Министерство образования и науки РФ

Ульяновский государственный технический университет

Курсовой проект

по курсу «теория оптимизации информационных систем»

Тема: Решение задач линейного программирования

Выполнил: студент 1 курса
Шаблыгин В.В.
гр. ИСТМД-11
Руководитель: Новиков А.А.

УЛЬЯНОВСК 2021

Оглавление

Введение

Линейное программирование – это раздел математики, ориентированный на нахождение экстремума (максимума или минимума) в задачах, которые описываются линейными уравнениями. Причем, линейным уравнением описывается как сама целевая функция, так и переменные (входные параметры).

Необходимым условием задач линейного программирования является обязательное наличие ограничений на ресурсы (сырье, материалы, финансы, спрос).

Еще одним условием решения задачи является выбор критерия – основа алгоритма, т.е. целевая функция должна быть оптимальной, и эта оптимальность должна быть выражена количественно.

Критерий оптимальности дожжен удовлетворять следующим требованиям:

1) Быть единственным для данной задачи;

2) Измеряться в единицах количества;

3) Линейно зависеть от входных параметров.

Стандартная задача линейного программирования – задача, в которой требуется определить максимальное либо минимальное значение целевой функции при ограничениях неравенств и условиях.

Каноническая задача линейного программирования – задача, которая заключается в определении максимального значения целевой функции при выполнении ограничений уравнений.

Исходя из вышесказанного, можно сформулировать задачу линейного программирования в общем виде:

найти экстремум целевой функции

$$F(X) = c_1 x_1 + c_2 x_2 + c_3 x_3 + \ldots + c_n x_n \to \max(\min)$$

при ограничениях в виде равенств:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \ldots + a_{1n}x_n = b_1; \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + \ldots + a_{2n}x_n = b_2; \\ \ldots \\ a_{n1}x_1 + a_{n2}x_2 + a_{n3}x_3 + \ldots + a_{nm}x_n = b_m; \end{cases}$$

при ограничениях в виде неравенств:

$$\begin{cases} a_{p1}x_1 + a_{p2}x_2 + a_{p3}x_3 + \ldots + a_{pn}x_n < b_p; \\ a_{r1}x_1 + a_{r2}x_2 + a_{r3}x_3 + \ldots + a_{rn}x_n < b_r; \\ \ldots \\ a_{v1}x_1 + a_{v2}x_2 + a_{v3}x_3 + \ldots + a_{vn}x_n < b_v; \end{cases}$$

и условиях неотрицательности входных параметров:

$$x_1 \geq 0; \quad x_2 \geq 0; \quad \ldots; \quad x_n \geq 0.$$

В краткой форме задача линейного программирования может быть записана так:

$$L_{(x)} = \sum_{i=1}^{n} c_i x_i$$

при условии

$$+\sum_{i=1}^{n} a_{ij}x_i \leq b_i \; (\text{или} \geq b_1) \text{ при } j = 1,2,\ldots,m, \quad x_i \geq 0, \quad i = \overline{1,n},$$

где $x_1, \ldots, x_n$ - входные переменные;

$c_1, \ldots, c_n;\ a_{11}, \ldots, a_{nv};\ b_1, \ldots, b_v$ - числа положительные, отрицательные и равные нулю.

В матричной форме эта задача может быть записана так:

$$cx \to \max, \text{при } \boldsymbol{Ax} \leq \boldsymbol{b} \text{ или } \boldsymbol{Ax} \geq \boldsymbol{b};\ x \geq 0$$

Задачи линейного программирования можно решить аналитически и графически.

Основные теоремы линейного программирования

Чтобы найти оптимальное решение среди бесчисленного множества допустимых решений системы ограничений в задаче линейного программирования любого вида, понадобится ряд теорем, к рассмотрению которых мы и переходим.

Теорема 1. Множество всех допустимых решений системы ограничений задачи линейного программирования является выпуклым.

Множество решений задачи линейного программирования определяется совокупностью линейных ограничений, поэтому такое множество геометрически представляет собой выпуклый многогранник или неограниченную многогранную область, за исключением тех случаев, когда система ограничений несовместна.

О том, что такое выпуклые множества - на уроке Системы линейных неравенств и выпуклые множества точек.

Теорема 2. Если существует, и притом единственное, оптимальное решение задачи линейного программирования, то оно совпадает с одной из угловых точек множества допустимых решений.

Эта теорема позволяет сделать вывод, что поиски оптимального решения можно ограничить перебором конечного числа угловых точек. Однако для отыскания угловых точек требуется построение области решений системы ограничений. Это построение возможно только для двух- или трёхмерного пространства, а в общем случае задача остаётся неразрешимой. Следовательно, нужно располагать каким-то аналитическим методом, позволяющим находить координаты угловых точек. Для этого понадобятся следующие две теоремы.

Теорема 3. Каждому допустимому базисному решению задачи линейного программирования соответствует угловая точка области допустимых решений системы ограничений.

Теорема 4 (обратная). Каждой угловой точке множества допустимых решений системы ограничений соответствует допустимое базисное решение.

Следствие. Если существует, и притом единственное, оптимальное решение задачи линейного программирования, то оно совпадает с одним из допустимых базисных решений системы ограничений.

Справедливость этого утверждения вытекает из теорем 2 и 4.

Итак, оптимум линейной формы нужно искать среди конечного числа допустимых базисных решений. Однако даже в простейших задачах линейного программирования (при небольших значениях m и n) нахождение оптимального решения путём рассмотрения всех базисных решений является крайне трудоёмким процессом, поскольку число базисных решений может быть весьма велико. Поэтому нужна какая-то вычислительная схема, позволяющая осуществлять переход от одного допустимого базисного решения к другому, при котором линейная форма или приблизилась к оптимуму, или, по крайней мере не изменила своего значения. Такой вычислительной схемой является, например, симплекс-метод решения задач линейного программирования.

