# СОДЕРЖАНИЕ

# СОДЕРЖАНИЕ

# СОДЕРЖАНИЕ

# СОДЕРЖАНИЕ

# ВВЕДЕНИЕ

Теория принятия решений рассматривает процесс, включающий несколько ключевых этапов: определение проблемы, поиск информации, анализ альтернатив, выбор наилучшего варианта и реализацию решения. В современных условиях, когда ежедневно приходится сталкиваться с огромным количеством информации и множеством вариантов, правильное принятие решений становится особенно важным.

Одним из центральных аспектов теории принятия решений является определение целей и критериев, а также выбор наиболее подходящего метода принятия решений в зависимости от конкретной ситуации. Для решения различных задач могут использоваться разнообразные методы, каждый из которых обладает своими преимуществами и особенностями.

Метод Парето используется для нахождения оптимальных решений в условиях многокритериальности, позволяя выделить те варианты, которые являются наилучшими по нескольким параметрам одновременно. Метод ELECTRE, в частности версия ELECTRE II, помогает ранжировать альтернативы, основываясь на сравнении их по множеству критериев и учитывая предпочтения лиц, принимающих решения.

Метод анализа иерархий (AHP) применяется для структурирования сложных решений с помощью иерархии критериев и альтернатив. Графический метод используется для наглядного представления данных и поиска решений, особенно в случае двухмерных задач. Симплексный метод является эффективным инструментом для решения задач линейного программирования, позволяя находить оптимальные решения в многомерных пространствах. Двойственная задача связана с исходной задачей линейного программирования и помогает проверить правильность найденного решения, а также получить дополнительную информацию о проблеме.

# 1 МЕТОД ПАРЕТО

## 1.1 Выбор Парето-оптимального множества

Приведем пример выбора барбершопа с использованием Парето-оптимального множества решений. Проанализировав информацию на различных сайтах, были выделены варианты решений (альтернативы) и их оценки, и сведены в (табл. 1.1)

*Таблица 1.1. — Альтернативы и критерии*

| № | Вариант решений | Критерии | | | |
|---|---|---|---|---|---|
| | | Средний чек (руб.) (-) | Удалённость локации (км) (-) | Количество услуг (+) | Рейтинг (от 1 до 5) (+) |
| A1 | OldBoy | 2 300 | 4,30 | 16 | 4,80 |
| A2 | Метод | 1 800 | 2,30 | 17 | 5 |
| A3 | FIDEL | 1 700 | 2,70 | 20 | 4,90 |
| A4 | Чёрная кость | 1 200 | 1,60 | 13 | 4,40 |
| A5 | Бритый Ёж | 800 | 9,30 | 8 | 3,80 |
| A6 | БородаВайб | 1 950 | 2 | 16 | 5 |
| A7 | Чёлочка | 500 | 11,10 | 4 | 2,70 |
| A8 | Бритва | 2 600 | 2,30 | 22 | 4,90 |
| A9 | Baradach | 1 400 | 5,80 | 16 | 4,80 |
| A10 | BomboKlak | 3 500 | 3,20 | 19 | 4,70 |

Примечание: Знаком (-) указывается отрицательное стремление критерия (чем меньше, тем лучше), а знаком (+) – положительное (чем больше, тем лучше).

Было определено, что оптимизация по Парето использует отношение Парето-доминирования, которое отдаёт предпочтение одному объекту перед другим только» том случае, когда первый объект по всем критериям не хуже второго и, хотя бы, но одному из них лучше. При истинности этого условия первый объект считается доминирующим, а второй - доминируемым. Два объекта, для которых предпочтение хотя бы, по одному критерию расходится, считаются несравнимыми. Сравним попарно все альтернативы и сведем их в (табл. 1.2)

*Таблица 1.2 — Сравнения альтернатив*

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Х | Х | Х | Х | Х | Х | Х | Х | Х | Х |
| 2 | А2 | Х | Х | Х | Х | Х | Х | Х | Х | Х |
| 3 | А3 | н | Х | Х | Х | Х | Х | Х | Х | Х |
| 4 | н | н | н | Х | Х | Х | Х | Х | Х | Х |
| 5 | н | н | н | н | Х | Х | Х | Х | Х | Х |
| 6 | А6 | н | н | н | н | Х | Х | Х | Х | Х |
| 7 | н | н | н | н | н | н | Х | Х | Х | Х |
| 8 | н | н | н | н | н | н | н | Х | Х | Х |
| 9 | н | н | н | н | н | н | н | н | Х | Х |
| 10 | н | н | А3 | н | н | н | н | А8 | н | Х |

Примечание: Знаком (н) указываются несравнимые альтернативы.

Парето-оптимальное множество определено альтернативами {А2, А3, А6, А8} представлено в (табл. 1.3).

