Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Саратовский государственный технический университет

имени Гагарина Ю.А.»

Институт электронной техники и приборостроения

Кафедра Информационная безопасность автоматизированных систем

Специальность 10.05.03 Информационная безопасность автоматизированных систем

Практическая работа № 6

Тема «Симплекс-метод»

|  |  |
| --- | --- |
|  | Выполнил: студент 3 курса  учебной группы с-ИБС 32  очной формы обучения  Солодилов В.В.  Проверил: доцент кафедры ИБС Кожанова Е.Р. |

Саратов 2022

Цель: формирование практических навыков решения задач симплекс-методом с их программной реализацией.

**ЗАДАНИЕ НА ПРАКТИЧЕСКУЮ РАБОТУ**

1. Изучить основные понятия симплекс-метода;
2. Выполнить расчёт по варианту № 17;
3. Разработать алгоритм решения задачи;
4. Написать программу для реализации алгоритма;
5. Выполнить контрольный тест работы программы и сравнить с результатами, полученными в п.2;
6. На основе полученных результатов сделать выводы.
7. **ОСНОВНЫЕ ПОНЯТИЯ**

**Постановка задачи линейного программирования**

Пусть S – система линейных ограничений, т.е. линейных уравнений или нестрогих линейных неравенств с n переменными x1, x2,…, xn, а Z(Х) – линейная функция вида

*Z(Х)=c1x1+c2x2+…+cnxn+с0.* (1)

Как правило, система S содержит условия неотрицательности всех (или части) переменных

, (2)

что вытекает из реального экономического смысла чисел *x1,x2,…,xn*.

Ограничения вида (2) называются тривиальными.

Необходимо найти максимум (или минимум) линейной функции

при условиях S.

Функцию Z называют *целевой функцией*, *критерием оптимальности или линейной формой*.

Совокупность значений неизвестных *=(x1,x2,…,xn*), удовлетворяющих условиям задачи (1)-(2), называется *допустимым решением* (или *допустимым планом*). Решение (план) \*=(*x1\*,x2\*,…,xn\**) называется *оптимальным*, если оно обеспечивает максимальное (минимальное) значение целевой функции (1).

Наиболее часто встречаются следующие две разновидности задачи ЛП:

1. Каноническая задача ЛП. Для такой задачи система S, помимо тривиальных ограничений (2) включает в себя только уравнения.

2. Стандартная задача ЛП. Для этой задачи S состоит только из неравенств, в число которых входят и тривиальные ограничения вида (2).

Указанные две разновидности сводятся одна к другой. Чтобы свести стандартную задачу ЛП к канонической, нужно в неравенства вида «» добавить к левой части дополнительные неотрицательные переменные, а в неравенства вида «» - вычесть из его левой части дополнительные неотрицательные переменные. Например, неравенства

*ai1x1+ai2x2+…+ainxnbi*, *i*=

преобразуются в равенства путем добавления к левой части соответствующих дополнительных неизвестных *yi (i*=):

*ai1x1+ai2x2+…+ainxn+ yi =bi* , *yi0, i=*,

а неравенства

*ai1x1+ai2x2+…+ainxn bi , i=,*

после вычитания дополнительных неизвестных преобразуются в равенства вида

*ai1x1+ai2x2+…+ainxn - yi =bi , yi0, i*=,

Отметим, что дополнительные переменные *yi* обычно называют *балансовыми.*

**Симплекс-метод решения задачи линейного программирования**

Рассмотрим задачу ЛП в канонической форме, где система ограничений задана совместной системой линейных уравнений

(3)

Среди неотрицательных решений системы (3) нужно найти такое, которое минимизирует линейную функцию (1)

*Z(Х)=c1x1+c2x2+…+cnxn+с0*

Пусть можно выразить *x1,x2,…,xr (rm)* через остальные переменные:

(4)

где , т.е. систему (3) можно заменить эквивалентной ей системой (4).

Переменные *x1,x2,…,xr*  называются *базисными*, а весь набор

{ *x1,x2,…,xr* }

– *базисом*, остальные переменные называются свободными, система ограничений (4) называется системой, приведенной к единому базису.

Подставляя в линейную форму (1) вместо базисных переменных их выражения через свободные из системы (4), получим

(5)

Теперь, полагая все свободные переменные равными нулю, найдем значения базисных переменных: Таким образом, решение () системы (4) является допустимым и называется *базисным*.

Для полученного базисного решения значение линейной формы .

Равенство (5) называют приведенным к свободным переменным выражением для функции *Z(Х*), а коэффициенты - оценками (индексами) соответствующих свободных переменных .

Решение ЗЛП симплекс-методом с помощью симплексных таблиц состоит в следующем.

1.Выбираем разрешающий столбец из условия: оценка <0 и хотя бы один элемент в столбце с номером *р* больше 0, т.е. >0.

2. Выбираем *q* – ю разрешающую строку из условия:

для >0.

3.Пересчитываем элементы разрешающей строки по формуле:

.

4.Вычисляем элементы всех остальных строк (при ) по формуле (правилу прямоугольника):

.

При этом используем основную теорему симплекс-метода:

Теорема. Если после выполнения очередной итерации:

1. найдется хотя бы одна отрицательная оценка и в каждом столбце с такой оценкой окажется хотя бы один положительный элемент, т.е. для некоторых *k* и для тех же *k* и некоторого *i*, то можно улучшить решение, выполнив следующую итерацию;
2. найдется хотя бы одна отрицательная оценка, столбец которой не содержит положительных элементов, т.е. для какого-то *k* и всех *i*, то функция *Z* неограничена в области допустимых значений ();
3. все оценки окажутся неотрицательными, т.е. для всех *k*, то достигнуто оптимальное решение.

