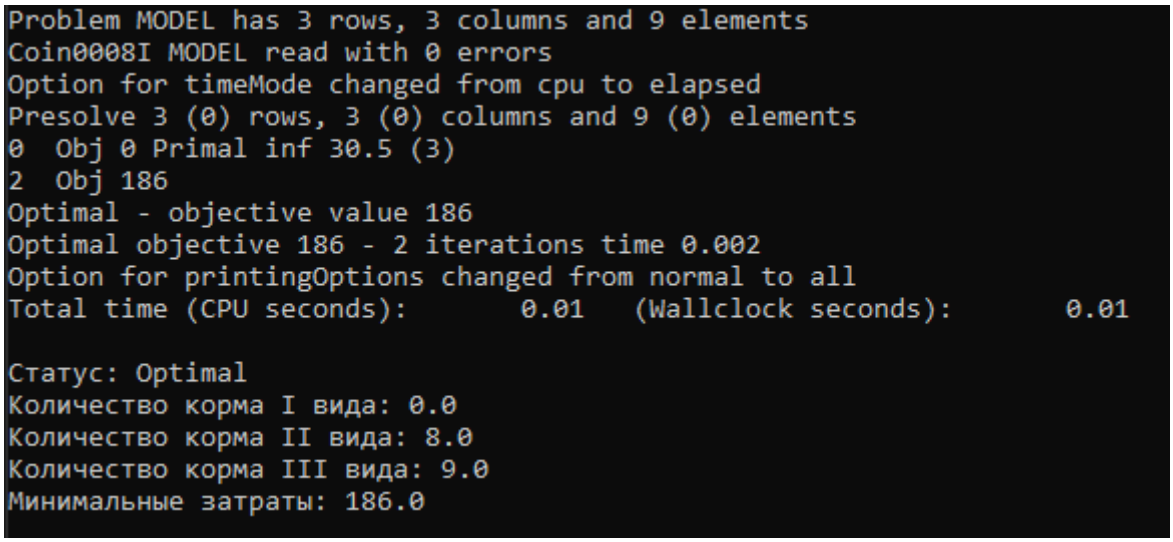
Скриншоты решения Excell

Я использовал LibreOffice Calc так как писал работу из под дистрибутива на linux

	A	B	C	D	E	F	G	H	I	J	K
1	**Корм**	**Цена (руб)**	**A**	**B**	**C**	**Количество**	**Суммарная стоимость**		**Питательные вещества**	**Расчетное значение**	**Минимум**
2	I	9	1	2	1	0	0		A	60	60
3	II	12	3	4	4	8	96		B	50	50
4	III	10	4	2	3	9	90		C	59	12
5	**Итого**						186				

Рисунок 1 – Результаты решения

Скриншоты работы программы на python

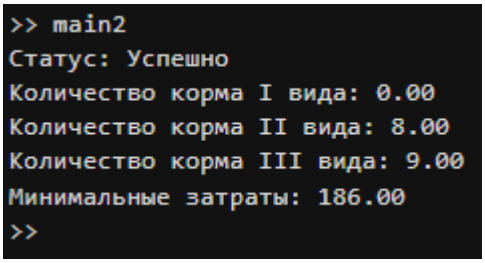


Листинг программы

```
from pulp import LpProblem, LpMinimize, LpVariable, lpSum, LpStatus, value


# Содержание питательных веществ в 1 кг каждого вида корма
feed_nutrients = {
    "I": [1, 2, 1],   # [A, B, C]
    "II": [3, 4, 4],
    "III": [4, 2, 3]
}


# Цены на 1 кг каждого вида корма
feed_prices = {
    "I": 9,
    "II": 12,
    "III": 10
}


# Минимальные требования к питательным веществам
min_requirements = {
    "A": 60,
    "B": 50,
```

```
   "C": 12
}


# Создание задачи
problem = LpProblem("Animal_Feed_Optimization", LpMinimize)


# Переменные
x1 = LpVariable("Feed_Type_I", lowBound=0)
x2 = LpVariable("Feed_Type_II", lowBound=0)
x3 = LpVariable("Feed_Type_III", lowBound=0)


# Целевая функция
problem += feed_prices["I"] * x1 + feed_prices["II"] * x2 + feed_prices["III"] * x3


# Ограничения
problem += feed_nutrients["I"][0] * x1 + feed_nutrients["II"][0] * x2 + feed_nutrients["III"][0] * x3 >= min_requirements["A"]  # A
problem += feed_nutrients["I"][1] * x1 + feed_nutrients["II"][1] * x2 + feed_nutrients["III"][1] * x3 >= min_requirements["B"]  # B
problem += feed_nutrients["I"][2] * x1 + feed_nutrients["II"][2] * x2 + feed_nutrients["III"][2] * x3 >= min_requirements["C"]  # C


# Решение задачи
problem.solve()


# Результаты
print("Статус:", LpStatus[problem.status])
print("Количество корма I вида:", value(x1))
print("Количество корма II вида:", value(x2))
print("Количество корма III вида:", value(x3))
print("Минимальные затраты:", value(problem.objective))
```