*Таблица 1.3 — Парето-оптимальное множество*

| № | Вариант решений | Критерии | | | |
|---|---|---|---|---|---|
| | | Средний чек (руб.) (-) | Удалённость локации (км) (-) | Количество услуг (+) | Рейтинг (от 1 до 5) (+) |
| А2 | Метод | 1 800 | 2,30 | 17 | 5 |
| А3 | FIDEL | 1 700 | 2,70 | 20 | 4,90 |
| А6 | БородаВайб | 1 950 | 2 | 16 | 5 |
| А8 | Бритва | 2 600 | 2,30 | 22 | 4,90 |

Очевидно, что выделение множества Парето часто не является удовлетворительным решением. Это связано с тем, что при достаточно большом исходном множестве вариантов множество Парето оказывается недопустимо большим для того, чтобы ЛПР было бы в состоянии осуществить выбор самостоятельно. Таким образом, выделение множества Парето можно рассматривать лишь как предварительный этап оптимизации, и налицо проблема дальнейшего сокращения этого множества.

Наиболее логичным и последовательным представляется путь построения бинарного отношения предпочтения (упорядочивание по желательности), более сильного, чем отношение Парето, позволяющего сузить множество выбираемых вариантов до приемлемых с точки зрения лицо принимающие решение (ЛПР) размеров. Разумеется, для этого потребуется некоторая дополнительная информация, которую придется получить от ЛПР. Это может быть информация о критериях, о самих сравниваемых вариантах и т.п. Таким образом, общая методика исследования задач принятия решения на основе математического моделирования для задач многокритериальной оптимизации может быть реализована в рамках одного из следующих подходов.

**Первый подход**. Для заданной многокритериальной задачи оптимизации находится множество её Парето-оптимальных решений, а выбор конкретного оптимального варианта из множества Парето-оптимальных предоставляется ЛПР.

**Второй подход**. Как уже было сказано выше, производится сужение множества Парето-оптимальных исходов (в идеале – до одного элемента) с помощью некоторых формализованных процедур, что облегчает окончательный исход для ЛПР.

## 1.2 Указание верхних/нижних границ критериев

Установим для приведенного примера верхнюю границу: средний чек не больше 1700 руб., удалённость локации меньше 4,30 км.

В результате установки границ не остается единственной альтернативы, представленная в (табл. 1.4).

*Таблица 1.4 — Результат указания верхних и нижних границ критериев*

| № | Вариант решений | Критерии | | | |
|---|---|---|---|---|---|
| | | Средний чек (руб.) (-) | Удалённость локации (км) (-) | Количество услуг (+) | Рейтинг (от 1 до 5) (+) |
| A2 | Метод | 1 800 | 2,30 | 17 | 5 |
| A3 | FIDEL | 1 700 | 2,70 | 20 | 4,90 |
| A6 | БородаВайб | 1 950 | 2 | 16 | 5 |
| A8 | Бритва | 2 600 | 2,30 | 22 | 4,90 |
| A10 | BomboKlak | 3 500 | 3,20 | 19 | 4,70 |

Варианты, удовлетворяющие этим дополнительным ограничениям: {2, 3, 6, 8, 10}; из них оптимальными по Парето является варианты 2, 3, 6 , 8.

Вывод: основной недостаток метода состоит в том, что оптимальное решение становится здесь субъективным или вообще может отсутствовать, так как зависит, во-первых, от величин назначаемых верхних/нижних границ критериев и, во-вторых, от окончательного выбора, совершаемого принимающим решение. Однако данный метод позволяет поставить ограничения по тем критериям, которые непосредственно относятся к интересам субъекта.

## 1.3 Субоптимизация

Субоптимизацию производят следующим образом: выделяют один из критериев, а по всем остальным критериям назначают нижние границы.

Пусть в примере главным критерием выступает количество услуг; ограничения: средний чек от 1200 руб. и больше; рейтинг больше 4,6; удалённость локации меньше 2,7 км. Отбросим варианты, которые не удовлетворяют заданным ограничениям и составим (табл. 1.5).

*Таблица 1.5 — Результат субоптимизации*

| № | Вариант решений | Критерии | | | |
|---|---|---|---|---|---|
| | | Средний чек (руб.) (-) | Удалённость локации (км) (-) | Количество услуг (+) | Рейтинг (от 1 до 5) (+) |
| А2 | Метод | 1 800 | 2,30 | 17 | 5 |
| А6 | БородаВайб | 1 950 | 2 | 16 | 5 |
| А8 | Бритва | 2 600 | 2,30 | 22 | 4,90 |

Из (табл. 1.5) видно, остаются варианты {2, 6, 8}. Из них минимальный средний чек имеет вариант 2 (Метод). Этот вариант и будет оптимальным.

## 1.4 Лексикографическая оптимизация

Упорядочим критерии в примере по относительной важности, например, следующим образом: важнейший критерий – количество услуг, следующий за ним по важности – удалённость локации.

