Министерство науки и образования РФ

Федеральное государственное бюджетное учреждение

высшего образования

**«Тверской государственный технический университет»**

(ТвГТУ)

Кафедра программного обеспечения

**Отчет по лабораторной работе №3**

по дисциплине: «Исследование операций»

Тема: «Многокритериальная оптимизация.

Решение задачи методами целевого программирования»

|  |
| --- |
| Выполнил:  студент группы  ПИН.РИС – 17-06  Великов Д.А. |
| Проверила:  ассистент кафедры ПО  Корнеева Е.И. |

Тверь 2020

**Постановка задачи**

1. Изучить ход решения задачи из примера 8.2.1. (С. 387 - 388 в учебнике Таха Х.А.) методом весовых коэффициентов;

2. Реализовать задачу на языке Python с использованием фреймворка pyQt5 для реализации интерфейса задачи;

3. Изучить ход решения задачи из примера 8.2.2. (С. 390 - 294 в учебнике Таха Х.А.) методом приоритетов

4. Реализовать задачу на языке Python с использованием фреймворка pyQt5 (либо другой библиотеки интерфейса по выбору). Предусмотреть возможность изменения исходных данных задачи (ввод данных из интерфейса и запись их в переменные).



**Решение методом весовых коэффициентов**

Обозначим через Х1 и Х2 количество рекламного времени, закупленного соответственно на радио и телевидении. Для данной задачи целевого программирования можно задать следующие частные задачи:

Минимизировать G1 = S1+(для выполнения ограничения по рекламной аудитории)

Минимизировать G2 = S2-(для выполнения условия по бюджету)

при выполнении ограничений

4Х1+8Х2+S1+ -S1- = 45 (условие по рекламной аудитории),

8Х1+24Х2+ S2+ - S2- = 100 (условие по бюджету),

Х1+2Х2 ≤ 10 (ограничение по рекламным агентам),

Х1 ≤ 6 (ограничение на рекламу по радио),

Х1, Х2, S1+,S1-, S2+, S2- ≥ 0

Менеджеры рекламного агентства считают, что выполнение условия по объёму рекламной аудитории в два раза важнее, чем выполнение условия по бюджету. Поэтому обобщённая целевая функция будет записана следующий образом.

Минимизировать Z =2G1 + G2 = 2S1+ + S2-.

Оптимальное решение этой задачи следующее : Z = 10, Х1 = 5 минут, Х2 = 2,5 минуты, S1+= 5 миллионов человек, остальные переменные равны нулю.

**Программная реализация**

***def task1(self):***

***self.paramsReset()***

***res = self.solve([0, 0, 2, 0, 0, 1],[],[])***

***self.Result.setText(res)***

***def solve(self, c, A, b):***

***A\_eq = [[int(self.radioAutitory), int(self.televisionAuditory), 1, -1, 0, 0],[int(self.radioMinuteCost), int(self.televisionMinuteCost), 0, 0, 1, -1]]***

