Министерство науки и образования РФ

Федеральное государственное бюджетное учреждение

высшего образования

**«Тверской государственный технический университет»**

(ТвГТУ)

Кафедра программного обеспечения

**Отчет по лабораторной работе №2**

по дисциплине: «Исследование операций»

Тема: «Линейное программирование. Двойственная задача»

Вариант 4

|  |
| --- |
| Выполнил:  студент группы  ПИН.РИС – 17.06  Великов Д.А. |
| Проверила:  ассистент кафедры ПО  Корнеева Е.И. |

Тверь 2020

**Задание 1**

**Постановка задачи**

А. Привести задачу ЛП к стандартному и каноническому виду;

Б. Написать к исходной задаче двойственную задачу;

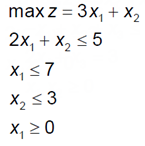
В. Найти оптимальное решение одной из задач геометрически (нарисовать задачу с использование библиотеки mathplotlib);

Г. Реализовать на языке Python консольное приложение с алгоритмом симплекс-метода.

Д. Решить реализованным алгоритмом прямую и двойственную задачи.

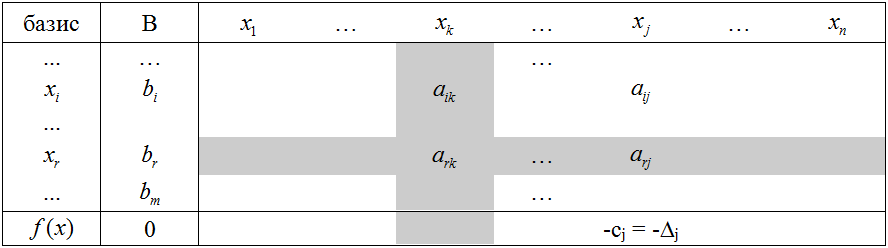
Е. Используя одну из библиотек (pulp, cvxopt.modelling, scipy.optimize) решить прямую и двойственные задачи.

Вариант задания:



**Алгоритм симплекс-метода**

В начале исходную задачу линейного программирования приводят к каноническому виду, затем составляют симплекс-таблицу вида:

[](https://vscode.ru/wp-content/uploads/2015/11/simplextable.png)

где в столбце «базис» указываются базисные переменные, а в последней строке столбца «базис» пишется f(x). В столбец «B» записываются свободные члены ограничений bi и значение целевой функции (на 1-м этапе оно равно 0, т.е. никакой прибыли).

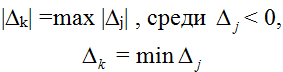
В столбцах xj для не базисных переменных указываются коэффициенты при не базисных переменных из ограничений задачи. В столбцах базисных переменных содержится только 0 или 1 на пересечении столбца с соответствующей строкой базисной переменной.

В последней строке -cj – это коэффициенты при переменных целевой функции взятые с противоположным знаком.

Симплекс-таблица составлена, теперь опишем сам симплекс-метод.

**Шаг 1:** Выполняется проверка полученного базисного плана на оптимальность по условию: если при каком-либо ДБР (допустимое базисное решение) в симплекс-таблице все коэффициенты строки f(x) (то есть -cj) не отрицательны, то данное ДБР оптимально, следовательно **КОНЕЦ** решения. В противном случае:

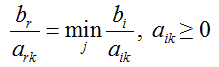
**Шаг 2:** Переход к новому базисному плану. Для этого из числа небазисных переменных с отрицательными значениями в последней строке (то есть -cj < 0) выбирается переменная, вводимая в базис – xk, это переменная которой соответствует наибольшая по модулю отрицательная оценка:

[](https://vscode.ru/wp-content/uploads/2015/11/leading-the-column.png)

Столбец, отвечающий переменной xk, называется главным, или ведущим. Элементы данного столбца обозначаются через aik.

Если окажется несколько одинаковых наибольших по модулю отрицательных оценок, то выбирается любая из соответственных переменных.