Глава 1. Решение задач линейного программирования графическим методом.

Теоретические основы.

В этой главе рассмотрим графический метод решения задач линейного программирования, то есть, таких задач, в которых требуется найти такое решения системы линейных уравнений и (или) неравенств (системы ограничений), при котором функция цели - линейная функция - принимает оптимальное значение.

Ввиду того, что наглядность графического решения достигается лишь на плоскости, мы можем познакомиться с графическим представлением задачи только в двумерном пространстве. Это представление пригодно для системы ограничений-неравенств с двумя переменными или для систем уравнений, в которых число переменных на 2 превышает число уравнений, то есть число свободных переменных равно двум.

Поэтому графический метод имеет такие узкие рамки применения, что о нём как об особом методе решения задач линейного программирования говорить нельзя.

Однако для выработки наглядных представлений о решениях задач линейного программирования графический метод представляет определённый интерес. Кроме того, он позволяет геометрически подтвердить справедливость теорем линейного программирования.

Итак, задача линейного программирования. Требуется найти неотрицательные значения переменных x_1 и x_2, удовлетворяющих системе неравенств

$$\begin{cases} a_{11}x_1 + a_{12}x_2 \le b_1 \\ a_{21}x_1 + a_{22}x_2 \le b_2 \\ \dots \\ a_{n1}x_1 + a_{n2}x_2 \le b_n \end{cases}$$

при которых линейная форма $F = c_1x_1 + c_2x_2$ принимает оптимальное значение.

1.2 Пример решения задачи линейного программирования графическим методом.

Задача: найти минимум целевой функции:

$Z = 13x1 + 29x2 + 19x3 \rightarrow \min$

При следующих ограничениях:

$x1 + x2 + x3 = 13$

$3 \geq x1 \geq 2$

$x2 \geq 0.2$

$x3 \geq 0.2$

Выразим x3 через две другие переменные:

$x3 = 13 – x1 – x2$,

тогда целевая функция примет вид:

$Z = 13x1 + 29x2 + 19(13 - x1 - x2) =>$

$Z = -6x1 + 10x2 + 247$

Построим область допустимых значений, т.е. решим графически систему неравенств. Для этого построим каждую прямую и определим полуплоскости, заданные неравенствами (рис. №2).

$x1 + x2 = 0$

$x1 \geq 2$

$x1 \leq 3$

$x2 \geq 0.2$

Определим направление возрастания функции:

Для $Z = -6x1 + 10x2 + 247$

$dZ / dx1 = -6$

$dZ / dx2 = 10$

Значит вершина вектора-градиента находится в точке (-6, 10), а начало в точке (0, 0).

Если прямую, перпендикулярную вектору-градиенту, двигать в противоположную направлению возрастания сторону, то последняя точка соприкосновения с областью допустимых значений, будет минимумом функции (см. рис №1).

В нашем случае это точка **d** со значениями:

$x1 = 3, x2 = 0.2$

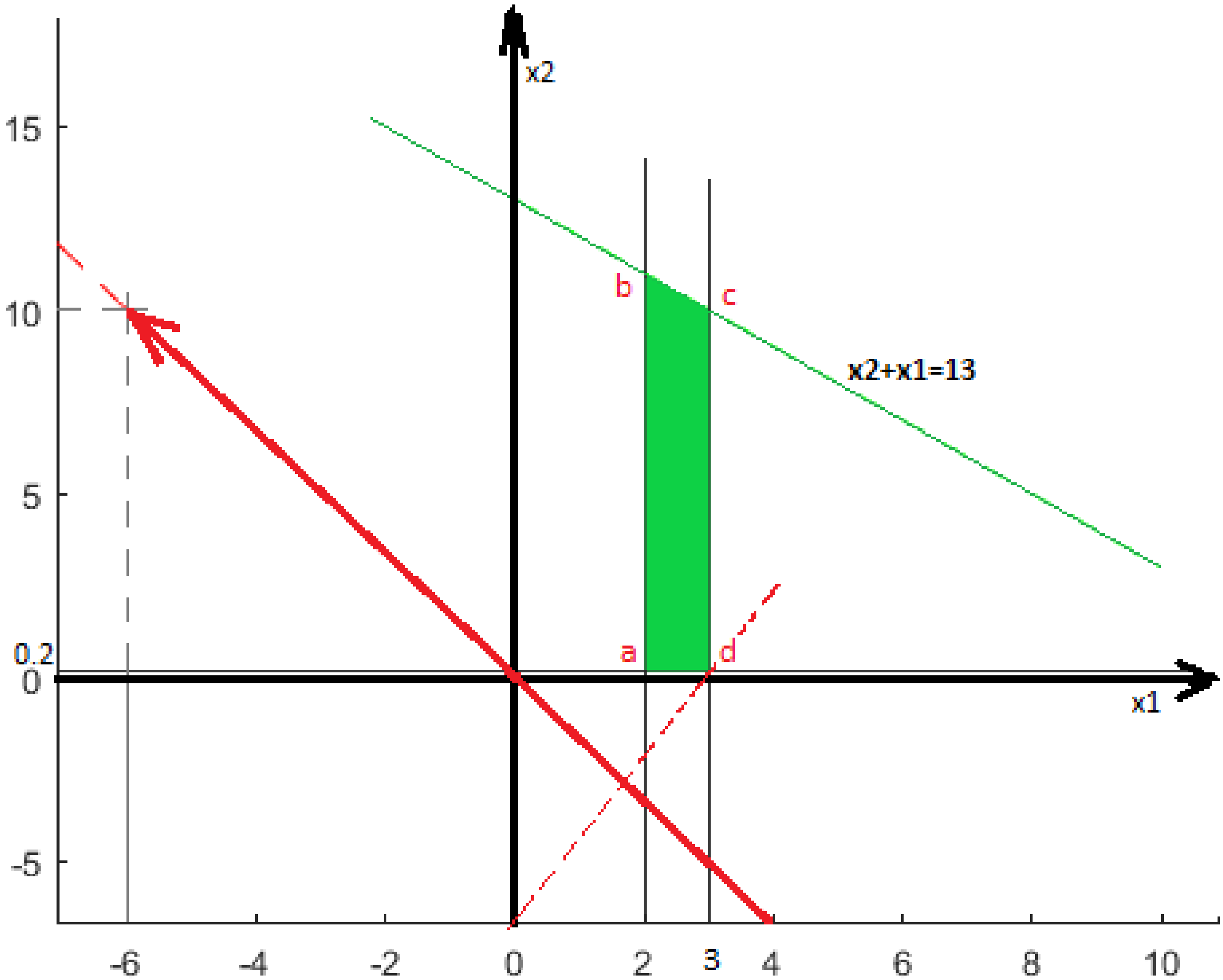

Рис. №1. Графическое решение ЗЛП.

Найдем целевой функции в точке минимума:
x3 = 13 - x1 - x2 = 13 – 3 – 0.2 = 9.8
Z = 13*3 + 29*0.2 + 19*9.8 = 231

Вычислим значения целевой функции в других точках.
Для точки **a**:
x1 =2
x2 = 0.2
x3 = 13 – 2 – 0.2 = 10.8
Z = 13*2 + 29*0.2 + 19*10.8 = 237

Для точки **b**:
x1 =3
x2 = 13 - 3 = 10
x3 = 0
Z = 13*3 + 29*10 + 19*0 = 287

Для точки **c**:
x1 =2

x2 = 13 - 2 = 9

x3 = 13 – 2 – 9 = 2

Z = 13*2 + 29*9 + 19*2 = 325

В других точках значение целевой функции больше, значит решение верное.

Вывод: целевая функция Z = 13x1 + 29x2 + 19x3 принимает минимальное значение при следующих значениях переменных:

x1 = 3

x2 = 0.2

x3 = 9.8

Глава 2. Решение задач линейного программирования симплекс-методом.

2.1 Теоретические основы.

Симплекс метод — это метод последовательного перехода от одного базисного решения (вершины многогранника решений) системы ограничений задачи линейного программирования к другому базисному решению до тех пор, пока функция цели не примет оптимального значения (максимума или минимума).

Симплекс-метод является универсальным методом, которым можно решить любую задачу линейного программирования, в то время, как графический метод пригоден лишь для системы ограничений с двумя переменными.

Симплекс метод был предложен американским математиком Р.Данцигом в 1947 году, с тех пор для нужд промышленности этим методом нередко решаются задачи линейного программирования с тысячами переменных и ограничений.

Всякое неотрицательное решение системы ограничений называется допустимым решением.

Пусть имеется система m ограничений с n переменными (m < n).

Допустимым базисным решением является решение, содержащее m неотрицательных основных (базисных) переменных и n - m неосновных. (небазисных, или свободных) переменных. Неосновные переменные в базисном решении равны нулю, основные же переменные, как правило, отличны от нуля, то есть являются положительными числами.

Любые m переменных системы m линейных уравнений с n переменными называются основными, если определитель из коэффициентов при них отличен от нуля. Тогда остальные n - m переменных называются неосновными (или свободными).

Алгоритм симплекс метода

Шаг 1. Привести задачу линейного программирования к канонической форме. Для этого перенести свободные члены в правые части (если среди этих свободных членов окажутся отрицательные, то соответствующее уравнение или неравенство умножить на - 1) и в каждое ограничение ввести дополнительные переменные (со знаком "плюс", если в исходном неравенстве знак "меньше или равно", и со знаком "минус", если "больше или равно").

Шаг 2. Если в полученной системе m уравнений, то m переменных принять за основные, выразить основные переменные через неосновные и найти соответствующее базисное решение. Если найденное базисное решение окажется допустимым, перейти к допустимому базисному решению.

Шаг 3. Выразить функцию цели через неосновные переменные допустимого базисного решения. Если отыскивается максимум (минимум) линейной формы и в её выражении нет неосновных переменных с отрицательными (положительными) коэффициентами, то критерий оптимальности выполнен и полученное базисное решение является оптимальным - решение окончено. Если при нахождении максимума (минимума) линейной формы в её выражении имеется одна или несколько неосновных переменных с отрицательными (положительными) коэффициентами, перейти к новому базисному решению.

Шаг 4. Из неосновных переменных, входящих в линейную форму с отрицательными (положительными) коэффициентами, выбирают ту, которой соответствует наибольший (по модулю) коэффициент, и переводят её в основные. Переход к шагу 2.

Важные условия

Если допустимое базисное решение даёт оптимум линейной формы (критерий оптимальности выполнен), а в выражении линейной формы через неосновные переменные отсутствует хотя бы одна из них, то полученное оптимальное решение - не единственное.

Если в выражении линейной формы имеется неосновная переменная с отрицательным коэффициентом в случае её максимизации (с положительным - в случае минимизации), а во все уравнения системы ограничений этого шага указанная переменная входит также с отрицательными коэффициентами или отсутствует, то линейная форма не ограничена при данной системе ограничений. В этом случае её максимальное (минимальное) значение записывают в виде $F_{\max} = \infty \quad (F_{\min} = -\infty)$.

2.2 **Пример решения задачи линейного программирования симплекс-методом.**

Задача: найти минимум целевой функции:

Z = 13x1 + 29x2 + 19x3 → min

При следующих ограничениях:

$$\begin{cases} 17x1 + 21x2 + 5x3 <= 150 \\ x1 + x2 + x3 = 13 \\ 3 > x1 > 2 \\ x2 > 0.2 \\ x3 > 0.2 \\ x1 > 0,\ x2 > 0,\ x3 >= 0 \end{cases}$$

Решение:

Свободные члены системы должны быть неотрицательными.