Пример.

Пусть предприятием выпускается продукция четырех видов П1, П2, П3, П4 с использованием для этого ресурсов, виды и нормы расхода по которым, а также прибыль, получаемая от их реализации, приведены в таблице 1.

Необходимо составить оптимальный план производства по критерию максимума прибыли.

Таблица 1.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Элемент модели | Вид продукции | | | | Величина  ресурса |
| П1 | П2 | П3 | П4 |
| Ресурсы: |  | | | |  |
| Трудовые S1 | 1 | 1 | 1 | 1 | 16 |
| Сырье S2 | 6 | 5 | 4 | 3 | 110 |
| Оборудование S3 | 4 | 6 | 10 | 13 |  |
| Прибыль с единицы продукции | 60 | 70 | 120 | 130 | 100 |
| План | х1 | х2 | х3 | х4 |  |

Решение: Математическую модель данной задачи можно записать так:

*Z(Х)=60x1+70x2+120x3+130x4 max*;

(6)

В математическую модель задачи (6) введем дополнительные переменные *y1, y2, y3* и запишем ограничения в виде уравнений:

*- Z(x)=-60x1-70x2-120x3-130x4 min;*

(7)

Задача (7) записана в каноническом виде и ее можно представить симплекс-таблицей 2.

Таблица 2.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Базисные переменные | Свободные члены | Вид продукции | | | | | | |
| *y1* | *y2* | *y3* | *х1* | *х2* | *х3* | *х4* |
| *y1* | 16 | 1 | 0 | 0 | 1 | 1 | 1 | 1 |
| *y2* | 110 | 0 | 1 | 0 | 6 | 5 | 4 | 3 |
| *y3* | 100 | 0 | 0 | 1 | 4 | 6 | 10 | 13 |
| *+Z(Х)* | 0 | 0 | 0 | 0 | -60 | -70 | -120 | -130 |

В этой таблице свободные переменные *х1, х2, х3, х4* по условию равны нулю, а базисные переменные *y1, y2, y3*, а также целевая функция *(+Z(Х)*) равны свободным членам, т.е. *y1=16, y2=110, y3=100, +Z(Х)=0*.

Выясняем, имеются ли в последней строке (индексной) отрицательные оценки. Таких чисел четыре: -60,-70,-120,-130. Берем, например, -60 и просматриваем столбец для *х1*, в этом столбце имеем три положительных элемента 1,6,4. Делим на эти числа соответствующие свободные члены: 16, , ; из полученных частных наименьшее есть 16, следовательно, разрешающим является элемент 1, стоящий на пересечении строки для *y1* и столбца для *х1*. Выделим эту строку и этот столбец рамками. На следующей итерации переменную *y1* нужно вывести из базисных, а переменную *х1* ввести в базисные. Новый базис состоит из . Для составления следующей таблицы умножим выделенную строку таблицы 3 на 1 и полученную строку пишем на месте прежней (в данном случае эта строка остается прежней, так как разрешающий элемент равен 1). К каждой из остальных строк прибавляем вновь полученную, умноженную на такое число, чтобы в клетках столбца *х1* появились нули и пишем преобразованные строки на месте прежних. Этим завершается первая итерация и в результате придем к таблице 3. Теперь все рассуждения повторяем применительно к таблице 4, т.е. выполняем вторую итерацию. Новый разрешающий элемент, находящийся на пересечении строки для *y3* и столбца для *х3*, есть 6.

Таблица 3.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Базисные переменные | Свободные члены | Вид продукции | | | | | | |
| *y1* | *y2* | *y3* | *х1* | *х2* | *х3* | *х4* |
| *х1* | 16 | 1 | 0 | 0 | 1 | 1 | 1 | 1 |
| *y2* | 14 | -6 | 1 | 0 | 0 | -1 | -2 | -3 |
| *y3* | 36 | -4 | 0 | 1 | 0 | 2 | 6 | 9 |
| *+Z(x)* | 960 | 60 | 0 | 0 | 0 | -10 | -60 | -70 |

Для составления новой таблицы умножим выделенную строку на, и полученную строку запишем на месте прежней; к каждой из остальных строк прибавим вновь полученную, умноженную на такое число, чтобы в клетках для столбца *х3* были нули и запишем преобразованные строки на месте прежних. Приходим к таблице 4.

Таблица 4.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Базисные переменные | Свободные члены | Вид продукции | | | | | | |
| y1 | y2 | y3 | х1 | х2 | х3 | х4 |
| х1 | 10 |  | 0 |  | 1 |  | 0 |  |
| y2 | 26 |  | 1 |  | 0 |  | 0 | 0 |
| х3 | 6 |  | 0 |  | 0 |  | 1 |  |
| +Z(Х) | 1320 | 20 | 0 | 10 | 0 | 10 | 0 | 20 |

Из симплексной таблицы 4 следует, что в столбце свободных членов все элементы положительные, следовательно, решение является допустимым. В строке целевой функции все элементы неотрицательные. Следовательно, это решение является оптимальным при максимизации целевой функции *Z(Х).*

При этом оптимальным планом является , а целевая функция *Z(Х)*=1320.

Если оптимальное решение не единственное, то говорят, что задача имеет альтернативный оптимум. Признаком наличия альтернативного оптимума является равенство нулю хотя бы одной из оценок свободных переменных. Можно получить еще одно оптимальное решение *XОПТ2* , выполнив еще одну итерацию, при которой свободная переменная с нулевой оценкой будет введена в состав базисных.

**Двойственные задачи**

Каждой задаче ЛП можно сопоставить определенным образом связанную с ней другую задачу, которая называется двойственной по отношению к исходной.