**Шаг 3:** Выбираем переменную r – переменную, которая выводится из базиса. Данная переменная находится из соотношения:

[](https://vscode.ru/wp-content/uploads/2015/11/leading-the-line.png)

Строка таблицы, в которой получено наименьшее отношение элемента столбца «В» к соответствующему положительному элементу ведущего столбца, является ведущей, или главной.

Элементы главной строки обозначаются через arj. Выбранная переменная xr будет выводиться из базиса, то есть это исключаемая переменная.

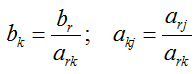
Если окажется несколько одинаковых наименьших значений отношений, то выбирается любая из соответствующих им переменных.

Элемент, который стоит на пересечении главного столбца и строки называется главным, или ведущим, и обозначается ark.

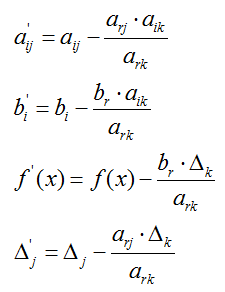
**Шаг 4:** Для определения нового базисного плана проводится пересчет элементов симплекс-таблицы, и результаты заносятся в новую таблицу. Выбранные переменные среди базисных и не базисных, лежащих на главной строке и главном столбце, меняются местами.

Процедура пересчета элементов выполняется следующим образом:

а) элементы главной строки необходимо разделить на ведущий элемент:

[](https://vscode.ru/wp-content/uploads/2015/11/the-change-of-basis-1.png)

б) элементы полученной строки умножаются на -aik, и результаты складываются с i-той строкой, причем i ≠ k:

[](https://vscode.ru/wp-content/uploads/2015/11/the-change-of-basis-2.png)

После определения новой симплекс-таблицы переходят к **шагу 1**.