Данное условие выполнено.

Каждое ограничение системы должно представлять собой уравнение, для этого введем дополнительные переменные x4, x5, x6, x7:

17x1	+ 21x2	+ 5x3	+ x4				= 150
x1	+ x2	+ x3					= 13
x1				+x5			= 3
	x2				- x6		= 0.2
		x3				-x7	= 0.2

x4 ≥ 0, x5 ≥ 0, x6 ≥ 0, x7 ≥ 0

Нахождение начального базиса и значения функции Z, которое соответствует найденному начальному базису.

17x1	+ 21x2	+ 5x3	**+ x4**				= 150
x1	+ x2	+ x3					= 13
x1				**+x5**			= 3
	x2				- x6		= 0.2
		x3				-x7	= 0.2

Для того, чтобы продолжить решение задачи, в каждом уравнении должна быть переменная с коэффициентом «1» и не встречающаяся в других уравнениях. Чтобы выполнить это условие, добавим дополнительные переменные x8, x9, x10 в те уравнения, где нет базисной переменной:

17x1	+ 21x2	+ 5x3	**+ x4**							150
x1	+ x2	+ x3					**+x8**			13
x1				**+x5**						3
	x2				- x6			**+x9**		0.2
		x3				-x7			**+x10**	0.2

Шаг 1.

Введем дополнительную функцию W и далее будем искать ее минимальное значение.

W = x8+x9+x10

x8= 13-x1-x2-x3, x9= 0.2-x2+x6, x10=0.2-x3+x7

тогда:

W = (13-x1-x2-x3) + (0.2-x2+x6) + (0.2-x3+x7) = 13.2 - x1 - 2x2 - 2x3 + x6 + x7

Приравняем свободные переменные (x1, x2, x3, x6, x7) к нулю, тогда:

x4 = 150, x5 = 3, x8 = 13, x9 = 0.2, x10 = 0.2

W = 13.4

В выделенной строке выбираем наименьший отрицательный коэффициент (можно выбрать любой отрицательный). Для положительных коэффициентов выбранного столбца считаем отношение свободный член/базис и выбираем наименьшее значение. Далее все элементы выбранной строки умножаем на нужный коэффициент и складываем с каждой строкой, таким образом, чтобы в выбранном столбце все значения, кроме базисного, стали нулями.

базис	св.ч	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	св.ч/б
X4	150	17	21	5	**1**	0	0	0	0	0	0	150:17=8.82
X8	13	1	1	1	0	0	0	0	**1**	0	0	13:1 = 13
X5	3	1	0	0	0	**1**	0	0	0	0	0	**3:1 = 3**
X9	0.2	0	1	0	0	0	-1	0	0	**1**	0	
X10	0.2	0	0	1	0	0	0	-1	0	0	**1**	
W	13.4	**-1**	-2	-2	0	0	1	1	0	0	0	

базис	св.ч	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	
X4	99	0	21	5	**1**	-17	0	0	0	0	0	
X8	10	0	1	1	0	0	0	0	**1**	0	0	
X1	3	1	0	0	0	**1**	0	0	0	0	0	
X9	0.2	0	1	0	0	0	-1	0	0	**1**	0	
X10	0.2	0	0	1	0	0	0	-1	0	0	**1**	
W	10.4	0	**-2**	-2	0	1	1	1	0	0	0	

Приравняем свободные переменные (x2, x3, x5, x6, x7) к нулю, тогда:
x1= 3, x4 = 99, x8 = 10,   x9 = 0.2,   x10 = 0.2
W = 10.4

Шаг 2.

базис	св.ч	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	св.ч/б
X4	99	0	21	5	**1**	-17	0	0	0	0	0	99/21=4.71
X8	10	0	1	1	0	0	0	0	**1**	0	0	10/1=10
X1	3	1	0	0	0	**1**	0	0	0	0	0	
X9	0.2	0	1	0	0	0	-1	0	0	**1**	0	0.2/1=0.2
X10	0.2	0	0	1	0	0	0	-1	0	0	**1**	
W	10.4	0	**-2**	-2	0	1	1	1	0	0	0	

базис	св.ч	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	
X4	94.8	0	0	5	**1**	-17	21	0	0	-21	0	
X8	9.8	0	0	1	0	0	1	0	**1**	-1	0	
X1	3	1	0	0	0	**1**	0	0	0	0	0	
X2	0.2	0	1	0	0	0	-1	0	0	**1**	0	
X10	0.2	0	0	1	0	0	0	-1	0	0	**1**	
W	10	0	**0**	-2	0	1	-1	1	0	2	0	

Приравняем свободные переменные (x3, x5, x6, x7, x9) к нулю, тогда:
x1= 3, x2 =0.2, x4 = 94.8   x8 = 9.8,   x10 = 0.2
W = 10

Шаг 3.

базис	св.ч	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	св.ч/б
X4	94.8	0	0	5	**1**	-17	21	0	0	-21	0	99.8/21=4.51
X8	9.8	0	0	1	0	0	1	0	**1**	-1	0	9.8/1=9.8
X1	3	1	0	0	0	**1**	0	0	0	0	0	
X2	0.2	0	1	0	0	0	-1	0	0	**1**	0	
X10	0.2	0	0	1	0	0	0	-1	0	0	**1**	
W	19.8	0	**0**	-2	0	1	-1	1	0	2	0	

базис	св.ч	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	
X6	158/35	0	0	5/21	**1/21**	-17/21	1	0	0	-1	0	
X8	37/7	0	0	16/21	-1/21	-4/21	0	0	**1**	0	0	
X1	3	1	0	0	0	**1**	0	0	0	0	0	
X2	33/7	0	1	5/21	1/21	-17/21	-1	0	0	**0**	0	
X10	0.2	0	0	1	0	0	0	-1	0	0	**1**	
W	192/35	0	0	-37/21	1/21	4/21	0	1	0	1	0	

Приравняем свободные переменные (x3, x4, x5, x7, x9) к нулю, тогда:
x1= 3, x2 =33/7, x4 = 158/35   x8 = 33/7,   x10 = 0.2
W = 192/35

Шаг 4.