Двойственную задачу составляем, используя следующие приемы.

1.Каждому *i* – му ограничению соответствует переменная двойственной задачи, называемая двойственной переменной и обозначаемая *zi*.

2.Каждой переменной исходной задачи соответствует ограничение двойственной задачи; так, если имеется четыре переменных х1, х2, х3, х4, то двойственная задача должна иметь четыре ограничения.

3.Матрица коэффициентов при двойственных переменных в ограничениях двойственной задачи является транспонированной матрицей коэффициентов при переменных, стоящих в ограничениях.

4.Если в исходной задаче ограничения имеют знаки неравенства типа «» (меньше или равно), то в двойственной задаче они меняются на противоположные типа «» (больше или равно).

5.Правые части ограничений в двойственной задаче равняются коэффициентам при переменных в целевой функции исходной задачи, а коэффициенты при двойственных переменных в целевой функции двойственной задачи равняются правым частям ограничений исходной задачи.

6.Максимизация целевой функции исходной задачи заменяется минимизацией целевой функции двойственной задачи.

В общем виде исходную задачу ЛП и соответствующую ей двойственную задачу ЛП можно записать следующим образом:

(8)

Важным свойством двойственной задачи является то, что .

Двойственная переменнаявыступает коэффициентом при и, следовательно, определяет зависимость целевой функции от изменения ресурса на единицу. Таким образом, двойственная переменная оценивает влияние изменения каждого вида ресурса на целевую функцию. В связи с этим двойственные переменные часто называют двойственными оценками, при этом существенно, что для нахождения двойственных оценок не требуется решать двойственную задачу.

Рассмотрим пример и симплексную таблицу 4. Значения двойственных оценок уже получены в этой симплексной таблице. Узнать эти значения можно следующим образом: если некоторый *i*-тый ресурс используется не полностью, то дополнительная переменная в ограничении для данного ресурса больше нуля. В данном примере таким ресурсом выступает сырье, так как и его резерв у2 = 26 . Следовательно, для второго ограничения . Таким образом, если по определенному ресурсу имеется резерв, то дополнительная переменная является базисной, а двойственная оценка такой переменной равна нулю. В рассматриваемом примере трудовые ресурсы и оборудование использованы полностью, поэтому их допол­нительные переменныеи равны нулю (таблица 4). Если ресурс используется полностью, то его изменение повлияет на объем выпускаемой продукции и в конечном счете на целевую функцию: целевая функция увеличивается или уменьшается на величину двойственной оценки. Значение двойствен­ной оценки находится в симплекс-таблице 4 на пересечении строки целевой функции со столбцом данной дополнительной переменной. Так, для трудовых ресурсов при = 0 двойственная оценка , а для оборудования при =0 двойственная оценка . Значения дополнительных переменных и двойственных оценок из таблицы 4 перенесем для наглядности в таблицу 5, откуда видно, что при росте трудовых ресурсов на единицу целевая функция возрастает на 20 единиц, достигнув значения = 1340 т.е. (1320 + 20 = 1340), а при их уменьшении =1300 т. е. (1320 - 20 = 1300). Аналогично при изменении ресурса второго вида, т.е. оборудова­ния, на единицу вызовет увеличение целевой функции на 10 (= 1320+10 = 1330), а при его снижении на единицу = 1310 ( = 1320-10 = 1310).

Таблица 5.

|  |  |  |
| --- | --- | --- |
| Ресурсы | Дополнительная переменная | Двойственная оценка |
| Трудовые |  |  |
| Сырье |  |  |
| Оборудование |  |  |

**Анализ ситуации отклонения ресурсов от первоначально запланированных**

При оперативном управлении особенно важно принимать верные решения в ситуациях отклонения ресурсов от первоначально запланированных. Появившееся отклонение нужно включать в модель задачи. Начнем с отклонений для сырья. Допустим, отклонение в поставке сырья составляет .

Тогда в математической модели ограничение для сырья имеет вид:

*6х1+5х2+4х3+3х4=110*+.

Данное ограничение в системе уравнений приобретает следующий вид:

*y2=110+-(6х1+5х2+4х3+3х4),*

а в симплекс-таблице отразится это изменение (таблица 6). После нахождения решения с учетом получим симплекс-таблицу 7. Согласно ей в оптимальном решении вошло только в свободный член для *y2=26+*.

Таблица 6.

|  |  |  |
| --- | --- | --- |
| Базисные переменные | Свободные члены | y1, y2, y3, x1, x2, x3, x4 |
| y1 | 16 |  |
| y2 | 110+ |  |
| y3 | 100 |  |
| Z | 0 |  |

Таблица 7.

|  |  |  |
| --- | --- | --- |
| Базисные переменные | Свободные члены | y1, y2 |
| х1 | 10 |  |
| y2 | 26+ |  |
| х3 | 6 |  |
| Z | 1320 |  |

Так как оптимальное решение должно быть прежде всего допустимым, то в симплекс-таблице все свободные члены должны быть неотрицательными. В данном случае *y2=26+*0, т.е. -26. Следовательно, уменьшение сырья на 26 единиц, т.е. на 23,6% от первоначального объема в 110 единиц не скажется на выполнении плана.

При анализе влияния отклонения трудовых ресурсов первое ограничение в модели запишется так:

*х1+х2+х3+х4 16+.*

Соответственно дополнительную переменную определим как

*y1=16+- (х1+х2+х3+х4)*

и аналогично изменится свободный член в начальной симплекс-таблице.

Симплекс-таблица 4 приобретает вид таблицы 8, где в ней значения в столбце свободных членов те же, что и в таблице 4, к которым прибавлены элементы столбца , умноженные на .