**Реализация на языке Python**

**def calculate(lines):**

**table = []**

**basis = []**

**result = []**

**rows = []**

**buffer = []**

**m = 0**

**n =0**

**#text = "25 -3 5\n30 -2 5\n10 1 0\n6 3 -8\n0 -6 -5"**

**#lines = text.split('\n')**

**m = len(lines)**

**for l in range(0,len(lines)):**

**rows = list(map(int, lines[l].split()))**

**buffer.append(rows)**

**n = len(rows)**

**for i in range(0,m):**

**table.append([])**

**for j in range(0,n+m-1):**

**if j<n:**

**table[i].append(buffer[i][j])**

**else:**

**table[i].append(0)**

**if (n+i)<n+m-1:**

**table[i][n+i]=1**

**basis.append(n+i)**

**for i in range(0,2):**

**result.append(0)**

**n = n+m-1**

**mainCol =0**

**mainRow = 0**

**while True:**

**flag = True**

**for j in range(1,n):**

**if(table[m-1][j]<0):**

**flag=False**

**break**

**if flag == True:**

**break**

**mainCol = 1**

**for j in range(2,n):**

**if(table[m-1][j]<table[m-1][mainCol]):**

**mainCol=j**

**mainRow = 0**

**for i in range(0,m-1):**

**if(table[i][mainCol]>0):**

**mainRow=i**

**break**

**for i in range(mainRow+1,m-1):**

**if ((table[i][mainCol]>0) and ((table[i][0] / table[i][mainCol])<(table[mainRow][0]/table[mainRow][mainCol]))):**

**mainRow=i**

**basis[mainRow] = mainCol**

**new\_table = []**

**for i in range(0,m):**

**new\_table.append([])**

**for j in range(0,n):**

**new\_table[i].append(0)**

**for j in range(0,n):**

**new\_table[mainRow][j] = table[mainRow][j]/table[mainRow][mainCol]**

**for i in range(0,m):**

**if i== mainRow:**

**continue**

**for j in range(0,n):**

**new\_table[i][j] = table[i][j]-table[i][mainCol]\*new\_table[mainRow][j]**

**table=new\_table**

**for i in range(0,len(result)):**

**k = -1**

**for j in range(0,len(basis)):**

**if basis[j]==i+1:**

**k=j**

**break**

**if k!=-1:**

**result[i]=table[k][0]**

**else:**

**result[i]=0**

**out = "Симплекс таблица:\n"**

**for i in range(0,m):**

**for j in range(0,n):**

**out=out+str(round(table[i][j],2))+" | "**

**out=out+"\n"**

**out=out+"\nРешение:\nX[1] = "+str(result[0])+"\nX[2] = "+str(result[1])**

**print(out)**

**Переход к стандартной форме ЗЛП**.   
  
**Переход к СЗЛП**.   
Расширенная матрица системы ограничений-равенств данной задачи:

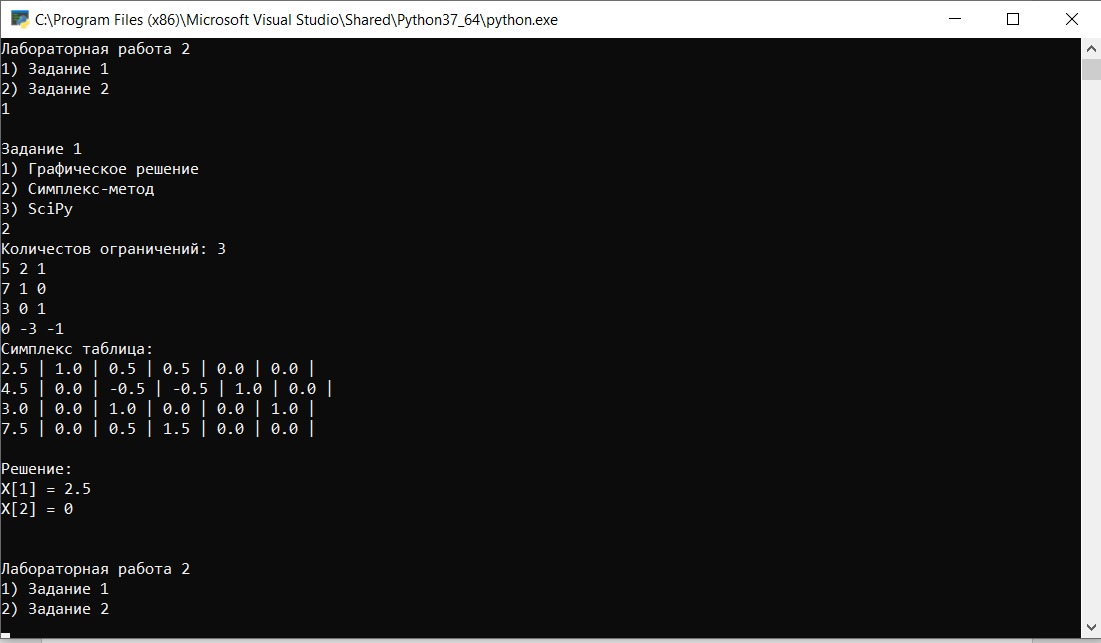
|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | | 2 | 1 | 5 | | 1 | 0 | 7 | | 0 | 1 | 3 | |  | |  |

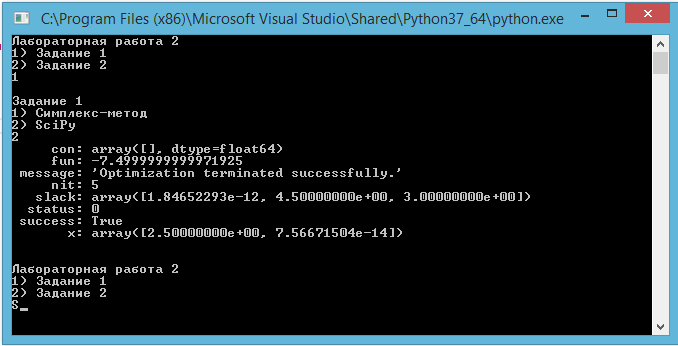
Приведем систему к единичной матрице методом жордановских преобразований.   
Поскольку в системе имеется единичная матрица, то в качестве базисных переменных принимаем X = ().   
Соответствующие уравнения имеют вид:   
2x1+x2 = 5   
x1 = 7   
x2 = 3   
Выразим базисные переменные через остальные:   
x = -2x1-x2+5   
x = -x1+7   
x = -x2+3   
Подставим их в целевую функцию:   
F(X) = 3x1+x2   
или   
F(X) = 3x1+x2 → max   
Система неравенств:   
-2x1-x2+5 ≥ 0   
-x1+7 ≥ 0   
-x2+3 ≥ 0   
Приводим систему неравенств к следующему виду:   
2x1+x2 ≤ 5   
x1 ≤ 7   
x2 ≤ 3   
F(X) = 3x1+x2 → max   
Упростим систему.   
2x1+x2 ≤ 5   
x1 ≤ 7   
x2 ≤ 3   
F(X) = 3x1+x2 → max

