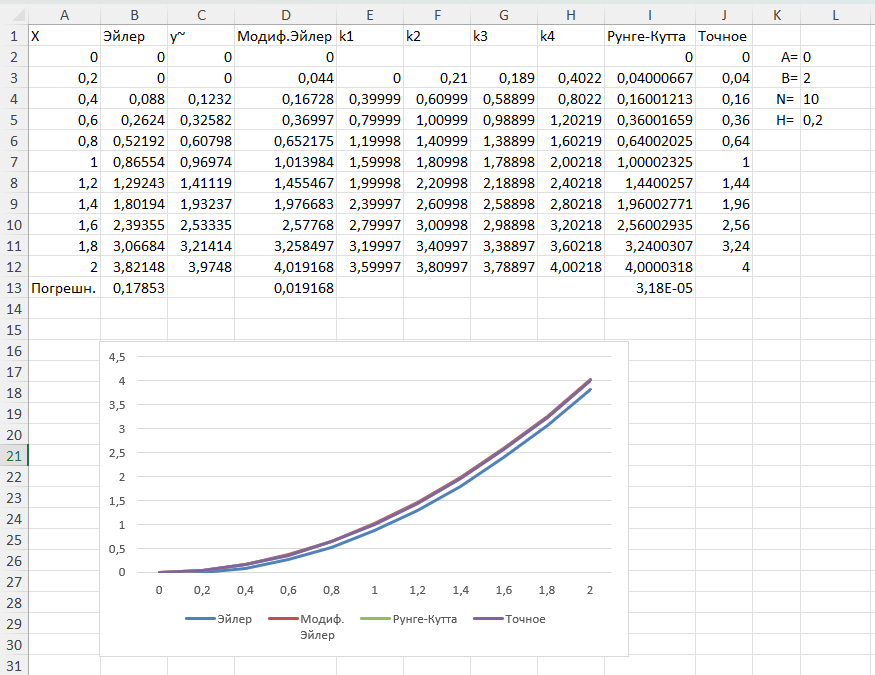
**Лабораторная работа №3-4**

**Цель работы:** определить оптимальное решение однокритериальных и многокритериальных задач в простейших случаях. Научиться сводить произвольную задачу линейного программирования к основной задаче линейного программирования. Решить задачу линейного программирования симплекс-методом.

**Часть 1**

**Задача 1**



**Часть 2**

**Задача 1**

а) введем дополнительные переменные x4, x5. Причем в первое неравенство введем переменную x4 со знаком плюс, а во второе – переменную x5 со знаком минус.

x1-2x2+x3+x4=4

x1+x2-3x3-x5=9

x1+3x2+2x3=10

x1≥0, x2≥0,x3≥0,x4≥0,x5≥0

Переведём max на min, домножив целевую функцию на -1

F=-2x1-x2+x3+0\*x4+0\*x5→min

что и даёт эквивалентную задачу в канонической форме

б) Всякое равенство в системе ограничений равносильно системе взаимопротивоположных неравенств, тогда получим:

x1-2x2+x3≥4

x1+x2-3x3≤9

x1+3x2+2x3≤10

-x1-3x2-2x3≤10

x1≥0, x2≥0,x3≥0,x4≥0,x5≥0

F=-2x1-x2+x3+0\*x4+0\*x5→min

**Задача 2**

1) Составим математическую модель задачи

Пусть x1-единицы готовой продукции вида A, а x2-единицы готовой продукции вида B.

Цель фабрики получить максимальную прибыль от реализации всей продукции видов A и B, тогда:

Целевая функция: F=16\*x1+19\*x2 –> max.

Система ограничений:

19x1+31x2≤1121

16x1+9x2≤706

19x1+x2≤1066

Условие неотрицательности:

x1 >= 0, x2 >= 0

2) Задачу приводим к каноническому виду

19x1+31x2+x3=1121

16x1+9x2-x4=706

19x1+x2+x5=1066

x1≥0, x2≥0,x3≥0,x4≥0,x5≥0

Переведём max на min, домножив целевую функцию на -1

F=-16\*x1-19\*x2+0\*x3+0\*x4+0\*x5 –> min

3) Базисные переменные выражаем через свободные

x3=1121-19x1-31x2

x4=706-16x1-9x2

x5=1066-19x1-x2

4) Записываем начальный план

X0=(0;0; 1121; 706; 1066)

5) Строим первую симплекс-таблицу

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Свободные переменные/Базистые переменные | -x1 | -x2 | Свободные члены | Симплексные отношения |
| x3 | 19 | 31 | 1121 | 1121/19=59 |
| x4 | 16 | 9 | 706 | 706/16=44,125 |
| x5 | 19 | 1 | 1066 | 1066/16=66,625 |
| F-строка | -16 | -19 | 0 |  |

6) Начальный план не оптимален, так как в строке есть отрицательные элементы

7) Улучшение плана. Строим вторую симплекс-таблицу, элементы которой пересчитываем по соответствующим формулам

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Свободные переменные/Базистые переменные | -x1 | -x3 | Свободные члены | Симплексные отношения |
| x2 | 19/31 | 1/31 | 59 | 59 |
| x4 | 325/31 | -9/31 | 44,125 | 11797/325 |
| x5 | 570/31 | -1/31 | 66,625 | 6385/114 |
| F-строка | -135/31 | 19/31 | 0 |  |