***b\_eq = [int(self.totalAuditory), int(self.totalBedget)]***

***A\_ub = [[int(self.radioAgents), int(self.televisionAgents), 0, 0, 0, 0], [1, 0, 0, 0, 0, 0]]***

***b\_ub = [int(self.totalAgentsCount), int(self.radioUpperLimit)]***

***bounds = [(0, None), (0, None), (0, None), (0, None), (0, None), (0, None)]***

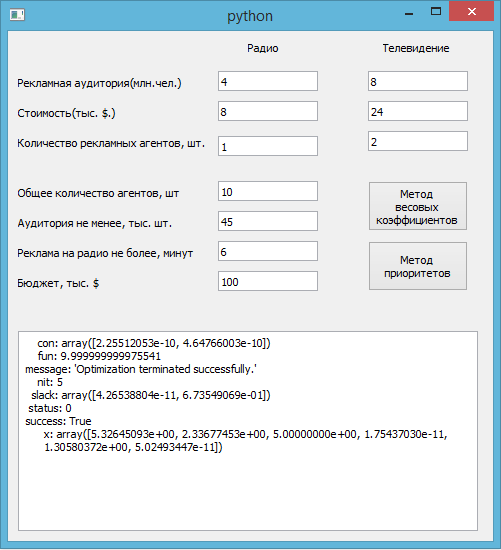
***if len(A) != 0:***

***A\_eq.append(A)***

***b\_eq.append(b)***

***res = linprog(c, A\_ub, b\_ub, A\_eq, b\_eq, bounds)***

***return res***



**Решение методом приоритетов**

В методе приоритетов поочередно решаются задачи с одной целевой функцией, начиная с задачи с целевой функцией, имеющей наивысший приоритет, и заканчивая задачей с целевой функцией, имеющей наинизший приоритет.

Предположим, что наибольший приоритет имеет частная целевая функция, соответствующая условию, накладываемому на объем рекламной аудитории.

Шаг 0. G1 ≤G2, где

G1: минимизироватьS1+ (условие по рекламной аудитории),

G2: минимизироватьS2-(условие по бюджету)

Шаг 1. Решаем первую задачу линейного программирования.

Минимизировать G1=S1+

при выполнении следующих ограничений

4Х1+8Х2+S1+ -S1- = 45 + (условие по рекламной аудитории),

8Х1+24Х2+ S2+ - S2- = 100 (условие по бюджету),

Х1+2Х2 ≤ 10 (ограничение по рекламным агентам),

+Х1 ≤ 6 (ограничение на рекламу по радио),

Х1, Х2, S1+,S1-, S2+, S2- ≥ 0

Оптимальное решение этой задачи, составляет Х1= 5 минут, Х2= 2,5 минуты,S1+= 5 миллионов человек, остальные переменные равны нулю. Решение показывает, что условие по объему рекламной аудитории не выполняется с дефицитом в 5 млн. человек. Над проектом работает 10 рекламных агентов, при этом на рекламу по радио используется 5 минут.

Поэтому введем в задачу новое ограничение S1+= 5.

Шаг 2. Решим вторую задачу линейного программирования.

Минимизировать G2=S2-

при выполнении тех же ограничений, что и в предыдущей задаче, плюс дополнительное ограничение S1+= 5

Но в данном случае в решении второй задачи нет необходимости, поскольку уже в решении первой имеем S2-= 0. Следовательно, решение первой задачи автоматически является оптимальным решением второй. Решение S2-= 0 показывает, что ограничение, касающееся бюджета рекламной компании, выполняется.

Таким образом, на основе метода приоритетов было найдено оптимальное решение задачи, которое состоит в следующем: количество минут, выделяемых на рекламу по радио и телевидению, составит 5 и 2,5 минут соответственно; над данным проектом работают 10 рекламных агентов, при этом реклама охватывает 40 миллионов рекламной аудитории и требует $100000 бюджетных вложений.

**Программная реализация**

***def task2(self):***

***self.paramsReset()***

***buf = self.solve([0, 0, 1, 0, 0, 0], [], [])***

***s1 = buf.x[2]***

***res = self.solve([0, 0, 0, 0, 0, 1], [0, 0, 1, 0, 0, 0], s1)***

***self.Result.setText(str(res))***

***def solve(self, c, A, b):***

***A\_eq = [[int(self.radioAutitory), int(self.televisionAuditory), 1, -1, 0, 0],[int(self.radioMinuteCost), int(self.televisionMinuteCost), 0, 0, 1, -1]]***