Таблица 8.

|  |  |  |
| --- | --- | --- |
| Базисные переменные | Свободные члены |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Анализ таблицы 6 показывает, что можно сделать следующие вы­воды:

1. Требование о сохранении допустимого решения имеет вид:

, , .

2. Решая систему этих неравенств определяем, что диапазон изменения , составляет -6< <3,55.

3. Переходя от приращений ресурсов к их предельным значениям, нахо­дим, что окончательно диапазон составляет .

4. Найденные пределы показывают, каковы могут быть колебания трудовых ресурсов, чтобы структура оптимального плана оставалась стабильной. А это означает, что при изменении трудовых ресурсов в найденных пределах оптимальным является выпуск той же самой продукции (и), но в объёме, составляющем при этом целевая функция составит .

Аналогично можно показать, что структура оптимального плана сохранится, если диапазон изменений составит 64< <160.

Анализ показывает, что двойственные оценки сохраняют свое значение в том же самом интервале изменения ресурсов, при котором сохраняется структура оптимального плана; следовательно в данном случае для трудовых ресурсов двойственные оценки справедливы при изменении ресурсов в пределах 10 <<19.55; аналогично анализу влияния ресурсов, можно также установить степень влияния коэффици­ентов в целевой функции , а также норм расхода ресурсов или произ­водительности труда и оборудования .

1. **Расчет по заданному варианту № 16**

*Задание 1.*

Для реализации трех групп товаров предприятие располагает тремя видами ограниченных материально-денежных ресурсов в количестве ,, единиц. При этом для продажи первой группы товаров на единицу товарооборота расходуется ресурса первого вида в количестве единиц, ресурса второго вида единиц, ресурса третьего вида единиц. Для продажи второй и третьей групп товаров на единицу товарооборота расходуется соответственно ресурса первого вида и единиц, ресурса второго вида в количестве и единиц, ресурса третьего вида в количестве и единиц. Прибыль от продажи трех групп товаров на единицу товарооборота составляет соответственно , и (руб.).

Требуется определить плановый объем и структуру товарооборота так, чтобы прибыль торгового предприятия была максимальной.

Решим прямую задачу линейного программирования симплекс-методом, с использованием симплексной таблицы в программе для работы с электронными таблицами Excel.

Для построения первого опорного плана систему неравенств приведём к системе уравнений путём введения дополнительных переменных (переход к канонической форме). В первом неравенстве вводим базисную переменную , Во 2-ом неравенстве , в 3-ем неравенстве . Тут базисными переменными мы будем называть те, которые входят только в одно уравнение системы ограничений и притом с единичным коэффициентом. Решим систему уравнений относительно базисных переменных , , . Полагая, что свободные переменные равны 0, получим первый опорный план:

Базисное решение будет являться допустимым, если оно неотрицательно.

Таблица 9.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **X** | **x₁** | **x₂** | **x₃** | **x₄** | **x₅** | **x₆** | **x\*** |
| **x₄** | 5 | 8 | 4 | 1 | 0 | 0 | 400 |
| **x₅** | 5 | 5 | 6 | 0 | 1 | 0 | 300 |
| **x₆** | 10 | 2 | 5 | 0 | 0 | 1 | 200 |
| **Z** | -4 | -3 | -2 | 0 | 0 | 0 | 0 |

Текущий опорный план не оптимален, так как в индексной строке находятся отрицательные коэффициенты. В качестве ведущего выберем столбец, соответствующий переменной , так как это наибольший коэффициент по модулю. Далее вычислим частные от деления элементов столбца и выбранного столбца и выберем из них наименьшее. Следовательно, вторая строка является ведущей. Разрешающий элемент равен 6 и находится на пересечении ведущего столбца и ведущей строки.

Далее ведём пересчёт симплекс-таблицы. Вместо переменной в план войдёт переменная . Строка, соответствующая переменной в плане , получена в результате деления всех элементов строки плана на разрешающий элемент (РЭ) 6. На месте разрешающего элемента получаем 1. В остальных клетках столбца записываем нули. Таким образом, в новом плане заполнены строка и столбец . Все остальные элементы нового плана , включая элементы индексной строки, определяются по правилу прямоугольника.

Для этого выбираем из старого плана четыре числа, которые расположены в вершинах прямоугольника и всегда включают РЭ. Новый элемент (НЭ) = старый элемент (СЭ) – произведение элементов старого плана (СТЭ) делённое на РЭ. СТЭ образуют прямоугольник с элементами СТЭ и РЭ. Получаем новую симплекс-таблицу:

Таблица 10.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **X** | **x₁** | **x₂** | **x₃** | **x₄** | **x₅** | **x₆** | **x\*** |
| **x₄** | 0 | 7 | 1,5 | 1 | 0 | -0,5 | 300 |
| **x₁** | 0 | 4 | 3,5 | 0 | 1 | -0,5 | 200 |
| **x₆** | 1 | 0,2 | 0,5 | 0 | 0 | 0,1 | 20 |
| **Z** | 0 | -2,2 | 0 | 0 | 0 | 0,4 | 80 |

Руководствуясь предыдущими вычислениями, продолжаем процесс вычислений, пока в индексной строке не останется отрицательных коэффициентов.