Скриншоты работы программы на matlab



Листинг программы

```
% Содержание питательных веществ в 1 кг каждого вида корма

feed_nutrients = [
1, 3, 4; % A
2, 4, 2; % B
1, 4, 3 % C
];
% Цены на 1 кг каждого вида корма
feed_prices = [9; 12; 10];
% Минимальные требования к питательным веществам
min_requirements = [60; 50; 12];
% Определение коэффициентов целевой функции
f = feed_prices;
% Определение матрицы ограничений
A = -feed_nutrients; % Умножаем на -1 для преобразования в стандартный вид
b = -min_requirements;
% Ограничения на неотрицательность
lb = [0; 0; 0]; % Нижние границы для переменных
% Решение задачи линейного программирования
options = optimoptions('linprog', 'Display', 'off');
[x, fval, exitflag] = linprog(f, A, b, [], [], lb, [], options);
% Проверка статуса решения
if exitflag == 1
fprintf('Статус: Успешно\n');
fprintf('Количество корма I вида: %.2f\n', x(1));
fprintf('Количество корма II вида: %.2f\n', x(2));
fprintf('Количество корма III вида: %.2f\n', x(3));
fprintf('Минимальные затраты: %.2f\n', fval);
else
fprintf('Статус: Не удалось найти решение\n');
end
```

Часть 2 – решение вариационной задачи

Вариант 17

17	$V[y(x)]=\int_1^4\left(\sqrt{x}y'^2(x)+\frac{y^2(x)}{2x\sqrt{x}}\right)dx, \quad y(1)=2, \quad y(4)=4\frac{1}{2};$

Скриншот решения на python

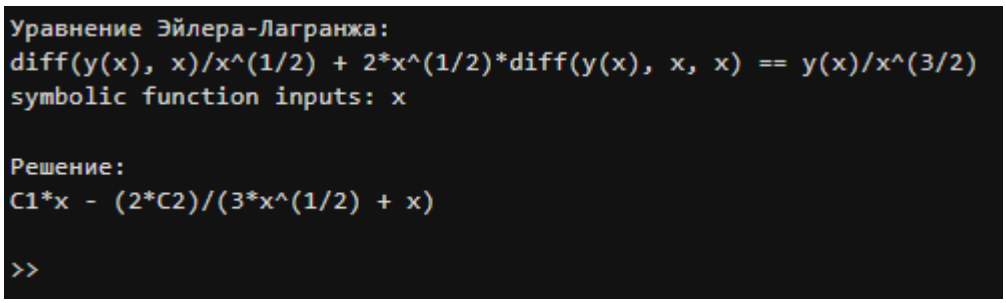
```
G:\PROJECTS\GUAP\Programming-GUAP\КомпМод\1>python main6.py
Уравнение Эйлера-Лагранжа:
Eq((-2*x**2*Derivative(y(x), (x, 2)) - x*Derivative(y(x), x) + y(x))/x**(3/2), 0)
Решение:
Eq(y(x), C1/sqrt(x) + C2*x)

G:\PROJECTS\GUAP\Programming-GUAP\КомпМод\1>
```

Листинг программы

```
from sympy import symbols, Function, sqrt, diff, Eq, simplify, dsolve, solve


# Определение переменных

x = symbols('x')

y = Function('y')(x)


# Первая производная y по x

dy = diff(y, x)


# Определение лагранжиана

F = sqrt(x) * dy**2 + (y**2) / (2 * x * sqrt(x))


# Частные производные
```

```
dFdy = diff(F, y)        # Производная по y

dFd1y = diff(F, dy)      # Производная по y'


# Производная по x от dF/dy'

d_dFd1y_dx = diff(dFd1y, x)


# Уравнение Эйлера-Лагранжа

L = dFdy - d_dFd1y_dx     # Левое выражение уравнения Эйлера-Лагранжа


# Приведение уравнения к стандартному виду

eqn = Eq(simplify(L), 0)


print("Уравнение Эйлера-Лагранжа:")

print(eqn)


# Решение уравнения

sol = dsolve(eqn)

print("Решение:")

print(sol)


eq1 = sol.subs({x:2, y:1})

eq2 = sol.subs({x:4 * (1/2), y:4})

coeffs = solve([eq1, eq2])

res = sol.subs(coeffs)

res.doit()
```

Скриншот решения на matlab



Листинг программы

```
% Очистка рабочей области и консоли

clear all
clc
% Объявляем переменные
syms x y(x)
% Первая производная y по x
dy = diff(y, x);
% Определение лагранжиана
F = sqrt(x) * dy^2 + (y^2) / (2 * x * sqrt(x));
% Частные производные
dFdy = diff(F, y); % Производная по y
dFd1y = diff(F, dy); % Производная по y'
% Производная по x от dF/dy'
d_dFd1y_dx = diff(dFd1y, x);
% Уравнение Эйлера-Лагранжа
L = dFdy - d_dFd1y_dx; % Левое выражение уравнения Эйлера-Лагранжа
% Приведение уравнения к стандартному виду
eqn = simplify(L == 0);
disp('Уравнение Эйлера-Лагранжа:');
disp(eqn);
% Решение уравнения
% sol = dsolve(eqn, y);
% disp('Решение:');
% disp(sol);
% Определяем переменные
syms y(x)
% Уравнение
eqn = diff(y, x) / sqrt(x) + 2 * sqrt(x) * diff(y, x, x) == y / x^(3/2);
% disp('Уравнение Эйлера-Лагранжа:');
% disp(eqn);
% Решение уравнения
sol = dsolve(eqn);
disp('Решение:');
disp(sol);
```