***b\_eq = [int(self.totalAuditory), int(self.totalBedget)]***

***A\_ub = [[int(self.radioAgents), int(self.televisionAgents), 0, 0, 0, 0], [1, 0, 0, 0, 0, 0]]***

***b\_ub = [int(self.totalAgentsCount), int(self.radioUpperLimit)]***

***bounds = [(0, None), (0, None), (0, None), (0, None), (0, None), (0, None)]***

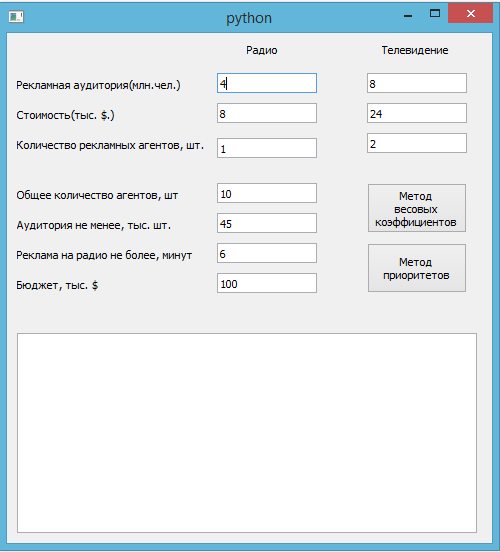
***if len(A) != 0:***

***A\_eq.append(A)***

***b\_eq.append(b)***

***res = linprog(c, A\_ub, b\_ub, A\_eq, b\_eq, bounds)***

***return res***



**Интерфейс на фреймворке PyQt5**

from PyQt5.QtWidgets import \*

from PyQt5.QtGui import QFont

def init(Form):

Form.radioAutitory = 0

Form.televisionAuditory = 0

Form.totalAuditory = 0

Form.radioUpperLimit = 0

Form.totalBedget = 0

Form.radioMinuteCost = 0

Form.televisionMinuteCost = 0

Form.radioAgents = 0

Form.televisionAgents = 0

Form.totalAgentsCount = 0

lbl1 = QLabel('Радио', Form)

lbl1.move(240, 10)

lbl2 = QLabel('Телевидение', Form)

lbl2.move(375, 10)

lbl3 = QLabel('Рекламная аудитория(млн.чел.)', Form)

lbl3.move(10, 45)

lbl4 = QLabel('Стоимость(тыс. $.)', Form)

lbl4.move(10, 75)

lbl5 = QLabel('Количество рекламных агентов, шт.', Form)

lbl5.move(10, 105)

lbl6 = QLabel('Бюджет, тыс. $', Form)

lbl6.move(10, 245)

lbl7 = QLabel('Общее количество агентов, шт', Form)

lbl7.move(10, 155)

lbl8 = QLabel('Аудитория не менее, тыс. шт.', Form)

lbl8.move(10, 185)

lbl9 = QLabel('Реклама на радио не более, минут', Form)

lbl9.move(10, 215)

Form.RadioAudience = QLineEdit(Form)

Form.RadioAudience.move(210, 40)

Form.RadioAudience.resize(100, 20)

Form.TVAudience = QLineEdit(Form)

Form.TVAudience.move(360, 40)

Form.TVAudience.resize(100, 20)

Form.Count = QLineEdit(Form)

Form.Count.move(210, 150)

Form.Count.resize(100, 20)

Form.GeneralAudience = QLineEdit(Form)

Form.GeneralAudience.move(210, 180)

Form.GeneralAudience.resize(100, 20)

Form.CostRadio = QLineEdit(Form)

Form.CostRadio.move(210, 70)

Form.CostRadio.resize(100, 20)

Form.TVCost = QLineEdit(Form)

Form.TVCost.move(360, 70)

Form.TVCost.resize(100, 20)

Form.Radio = QLineEdit(Form)

Form.Radio.move(210, 210)

Form.Radio.resize(100, 20)

Form.Budget = QLineEdit(Form)

Form.Budget.move(210, 240)

Form.Budget.resize(100, 20)

Form.AgentsRadio = QLineEdit(Form)

Form.AgentsRadio.move(210, 105)

Form.AgentsRadio.resize(100, 20)

Form.TVAgents = QLineEdit(Form)

Form.TVAgents.move(360, 100)

Form.TVAgents.resize(100, 20)

Form.Result = QTextEdit(Form)

Form.Result.move(10, 300)

Form.Result.resize(460, 200)

btn1 = QPushButton('Метод\nвесовых\nкоэффициентов', Form)

btn1.resize(100, 50)

btn1.move(360, 150)

btn1.clicked.connect(Form.task1)

btn2 = QPushButton('Метод\nприоритетов', Form)

btn2.resize(100, 50)

btn2.move(360, 210)

btn2.clicked.connect(Form.task2)

Form.show()