базис	св.ч	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	св.ч/б
X6	158/35	0	0	5/21	**1/21**	-17/21	1	0	0	-1	0	158/35 : 5/21 = 18,96
X8	37/7	0	0	16/21	-1/21	-4/21	0	0	**1**	0	0	37/7 : 16/21 ≈ 6,938
X1	3	1	0	0	0	**1**	0	0	0	0	0	
X2	33/7	0	1	5/21	1/21	-17/21	-1	0	0	**0**	0	33/7 : 5/21 = 19,8
X10	0.2	0	0	1	0	0	0	-1	0	0	**1**	1/5 : 1 = 0,2
Z												
W	192/35	0	**0**	**-37/21**	1/21	4/21	0	1	0	1	0	

базис	св.ч	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	св.ч/б
X6	67/15	0	0	0	**1/21**	-17/21	1	5/21	0	-1	-5/21	
X8	77/15	0	0	16/21	-1/21	-4/21	0	16/21	**1**	0	-15/21	
X1	3	1	0	0	0	**1**	0	0	0	0	0	
X2	14/3	0	1	0	1/21	-17/21	0	5/21	0	**0**	-5/21	
X3	0.2	0	0	1	0	0	0	-1	0	0	**1**	
W	77/15	0	**0**	**-37/21**	1/21	4/21	0	1	0	1	0	

Приравняем свободные переменные (x4, x5, x7, x9, x10) к нулю, тогда:
x1= 3, x2 =14/3, x3 = 1/5, x6 = 67/15, x8 = 77/15
W = 77/15

Шаг 5.

базис	св.ч	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	св.ч/б
X6	67/15	0	0	0	**1/21**	-17/21	1	5/21	0	-1	-5/21	67/15 : 5/21 = 18,76
X8	77/15	0	0	16/21	-1/21	-4/21	0	16/21	**1**	0	-15/21	77/15 : 16/21 ≈ 6,737
X1	3	1	0	0	0	**1**	0	0	0	0	0	
X2	14/3	0	1	0	1/21	-17/21	0	5/21	0	**0**	-5/21	14/3 : 5/21 = 19,6
X3	0.2	0	0	1	0	0	0	-1	0	0	**1**	
W	77/15	0	**0**	**-37/21**	1/21	4/21	0	1	0	1	0	

базис	св.ч	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	св.ч/б
X6	229/80	0	0	0	**1/16**	-3/4	1	0	-5/16	-1	0	
X7	539/80	0	0	0	-1/16	-1/4	0	1	**28/16**	0	-1	
X1	3	1	0	0	0	**1**	0	0	0	0	0	
X2	49/16	0	1	0	1/16	-3/4	0	0	-5/16	**0**	0	
X3	111/16	0	0	1	-1/16	-1/4	0	0	21/16	0	**1**	
W	0	0	**0**	**-37/21**	1/21	4/21	0	1	0	1	0	

Приравняем свободные переменные (x4, x5, x8, x9, x10) к нулю, тогда:
x1= 3, x2 =49/16, x3 = 111/16,   x6 =229/80,   x7 = 539/80
W = 0
Среди коэффициентов нет отрицательных. Следовательно, найдено наименьшее значение функции W.
Избавляемся от искусственных переменных x8, x9, x10 и получаем:

			+1/16*x4	-3/4*x5	+X6		= 229/80
			-1/16*x4	-1/4x5		+X7	= 539/80
X1				+X5			= 3
	X2		+1/16	-3/4x5			= 49/16
		X3	-1/16*x4	-1/4x5			= 111/16

Приравниваем свободные переменные нулю, тогда:
x1 = 3, x2 = 49/16, x3 = 111/16, x6 = 229/80, x7 = 539/80
Z = 13x1+29x2+19x3 = 259.625
Найден начальный базис, и значение функции, ему соответствующее.

Нахождение наименьшего значения функции Z.

базис	св.ч	X1	X2	X3	X4	X5	X6	X7	св.ч/б
X6	229/80	0	0	0	1/16	-3/4	1	0	229/80 :1/16 = 45.8
X7	539/80	0	0	0	-1/16	-1/4	0	1	
X1	3	1	0	0	0	1	0	0	
X2	49/16	0	1	0	1/16	-3/4	0	0	49/16: 1/16 = 49
X3	111/16	0	0	1	-1/16	-1/4	0	0	
Z	2077/8	0	0	0	-5/8	27/2	0	0	

базис	св.ч	X1	X2	X3	X4	X5	X6	X7	св.ч/б
X6	229/5	0	0	0	1	-12	16	0	
X7	48/5	0	0	0	0	-1	1	1	
X1	3	1	0	0	0	1	0	0	
X2	1/5	0	1	0	0	0	-1	0	
X3	49/5	0	0	1	0	-1	1	0	
Z	231	0	0	0	0	6	10	0	

Приравниваем свободные переменные x5, x6 к нулю, находим значение функции:

x1 = 3, x2 = 1/5, x3 = 49/5, x6 = 229/5, x7 = 48/5

Z = 13x1+29x2+19x3 = 231

Среди коэффициентов нет отрицательных. Следовательно, найдено наименьшее значение функции Z.

Глава 3. Решение транспортной задачи.

3.1 Теоретические основы.