**Решение задачи симплекс-методом**

Определим максимальное значение целевой функции F(X) = 3x1+x2 при следующих условиях-ограничений.   
2x1+x2≤5   
x1≤7   
x2≤3   
Для построения первого опорного плана систему неравенств приведем к системе уравнений путем введения дополнительных переменных (**переход к канонической форме**).   
В 1-м неравенстве смысла (≤) вводим базисную переменную x3. В 2-м неравенстве смысла (≤) вводим базисную переменную x4. В 3-м неравенстве смысла (≤) вводим базисную переменную x5.   
2x1+x2+x3 = 5   
x1+x4 = 7   
x2+x5 = 3   
Введем новую переменную x0 = 3x1+x2.   
Выразим базисные переменные <3, 4, 5> через небазисные (свободные).   
**Базисное решение** называется допустимым, если оно неотрицательно.   
x0 = 0+3x1+x2   
x3 = 5-2x1-x2   
x4 = 7-x1   
x5 = 3-x2   
Переходим к основному алгоритму симплекс-метода.   
Поскольку задача решается на максимум, то переменную для включения в текущий план выбирают по максимальному положительному числу в уравнении для x0.   
**1. Проверка критерия оптимальности**.   
В выражении для x0 присутствуют положительные элементы. Следовательно, текущий план неоптимален.   
**2. Определение новой базисной переменной**.   
Поскольку коэффициент при переменной x1 больше, чем при остальных переменных, то при увеличении x1 целевая функция будет увеличиваться быстрее.   
max(3,1,0,0,0) = 3   
x0 = 0+3x1+x2   
x3 = 5-2x1-x2   
x4 = 7-x1   
x5 = 3-x2   
В качестве новой переменной выбираем x1.   
**3. Определение новой свободной переменной**.   
Вычислим значения Di по всем уравнениям для этой переменной: bi / ai1 и из них выберем наименьшее:   
min (5 : 2 , 7 : 1 , - ) = 21/2   
Вместо переменной x3 в план войдет переменная x1.   
**4. Пересчет всех уравнений**.   
Выразим переменную x1 через x3   
x1 = 5/2-1/2x2-1/2x3   
и подставим во все выражения.   
x0 = 0+3(5/2-1/2x2-1/2x3)+x2   
x4 = 7-(5/2-1/2x2-1/2x3)   
x5 = 3-x2   
После приведения всех подобных, получаем новую систему, эквивалентную прежней:   
x0 = 15/2-1/2x2-3/2x3   
x1 = 5/2-1/2x2-1/2x3   
x4 = 9/2+1/2x2+1/2x3   
x5 = 3-x2   
Полагая небазисные переменные x = (1, 4, 5) равными нулю, получим новый допустимый вектор и значение целевой функции:   
x = (0, 1/2, 3/2, 0, 0), x0 = 15/2   
Выражение для x0 не содержит положительных элементов. Найден оптимальный план.   
Окончательный вариант системы уравнений:   
x0 = 15/2-1/2x2-3/2x3   
x1 = 5/2-1/2x2-1/2x3   
x4 = 9/2+1/2x2+1/2x3   
x5 = 3-x2   
Оптимальный план можно записать так:   
x1 = 21/2, x2 = 0   
F(X) = 3\*21/2 + 1\*0 = 71/2