Таблица 11.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **X** | **x₁** | **x₂** | **x₃** | **x₄** | **x₅** | **x₆** | **x\*** |
| **x₄** | 0 | 1 | 0,2143 | 0,143 | 0 | -0,071 | 42,86 |
| **x₁** | 0 | 0 | 2,6429 | -0,571 | 1 | -0,214 | 28,57 |
| **x₂** | 1 | 0 | 0,4571 | -0,029 | 0 | 0,114 | 11,43 |
| **Z** | 0 | 0 | 0,4714 | 0,314 | 0 | 0,243 | 174,3 |

Среди значений индексной строки нет отрицательных коэффициентов. Поэтому эта таблица определяет оптимальный план задачи. Оптимальный план можно записать так:

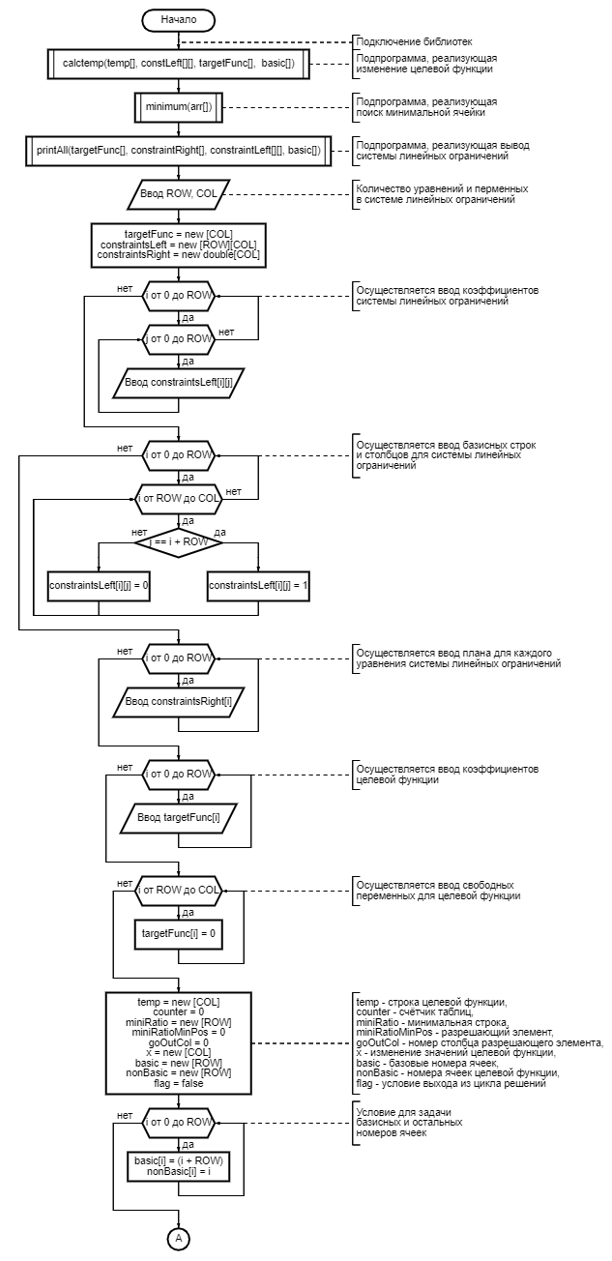
Проведём анализ оптимального плана. Значение в столбце означает, что использование – выгодно. Значение 0 в столбце означает, что использование – выгодно. Значение 0,4714 в столбце означает, что использование – выгодно. Значения теневых цен (двойственных оценок):

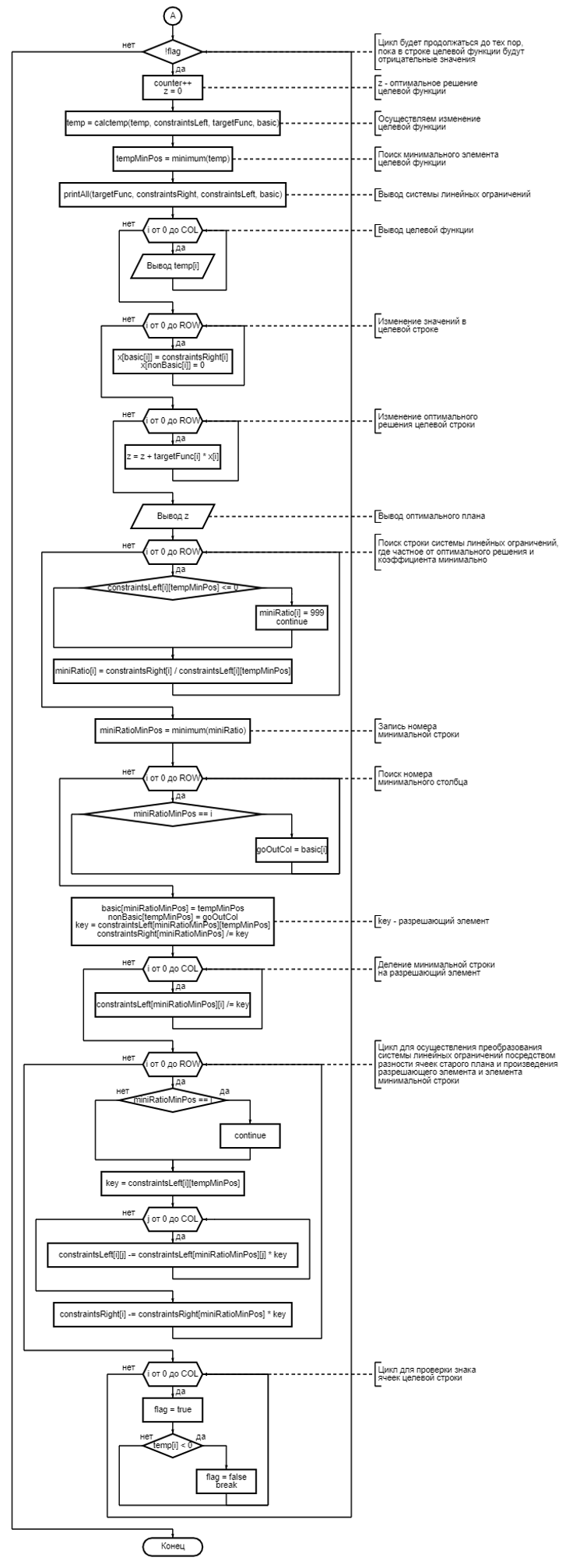
1. **Алгоритм решения задачи**

Метод стрельбы, описанный в данной практической работе, осуществляется согласно следующему алгоритму:

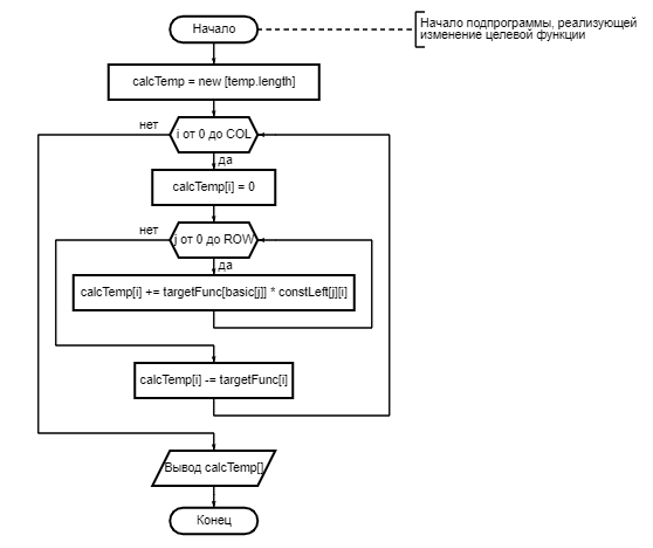
1. Задаём начальные значения: количество уравнений и переменных в системе линейных ограничений; коэффициенты системы линейных ограничений; допустимый план для каждого уравнения системы линейных ограничений; коэффициенты при целевой функции.
2. Согласно второму пункту основных понятий практической работы составляем симплекс-таблицу. Выбираем ведущий столбец, соответствующий наибольшему коэффициенту по модулю в целевой функции. Вычисляем частное элементов столбца и элементов ведущего столбца. Получаем разрешающий элемент, соответствующий минимальному частному.
3. Далее ведём пересчёт симплекс-таблицы. Далее осуществляется преобразование всей таблицы с помощью ключа, являющегося центральной ячейкой пересечения минимальной строки и минимального столбца по целевой функции (разрешающий элемент) посредством преобразования минимального столбца в столбец нулевых значений с единичной минимальной ячейкой. Минимальную строку делим на разрешающий элемент, а остальные строки складываем с отрицательным произведением элемента данной строки на предыдущем плане и минимальным элементом строки на текущем плане.
4. Совершаем весь этот процесс до тех пор, пока в строке целевой функции есть отрицательные коэффициенты. После того, как в ней останутся только положительные элементы, мы можем получить в столбце искомый оптимальный план.

Для основной программы была составлена блок-схема:

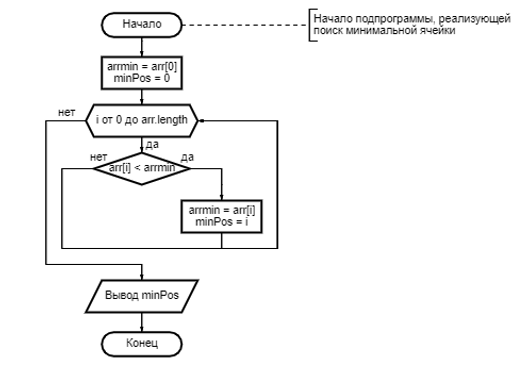




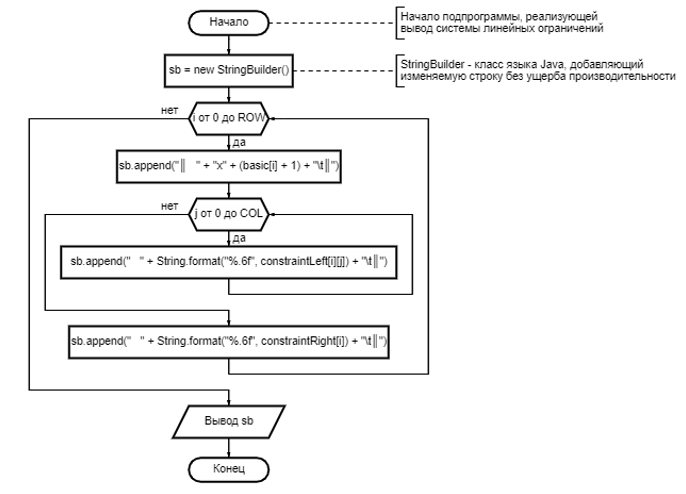
Для подпрограммы calcTemp также была составлена блок-схема:



Для подпрограммы minimum также была составлена блок-схема:



Для подпрограммы printAll также была составлена блок-схема:

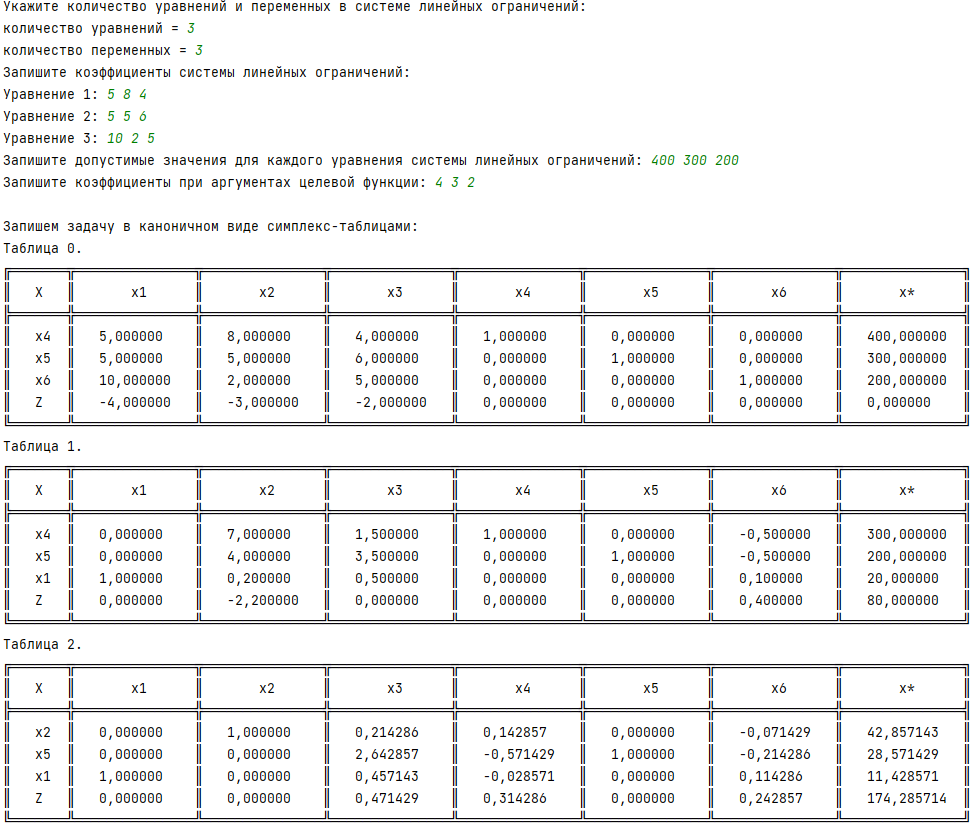


1. **Программа для реализации алгоритма**

Данная программа была реализована на языке Java:

import java.util.Scanner;  
  
public class Main {  
  
 private static int *ROW*;  
  
 private static int *COL*;  
  
 private static final Scanner *scanner* = new Scanner(System.*in*);  
  
 private static double[] calcTemp(double[] temp, double[][] constLeft, double[] targetFunc, int[] basic) { *// изменение целевой функции* double[] calcTemp = new double[temp.length];  
 for (int i = 0; i < *COL*; i++) {  
 calcTemp[i] = 0;  
 for (int j = 0; j < *ROW*; j++) {  
 calcTemp[i] += targetFunc[basic[j]] \* constLeft[j][i];  
 }  
 calcTemp[i] -= targetFunc[i];  
 }  
 return calcTemp;  
 }  
  
 private static int minimum(double[] arr) { *// поиск минимальной ячейки* double arrMin = arr[0];  
 int minPos = 0;  
  
 for (int i = 0; i < arr.length; i++) {  
 if (arr[i] < arrMin) {  
 arrMin = arr[i];  
 minPos = i;  
 }  
 }  
  
 return minPos;  
 }  
  
 private static void printAll(double[] constraintRight, double[][] constraintLeft, int[] basic) { *// вывод системы линейных ограничений* System.*out*.println(  
 "╔═══════╦═══════════════╦═══════════════╦═══════════════╦═══════════════╦═══════════════╦═══════════════╦═══════════════╗");  
 System.*out*.println(  
 "║ X" + "\t║ " + "\tx1" + "\t\t║ " + "\tx2" + "\t\t║ " + "\tx3" + "\t\t║ " + "\tx4" + "\t\t║ "  
 +  
 "\tx5" + "\t\t║ " + "\tx6" + "\t\t║ " + "\tx\*" + "\t\t║ ");  
 System.*out*.println(  
 "╠═══════╬═══════════════╬═══════════════╬═══════════════╬═══════════════╬═══════════════╬═══════════════╬═══════════════╣");  
 StringBuilder sb = new StringBuilder();  
  
 for (int i = 0; i < *ROW*; i++) {  
 sb.append("║ " + "x").append(basic[i] + 1).append("\t║");  
  
 for (int j = 0; j < *COL*; j++) {  
 sb.append(" ").append(String.*format*("%.6f", constraintLeft[i][j])).append("\t║");  
 }  
  
 sb.append(" ").append(String.*format*("%.6f", constraintRight[i])).append("\t║");  
 sb.append("\n");  
 }  
  
 System.*out*.print(sb);  
 }  
  
 public static void main(String[] args) {  
 System.*out*.println("Укажите количество уравнений и переменных в системе линейных ограничений: ");  
  
 System.*out*.print("количество уравнений = ");  
 *ROW* = *scanner*.nextInt();  
  
 System.*out*.print("количество переменных = ");  
 *COL* = *scanner*.nextInt() + *ROW*;  
  
 double[] targetFunc = new double[*COL*];  
 double[][] constraintsLeft = new double[*ROW*][*COL*];  
 double[] constraintsRight = new double[*COL*];  
  
 System.*out*.println("Запишите коэффициенты системы линейных ограничений: ");  
 for (int i = 0; i < *ROW*; i++) {  
 System.*out*.print("Уравнение " + (i + 1) + ": ");  
 for (int j = 0; j < *ROW*; j++) {  
 constraintsLeft[i][j] = *scanner*.nextDouble();  
 }  
 }  
  
 for (int i = 0; i < *ROW*; i++) {  
 for (int j = *ROW*; j < *COL*; j++) {  
 if (j == i + *ROW*) {  
 constraintsLeft[i][j] = 1;  
 } else {  
 constraintsLeft[i][j] = 0;  
 }  
 }  
 }  
  