Транспортная задача — это математическая задача по нахождению оптимального распределения поставок однородного «товара» (груза, вещества) между пунктами отправления и назначения при заданных, численно выраженных затратах (стоимостях, расходах) на перевозку. Общее решение изначально описано методами линейной алгебры, как для задачи линейного программирования специального вида. Транспортная задача может быть представлена на письме в виде прямоугольной таблицы.

Балансировка задачи. Если сумма запасов равна сумме потребностей, то транспортная задача называется *закрытой*. Если равенство не соблюдается, то задача называется *открытой*. Для решения транспортной задачи необходимо, чтобы она была приведена к закрытому виду.

Если это равенство не соблюдено, необходимо ввести фиктивного поставщика или фиктивного потребителя на недостающий или избыточный объем товара, которому нужно приписать нулевую цену доставки. Этот объем будет соответствовать недопоставке или, напротив, избытку товара на складе.

Решение транспортной задачи начинается с поиска допустимого начального решения (плана перевозок), чтобы все запасы поставщиков были распределены по потребителям. Допустимое начальное решение не обязательно оказывается оптимальным, а метод его нахождения может быть как простейшим (метод северо-западного угла или аналоги) или более сложным и приближенным к оптимальному решению (метод минимальных тарифов, метод Фогеля), или же вообще произвольным.

3.2 Пример решения транспортной задачи.

Условие задачи:

5. **Составьте план перевозок ресурсов от производителей к потребителям с минимальными затратами по условиям таблицы. Задачу решить методами наименьшего элемента и дельта-методом.**

Стоимость перевозки единицы ресурса, руб.					Производители ресурса	
					Объем производства ресурса	Наименование производителя
24	36	28	16	33	125	1
42	52	38	22	46	240	2
14	58	22	34	36	75	3
20	34	40	52	37	330	4
150	245	70	340	220	Объем производства ресурса	
1	2	3	4	5	Потребители ресурса	
Наименование потребителя						

Для решения задачи необходимо, чтобы суммарные запасы продукции у поставщиков равнялись суммарной потребности потребителей.

Проверим.

Запасы поставщиков: 125 + 240 + 75 + 330 = 770 ед.

Потребность потребителей: 150 + 245 + 70 + 340 + 220 = 1025 ед.

Нахватает 255 единиц продукции.

Введем в рассмотрение фиктивного производителя 5, с запасом продукции равным 255 единиц.

Стоимость доставки единицы продукции от производителя 5 ко всем потребителям примем равной нулю.

Стоимость перевозки					Производители	
					Объем пр-ва	наименование
24	36	28	16	33	125	1
42	52	38	22	46	240	2
14	58	22	34	36	75	3
20	34	40	52	37	330	4
0	0	0	0	0	255	5
150	245	70	340	220	Объем производства	
1	2	3	4	5	Потребители	
Потребители						

Построение опорного плана перевозок, методом наименьшего элемента.

Суть метода в том, что в матрице выбирается минимальная стоимость перевозки и назначается ей максимальный ресурс от производителя.

Матрица стоимостей

24	36	28	16	33	125
42	52	38	22	46	240
14	58	22	34	36	75
20	34	40	52	37	330
0	0	0	0	0	255
150	245	70	340	220	

Опорный план

После 1-го шага:

Матрица стоимостей

24	36	28	16	33	125
42	52	38	22	46	240
20	34	40	52	37	330
0	0	0	0	0	255
75	245	70	340	220	

Опорный план

75				

После 2-го шага:

Матрица стоимостей

42	52	38	22	46	240
20	34	40	52	37	330
0	0	0	0	0	255
150	245	70	215	220	

Опорный план

			125	
75				

После 3-го шага:

Матрица стоимостей

	52	38	22	46	240
	34	40	52	37	255
	0	0	0	0	255
	245	70	215	220	

Опорный план

			125	
75				
75				

После 4-го шага:

Матрица стоимостей

	52	38		46	25
	34	40		37	255
	0	0		0	255
	245	70		220	

Опорный план

			125	
			215	
75				
75				

После 5-го шага:

Матрица стоимостей

		38		46	25
		40		37	10
		0		0	255
		70		220	

Опорный план

			125	
			215	
75				
75	245			

После 6-го шага:

Матрица стоимостей

		38		46	25	
		0		0	255	
		70		210		

Опорный план

			125	
			215	
75				
75	245			10

После 7-го шага:

Матрица стоимостей

		0		0	255	
		45		210		

Опорный план

			125	
		25	215	
75				
75	245			10

После 8-го шага:

Матрица стоимостей

		0			45	
		45				

Опорный план

			125	
		25	215	
75				
75	245			10
				210

После 9-го шага:

Матрица стоимостей

Опорный план

			125	
		25	215	
75				
75	245			10
		45		210

Для полученного опорного плана посчитаем стоимость всех перевозок:

C = 125*16 + 25*38 + 215*22 + 75*14 + 75*20 + 245*34 + 10*37 + 45*0 + 210*0 = 18930

Дельта-метод.

Дельта-метод может использоваться только в том случае, если все значения элементов матрицы стоимости неотрицательные. Дельта-метод единственный, который при своей работе не требует опорного плана перевозок.

Выполним разреживание (зануление) матрицы:

Исходная матрица

24	36	28	16	33	125
42	52	38	22	46	240
14	58	22	34	36	75
20	34	40	52	37	330
150	245	70	340	220	

Преобразование строк

-16	8	20	12	0	17	125
-22	20	30	16	0	24	240
-14	0	44	8	20	22	75
-20	0	14	20	32	17	330
	150	245	70	340	220	

Преобразов. столбцов

0	-14	-8	0	-17	
8	6	4	0	0	125
20	16	8	0	7	240
0	30	0	20	5	75
0	0	12	32	0	330
150	245	70	340	220	

Определим дельта-оценку выполненных преобразований.