**Результат программного решения**





**Двойственная задача**

Так как в прямой задаче требуется найти максимум фунции, то приведем первоначальное условие к виду  
{F(x) = СT x| Ax≤B, xi ≥0, i = 1,m}  
В результате получим следующие матрицы:

|  |  |  |  |
| --- | --- | --- | --- |
| CT= | |  |  | | --- | --- | | 3 | 1 | |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| A= | |  |  | | --- | --- | | 2 | 1 | | 1 | 0 | | 0 | 1 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| B= | |  | | --- | | 5 | | 7 | | 3 | |

Для составления двойственной задачи линейного программирования найдем матрицы АT, BT, C

|  |  |  |  |
| --- | --- | --- | --- |
| C= | |  | | --- | | 3 | | 1 | |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| AT= | |  |  |  | | --- | --- | --- | | 2 | 1 | 0 | | 1 | 0 | 1 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| BT= | |  |  |  | | --- | --- | --- | | 5 | 7 | 3 | |

Следовательно, двойственная задача линейного программирования будет иметь вид:

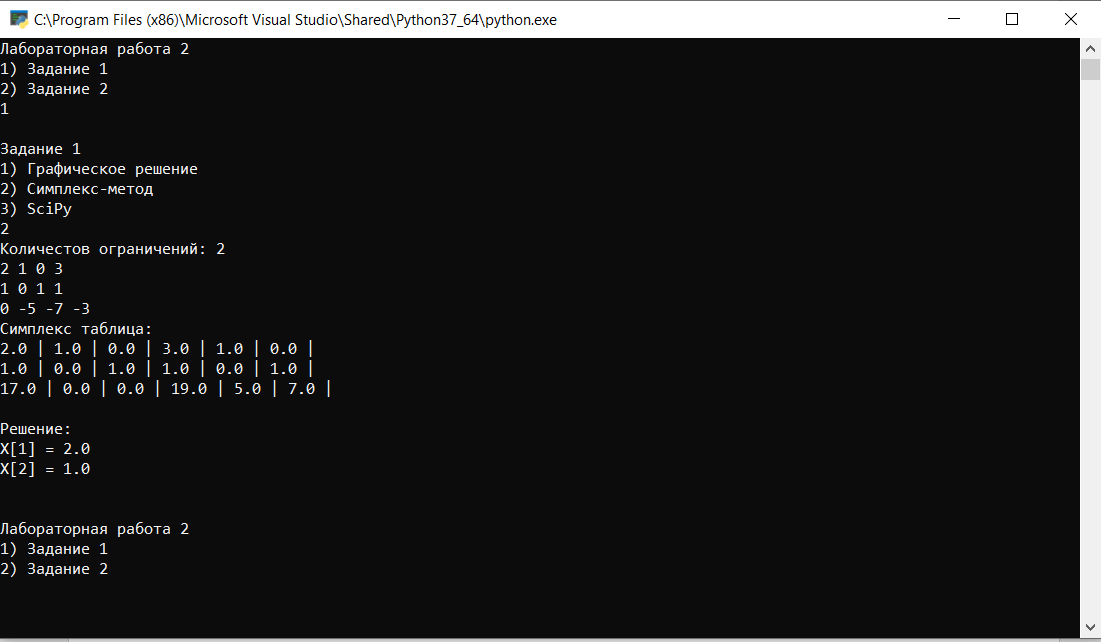
**F(Y)=5Y1+7Y2+3Y3 (min)**

Ограничения:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 2Y1 | + | 1Y2 | + | 0Y3 |  | ≥ | 3 |
| 1Y1 | + | 0Y2 | + | 1Y3 |  | ≥ | 1 |

|  |  |  |
| --- | --- | --- |
| Y1 | ≥ | 0 |
| Y2 | ≥ | 0 |
| Y3 | ≥ | 0 |

**Результат решения**



**Задание 2**

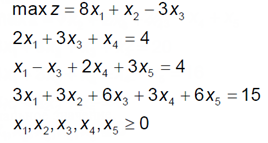
**Постановка задачи**

А. Построить двойственную задачу ЛП;

Б. Найти оптимальное решение двойственной задачи ЛП симплекс-методом;

В. Найти оптимальное решение двойственной задачи с помощью одной из библиотек (pulp, cvxopt.modelling, scipy.optimize).