8) План не оптимален, так как в строке есть отрицательные элементы

9) Улучшение плана. Строим третью симплекс-таблицу, элементы которой пересчитываем по соответствующим формулам

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Свободные переменные/Базистые переменные | -x3 | -x4 | Свободные члены | Симплексные отношения |
| x2 | 16/325 | -19/325 | 4522/325 | 59 |
| x1 | -9/325 | 31/325 | 11797/325 | 11797/325 |
| x5 | 31/65 | -114/65 | 23557/65 | 6385/114 |
| F-строка | 32/65 | 27/65 | 0 |  |

10) План оптимален

Ответ: x1≈36, x2≈14

**Контрольные вопросы:**

**1**. Задачи линейного программирования (ЗЛП) — это задачи оптимизации, в которых целевая функция и ограничения заданы линейными уравнениями или неравенствами.

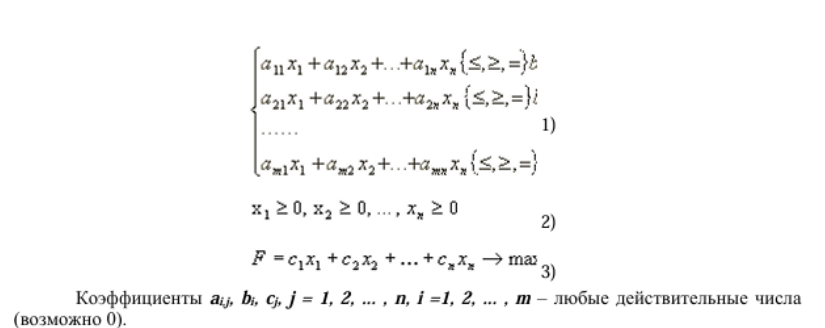
**2**. Линейное программирование – это направление математического программирования, изучающее методы решения экстремальных задач, которые характеризуются линейной зависимостью между переменным и линейным критерием.

**3**. Сущность линейного программирования состоит в нахождении точек наибольшего или наименьшего значения некоторой функции при определенном наборе ограничений, налагаемых на аргументы и образующих систему ограничений, которая имеет, как правило, бесконечное множество решений.

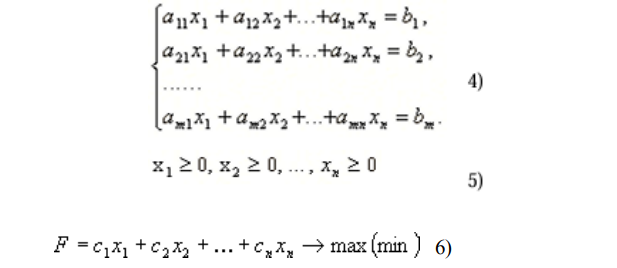
**4**. Каждая совокупность значений переменных (аргументов функции F), которые удовлетворяют системе ограничений, называется допустимым планом задачи линейного программирования.

**5**. Функция F, максимум или минимум которой определяется, называется целевой функцией задачи

**6**. Общая форма задачи линейного программирования формулируют следующим образом



**7**. В канонической форме задача является задачей на максимум (минимум) некоторой линейной функции F, ее система ограничений состоит только из равенств (уравнений). При этом переменные задачи х1, х2, ..., хn являются неотрицательными:



**8**. Правило приведения ЗЛП к каноническому виду:

1. Если в исходной задаче некоторое ограничение (например, первое) было неравенством, то оно преобразуется в равенство, введением в левую часть некоторой неотрицательной переменной, при чем в неравенства «≤» вводится дополнительная неотрицательная переменная со знаком «+»; в случаи неравенства «≥» - со знаком «-»

2. Если в исходной задаче некоторая переменная не подчинена условию неотрицательности, то ее заменяют (в целевой функции и во всех ограничениях) разностью неотрицательных переменных 16

3. Если в ограничениях правая часть отрицательна, то следует умножить это ограничение на (-1)

4. Наконец, если исходная задача была задачей на минимум, то введением новой целевой функции F1 = -F мы преобразуем нашу задачу на минимум функции F в задачу на максимум функции F1.

**9**. Идея симплекс-метода заключается в последовательном улучшении первоначального плана путем упорядоченного перехода от одного опорного плана к другому и завершается нахождением оптимального плана.

**10**. Условие оптимальности плана: если ЗЛП на максимум, то в F-строке не должно быть отрицательных элементов; если ЗЛП на минимум, то в F-строке не должно быть положительных элементов

**11**. Алгоритм решения:

1. Исходную задачу линейного программирования приводим к каноническому виду путем введения базисных переменных.

2. Базисные переменные выражаем через свободные переменные.

3. Строим начальный план, полагая свободные переменные равными нулю, тогда базисные переменные будут равны свободным членам.

4. Строим первую симплекс-таблицу.

5. Проверяем план на оптимальность. Если план не оптимален, то его улучшаем.

6. Улучшение плана.

**12**. Симплекс-отношения — это отношение свободных членов к положительным элементам разрешающего столбца в контексте симплекс-метода решения задач линейного программирования.