D = (150;245;70;340;220)(0;14;8;0;17) + (125;240;75;330)(16;22;14;20) =

= 245*14 + 70*8 + 220*17 + 125*16 + 240*22 + 75*14 + 330*20 = 22660

Выполним одну процедуру вредную (уменьшают количество нулей и уменьшают дельта-оценку), а вторую процедуру — полезную (увеличивает дельта-оценку):

Разреженная матрица

8	6	4	**0**	0	125
20	16	8	**0**	7	240
0	30	**0**	20	5	75
0	**0**	12	32	0	330
150	245	70	340	220	

4

Преобразование вредное

8	6	4	0	0	125
20	16	8	0	7	240
4	34	4	24	9	75
0	**0**	12	32	0	330
150	245	70	340	220	

Преобразование полезное -4

8	6	0	0	0	125
20	16	4	0	7	240
4	34	0	24	9	75
0	**0**	8	32	0	330
150	245	70	340	220	

Определим оптимальный план перевозок для нашего примера. На первом шаге назначаются максимально возможные перевозки для строк и столбцов, имеющих единственное нулевое значение

Шаг 1.

Матрица исходная

8	6	0	0	0	125
20	16	4	0	7	240
4	34	0	24	9	75
0	**0**	8	32	0	330
150	245	70	340	220	

План перевозок

					125
					240
		70			5
					330
150	245	70	340	220	

Шаг 2.

8	6		0	0	125
20	16		0	7	240
4	34		24	9	7
0	**0**		32	0	330
150	245		340	220	

					125
			240		0
		70			5
					330
150	245	0	100	220	

Шаг 3.

8	6		0	0	125
					0
4	34		24	9	5
0	**0**		32	0	330
150	245	0	100	220	

			100		25
			240		0
		70			5
					330
150	245	0	0	220	

Шаг 4.

8	6			0	25
					0
4	34			9	5
0	**0**			0	330
150	245	0	**0**	220	

			100	25	**0**
			240		0
		70			5
					330
150	245	0	0	195	

Шаг 5.

Т.к. очевидного выбора нет, то максимально возможный объем перевозки назначаем той клетке, которая имеет минимальную стоимость перевозок.

Матрица исходная

					0
					0
4	34			9	5
0	0			0	330
150	245	0	0	195	

План перевозок

			100	25	**0**
			240		0
5		70			**0**
					330
145	245	0	0	195	

Стоимость перевозок

24	36	28	16	33
42	52	38	22	46
14	58	22	34	36
20	34	40	52	37

Шаг 6.

Т.к. очевидного выбора нет, то максимально возможный объем перевозки назначаем той клетке, которая имеет минимальную стоимость перевозок.

Матрица исходная

					0
					0
					0
0	0			0	330
145	245	0	0	195	

План перевозок

			100	25	0
			240		0
5		70			0
145					185
0	245	0	0	195	

Стоимость перевозок

24	36	28	16	33
42	52	38	22	46
14	58	22	34	36
20	34	40	52	37

Шаг 7.

Т.к. очевидного выбора нет, то максимально возможный объем перевозки назначаем той клетке, которая имеет минимальную стоимость перевозок.

Матрица исходная

					0
					0
					0
	0			0	185
0	245	0	0	195	

План перевозок

			100	25	0
			240		0
5		70			0
145	185				0
0	60	0	0	195	

Стоимость перевозок

24	36	28	16	33
42	52	38	22	46
14	58	22	34	36
20	34	40	52	37

Ресурсы производителей исчерпаны, оптимальный план перевозок:

			100	25	125
			240		240
5		70			75
145	185				330
150	245	70	340	220	

Стоимость перевозок по оптимальному плану:
C = 100*16 + 25*33 + 240*22 + 5*14 + 70*22 + 145*20 + 185*34 = 18505

Глава 4. Программа для нахождения минимума функции симплекс-методом.

4.1 Для расчетов использованы библиотеки **numpy** и **scipy** языка Python3. Текст программы:

```
import numpy as np
from scipy.optimize import linprog

# коэффициенты целевой функции
z = list(map(float,input('введите коэффициенты целевой функции через
     пробел:').split()))
kn = int(input('введите количество ограничений-неравенств: '))

# коэффициенты из левой части неравенств (только "<=", неравенства вида ">="
# приводим к "<=")
aub = []
for i in range(kn):
    aub.append(list(map(float, input(f'введите коэффициенты из правой части
       неравенства {i+1} через пробел: ').split())))

# правая часть неравенств
bub = list(map(float, input(f'введите значения из левой части неравенств 1-{kn}
       через пробел: ').split()))

kr = int(input('введите количество ограничений-равенств: '))
aeq = [] # коэффициенты из левой части равенств
for i in range(kr):
    aeq.append(list(map(float, input(f'введите коэффициенты из левой части
       равенства {i+1} через пробел: ').split())))

# правая часть равенств
beq = list(map(float, input(f'введите значения из правой части равенств 1-{kr}
    через пробел: ').split()))


# готовим исходные данные
A_ub = np.array(aub, dtype=object)
b_ub = np.array(bub, dtype=object)
A_eq = np.array(aeq, dtype=object)
b_eq = np.array(beq, dtype=object)
c = np.array(z, dtype=object)


# Вызываем функцию с нашими аргументами:
res = linprog(c, A_ub, b_ub, A_eq, b_eq, bounds=(0, None))
```

```
# Смотрим результаты:
print('\n','Нахождение минимума целевой функции симплекс-методом.')

print('Исходные данные:')
print(f'коэффициенты целевой функции: {z}')
print('\n','Ограничения - неравенства:')
for id, i in enumerate(aub):
    print(f"{' '.join(map(str, i)):<15} {bub[id]}")

print('\n','Ограничения - равенства:')
for id, i in enumerate(aeq):
    print(f"{' '.join(map(str, i)):<15} {beq[id]}")

print('\n','Результаты вычислений:')
c = 0
for id, i in enumerate(res.x):
    print(f'x{id+1} = {round(i, 2)}')
```