Вариант задания:



Так как в прямой задаче требуется найти максимум фунции, то приведем первоначальное условие к виду  
{F(x) = СT x| Ax≤B, xi ≥0, i = 1,m}  
В результате получим следующие матрицы:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| CT= | |  |  |  |  |  | | --- | --- | --- | --- | --- | | 8 | 1 | -3 | 0 | 0 | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| A= | |  |  |  |  |  | | --- | --- | --- | --- | --- | | 2 | 0 | 3 | 1 | 0 | | 1 | 0 | -1 | 2 | 3 | | 3 | 3 | 6 | 3 | 6 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| B= | |  | | --- | | 4 | | 4 | | 15 | |

Для составления двойственной задачи линейного программирования найдем матрицы АT, BT, C

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| C= | |  | | --- | | 8 | | 1 | | -3 | | 0 | | 0 | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| AT= | |  |  |  | | --- | --- | --- | | 2 | 1 | 3 | | 0 | 0 | 3 | | 3 | -1 | 6 | | 1 | 2 | 3 | | 0 | 3 | 6 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| BT= | |  |  |  | | --- | --- | --- | | 4 | 4 | 15 | |

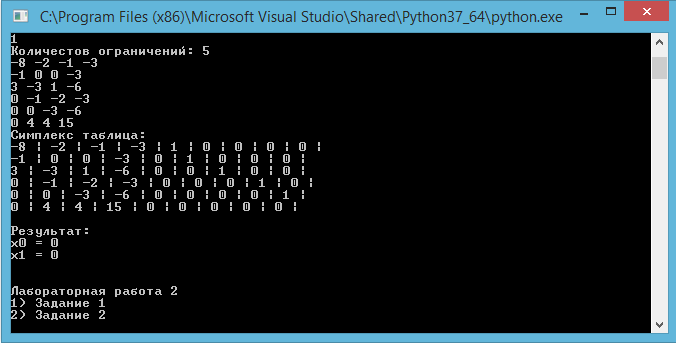
Следовательно, двойственная задача линейного программирования будет иметь вид:

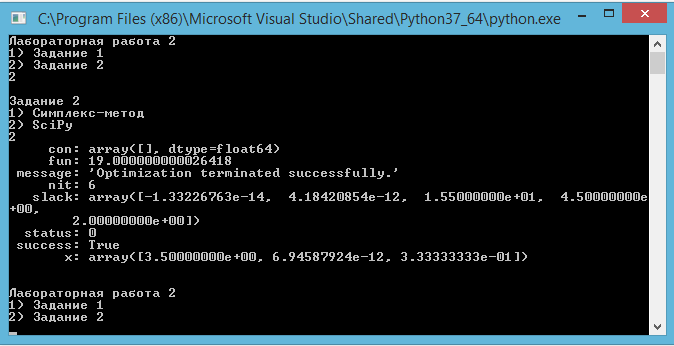
**F(Y)=4Y1+4Y2+15Y3 (min)**

Ограничения:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 2Y1 | + | 1Y2 | + | 3Y3 |  | ≥ | 8 |
| 0Y1 | + | 0Y2 | + | 3Y3 |  | ≥ | 1 |
| 3Y1 | - | 1Y2 | + | 6Y3 |  | ≥ | -3 |
| 1Y1 | + | 2Y2 | + | 3Y3 |  | ≥ | 0 |
| 0Y1 | + | 3Y2 | + | 6Y3 |  | ≥ | 0 |

**Результат решения**





Ссылка — <https://github.com/veldm/Simplex>