 System.*out*.print("Запишите допустимые значения для каждого уравнения системы линейных ограничений: ");  
 for (int i = 0; i < *ROW*; i++) {  
 constraintsRight[i] = *scanner*.nextDouble();  
 }  
  
 System.*out*.print("Запишите коэффициенты при аргументах целевой функции: ");  
 for (int i = 0; i < *ROW*; i++) {  
 targetFunc[i] = *scanner*.nextDouble();  
 }  
  
 for (int i = *ROW*; i < *COL*; i++) {  
 targetFunc[i] = 0;  
 }  
  
 *scanner*.close();  
  
 double[] temp = new double[*COL*]; *// строка целевой функции* double[] miniRatio = new double[*ROW*]; *// строка системы линейных ограничений с минимальным частным допустимого плана и значения коэффициента* double[] x = new double[*COL*]; *// изменение значений целевой функции* double key; *// значение в этой ячейке* double z; *// поиск оптимального решения целевой функции* int[] basic = new int[*ROW*]; *// базовые номера ячеек* int[] nonBasic = new int[*ROW*]; *// номера ячеек целевой функции* int goOutCol = 0; *// номер этого столбца* int counter = 0; *// счётчик таблиц* int tempMinPos; *// минимальная позиция целевой функции* int miniRatioMinPos; *// номер ячейки пересечения минимальной строки и столбца* boolean flag = false; *// условие выхода из цикла решений* for (int i = 0; i < *ROW*; i++) {  
 basic[i] = (i + *ROW*);  
 nonBasic[i] = i;  
 }  
  
 System.*out*.println("\nЗапишем задачу в каноничном виде симплекс-таблицами: ");  
 while (!flag) { *// цикл завершается при достижении оптимального решения и отсутствии отрицательных коэффициентов в строке целевой функции* System.*out*.println("Таблица " + counter + ".");  
  
 counter++;  
 z = 0;  
  
 temp = *calcTemp*(temp, constraintsLeft, targetFunc, basic); *// изменение целевой функции* tempMinPos = *minimum*(temp); *// поиск минимального элемента целевой функции  
 printAll*(constraintsRight, constraintsLeft, basic); *// вывод системы линейных ограничений* System.*out*.print("║ " + "Z" + "\t║");  
 for (int i = 0; i < *COL*; i++) {  
 System.*out*.print(" " + String.*format*("%.6f", temp[i]) + "\t║");  
 }  
  
 System.*out*.print(" ");  
 for (int i = 0; i < *ROW*; i++) { *// изменение значений в целевой строке* x[basic[i]] = constraintsRight[i];  
 x[nonBasic[i]] = 0;  
 }  
  
 for (int i = 0; i < *ROW*; i++) { *// изменение оптимального решения целевой строки* z = z + targetFunc[i] \* x[i];  
 }  
  
 System.*out*.println(String.*format*("%.6f", z) + "\t║");  
 System.*out* .print(  
 "╚═══════╩═══════════════╩═══════════════╩═══════════════╩═══════════════╩═══════════════╩═══════════════╩═══════════════╝\n");  
 for (int i = 0; i < *ROW*; i++) { *// поиск строки системы линейных ограничений, где частное от оптимального решения и коэффициента минимально* if (constraintsLeft[i][tempMinPos] <= 0) {  
 miniRatio[i] = 999;  
 continue;  
 }  
 miniRatio[i] = constraintsRight[i] / constraintsLeft[i][tempMinPos];  
 }  
  
 miniRatioMinPos = *minimum*(miniRatio); *// запись номера минимальной строки* for (int i = 0; i < *ROW*; i++) { *// поиск номера минимального столбца* if (miniRatioMinPos == i) {  
 goOutCol = basic[i];  
 }  
 }  
 *// далее осуществляется преобразование всей таблицы с помощью ключа, являющегося  
 // центральной ячейкой пересечения минимальной строки и минимального столбца по  
 // целевой функции посредством преобразования минимального столбца в столбец  
 // нулевых значений с единичной минимальной ячейкой* basic[miniRatioMinPos] = tempMinPos;  
 nonBasic[tempMinPos] = goOutCol;  
 key = constraintsLeft[miniRatioMinPos][tempMinPos]; *// разрешающий элемент* constraintsRight[miniRatioMinPos] /= key;  
  
 for (int i = 0; i < *COL*; i++) { *// деление минимальной строки на разрешающий элемент* constraintsLeft[miniRatioMinPos][i] /= key;  
 }  
  
 for (int i = 0; i < *ROW*; i++) { *// цикл для осуществления преобразования системы линейных ограничений  
 // посредством разности ячеек старого плана и произведения разрешающего  
 // элемента и элемента минимальной строки* if (miniRatioMinPos == i) {  
 continue;  
 }  
 key = constraintsLeft[i][tempMinPos];  
  
 for (int j = 0; j < *COL*; j++) {  
 constraintsLeft[i][j] -= constraintsLeft[miniRatioMinPos][j] \* key;  
 }  
  
 constraintsRight[i] -= constraintsRight[miniRatioMinPos] \* key;  
 }  
 for (int i = 0; i < *COL*; i++) { *// цикл для проверки знака ячеек целевой строки* flag = true;  
  
 if (temp[i] < 0) {  
 flag = false;  
 break;  
 }  
 }  
 }  
 }  
}

1. **КОНТРОЛЬНЫЙ ТЕСТ**



Результат работы программы на языке Java совпадает с результатом ручного расчёта, приведённого в пункте 2.

**6. Выводы**

В рамках данной работы были изучены основные понятия, разобрана суть симплекс-метода, а также написана программа, реализующая решение задания на языке Java. Результаты программы и ручного расчёта совпали.