4.2 Результат работы программы:

введите коэффициенты целевой функции через пробел: 13 29 19

введите количество ограничений-неравенств: 4

введите коэффициенты из правой части неравенства 1 через пробел: 17 21 5

введите коэффициенты из правой части неравенства 2 через пробел: 1 0 0

введите коэффициенты из правой части неравенства 3 через пробел: 0 -1 0

введите коэффициенты из правой части неравенства 4 через пробел: 0 0 -1

введите значения из левой части неравенств 1-4 через пробел: 150 3 -0.2 -0.2

введите количество ограничений-равенств: 1

введите коэффициенты из левой части равенства 1 через пробел: 1 1 1

введите значения из правой части равенств 1-1 через пробел: 13

коэффициенты целевой функции: [13.0, 29.0, 19.0]

Нахождение минимума целевой функции симплекс-методом.

Исходные данные:

Ограничения - неравенства:
17.0 21.0 5.0   150.0
1.0 0.0 0.0   3.0
0.0 -1.0 0.0   -0.2
0.0 0.0 -1.0   -0.2

Ограничения - равенства:
1.0 1.0 1.0   13.0

Результаты вычислений:
min(Z) = 231.0
x1 = 3.0
x2 = 0.2
x3 = 9.8

Заключение.

В данном курсовом проекте произведена минимизации функции Z = 13x1 + 29x2 + 19x3 графическим методом и симплекс методом.

В результате решения задачи минимизации обоими методами получено следующее значение функции: Z = 231. Данный оптимум достигается в точке (x1 = 3.0, x2 = 0.2, x3 = 9.8). Это же значение получено при решении задачи с помощью программы.

Также в данном проекте был найден оптимальный план перевозок для транспортной задачи.

Список использованной литературы

1. Агальцов, В. П. Математические методы в программировании / В.П. Агальцов. - М.: Форум, 2010.

2. Аттетков, А.В. Методы оптимизации: Учебное пособие / А.В. Аттетков, В.С. Зарубин, А.Н. Канатников. - М.: ИЦ РИОР, НИЦ Инфра-М, 2013. - 270 с.

3. Аттетков, А.В. Введение в методы оптимизации / А.В. Аттетков, В.С. Зарубин, А.Н. Канатников. - М.: Финансы и статистика, 2008. - 272 с.

4. Аттетков, А.В. Методы оптимизации: Учебное пособие / А.В. Аттетков, В.С. Зарубин, А.Н. Канатников. - М.: Риор, 2016. - 48 с.

5. Банди Б. Основы линейного программирования: пер. с англ. - М.: Радио и связь, 1989.

6. Банди Б. Методы оптимизации. Вводный курс. - М.: Мир, 1988.
Васильев. - М.: МЦНМО, 2011. - 433 с.

7. Васильев, Ф.П. Методы оптимизации в 2-х книгах. Кн.1 / Ф.П. Васильев. - М.: МЦНМО, 2011. - 619 с.

8. Васильев, Ф.П. Методы оптимизации в 2-х книгах кн.2 / Ф.П. Васильев. - М.: МЦНМО, 2011. - 433 с.

9. Васильев, Ф.П. Методы оптимизации в 2-х книгах кн.1 / Ф.П. Васильев. - М.: МЦНМО, 2011. - 619 с.

10. Гончаров, В.А. Методы оптимизации: Учебное пособие для ВУЗов / В.А. Гончаров. - Люберцы: Юрайт, 2016. - 191 с.

11. Горелик, В.А. Исследование операций и методы оптимизации: Учебник / В.А. Горелик. - М.: Academia, 2018. - 384 с.

12. Горелик, В.А. Исследование операций и методы оптимизации: Учебник / В.А. Горелик. - М.: Академия, 2014. - 128 с.

13. Елизаров, А.М. Обратные краевые задачи аэрогидродинамики. Теория и методы проектирования и оптимизации формы крыловых профилей / А.М. Елизаров. - Магадан: Магадан, 2011. - 436 с.

14. Емельянов, С.В. Методы нелинейного анализа в задачах управления и оптимизации / С.В. Емельянов, С.К. Коровин, Н.А. Бобылев. - М.: УРСС, 2002. - 120 с.
15. Зайцев, М.Г. Методы оптимизации управления для менеджеров: компьютерно-ориентированный подход / М.Г. Зайцев. - М.: Дело АНХ, 2016. - 312 с.
16. Зайцев, М.Г. Методы оптимизации управления и принятия решений: Примеры, задачи, кейсы: Учебное пособие / М.Г. Зайцев, С.Е. Варюхин; Рецензент С.Р. Филонович. - М.: ИД Дело РАНХиГС, 2011. - 640 с.
17. Зайцев, М.Г. Методы оптимизации управления для менеджеров: компьютерно-ориентированный подход. / М.Г. Зайцев. - М.: Дело АНХ, 2015. - 312 с.
18. Зайцев, М.Г. Методы оптимизации управления и принятия решений: примеры, задачи, кейсы / М.Г. Зайцев, С.Е. Варюхин. - М.: Дело АНХ, 2015. - 640 с.
19. Золотарев, А.А. Методы оптимизации распределительных процессов / А.А. Золотарев. - Вологда: Инфра-Инженерия, 2014. - 160 с.
20. Золоторев, А.А. Методы оптимизации распределительных процессов / А.А. Золоторев. - Вологда: Инфра-Инженерия, 2014. - 160 с.
21. Зубарев, Ю.М. Специальные методы оптимизации: Учебное пособие / Ю.М. Зубарев. - СПб.: Лань, 2014. - 384 с.
22. Измаилов, А.Ф. Численные методы оптимизации / А.Ф. Измаилов, М.В. Солодов. - М.: Физматлит, 2008. - 320 с.