Из (табл. 1.6) видно, что осталась одна альтернатива.

*Таблица 1.6 — Результат лексикографической оптимизации*

| № | Вариант решений | Критерии | | | |
|---|---|---|---|---|---|
| | | Средний чек (руб.) (-) | Удалённость локации (км) (-) | Количество услуг (+) | Рейтинг (от 1 до 5) (+) |
| А8 | Бритва | 2 600 | 2,30 | 22 | 4,90 |

## 1.5 Результат работы программы

```
 Вывод парето-оптимальных альтернатив
    Средний чек (руб.)  Удалённость локации (км)  Количество услуг  Рейтинг (от 1 до 5)
A2            0.000556                  0.434783                17                  5.0
A3            0.000588                  0.370370                20                  4.9
A6            0.000513                  0.500000                16                  5.0
A8            0.000385                  0.434783                22                  4.9
```

**Рисунок 1.2 – Список Парето-Оптимальных альтернатив**

```
Результат указания верхней/нижней границы: ('Средний чек (руб.)' >= 1700 , 'Удалённость локации (км)' < 4.30)
    Средний чек (руб.)  Удалённость локации (км)  Количество услуг  Рейтинг (от 1 до 5)
A2              1800.0                       2.3                17                  5.0
A3              1700.0                       2.7                20                  4.9
A6              1950.0                       2.0                16                  5.0
A8              2600.0                       2.3                22                  4.9
A10             3500.0                       3.2                19                  4.7
```

**Рисунок 1.3 – Результат указания верхних/нижних границ**

```
 Результат отбора вариантов, удовлетворяющих заданным критериям: главный критерий:
 Средний чек (руб.) >= 1200, Рейтинг (от 1 до 5) >= 4.6, Удалённость локации (км) < 2.70
    Средний чек (руб.)  Удалённость локации (км)  Количество услуг  Рейтинг (от 1 до 5)
A2              1800.0                       2.3                17                  5.0
A6              1950.0                       2.0                16                  5.0
A8              2600.0                       2.3                22                  4.9
```

**Рисунок 1.4 – Результат субоптимизации**

```
 Результат лексикографической оптимизации: (Самая важная: Количество услуг)
    Средний чек (руб.)  Удалённость локации (км)  Количество услуг  Рейтинг (от 1 до 5)
A8            0.000385                  0.434783                22                  4.9
```

**Рисунок 1.5 – Результат лексикографической оптимизации**

# 2 МЕТОД ЭЛЕКТРА II

## 2.1 Выбор лучшего варианта

*Таблица 2.1 — Альтернативы и критерии*

| № | Вариант решений | Критерии | | | |
|---|---|---|---|---|---|
| | | Средний чек (руб.) (-) | Удалённость локации (км) (-) | Количество услуг (+) | Рейтинг (от 1 до 5) (+) |
| A1 | OldBoy | 2 300 | 4,30 | 16 | 4,80 |
| A2 | Метод | 1 800 | 2,30 | 17 | 5 |
| A3 | FIDEL | 1 700 | 2,70 | 20 | 4,90 |
| A4 | Чёрная кость | 1 200 | 1,60 | 13 | 4,40 |
| A5 | Бритый Ёж | 800 | 9,30 | 8 | 3,80 |
| A6 | БородаВайб | 1 950 | 2 | 16 | 5 |
| A7 | Чёлочка | 500 | 11,10 | 4 | 2,70 |
| A8 | Бритва | 2 600 | 2,30 | 22 | 4,90 |
| A9 | Baradach | 1 400 | 5,80 | 16 | 4,80 |
| A10 | BomboKlak | 3 500 | 3,20 | 19 | 4,70 |

Обращаясь к таблице 2.1, составлена таблица критериев, по которым оцениваются проекты (Таблица 2.2).

*Таблица 2.2 – Таблица критериев для оценки альтернатив*

| Критерии | Вес критерия | Шкала | Код | Стремление |
|---|---|---|---|---|
| Средний чек (руб.) | 5 | Дорого<br>Средне<br>Дешево | 15<br>10<br>5 | min |
| Количество услуг | 4 | Много<br>Средне<br>Мало | 15<br>10<br>5 | max |
| Удалённость локации (км) | 4 | Далеко<br>Нормально<br>Близко | 15<br>10<br>5 | min |
| Рейтинг (от 1 до 5) | 5 | Очень большой<br>Большой<br>Средний<br>Маленький | 5<br>4<br>3<br>2 | max |

Составлена таблица оценок выбора лучшего мобильного оператора. Для 10-ти альтернатив заполняем Таблицу 3.

*Таблица 2.3 – Таблица оценок по критериям*

| № | Вариант решений | Критерии | | | |
|---|---|---|---|---|---|
| | | Средний чек (руб.) (-) | Удалённость локации (км) (-) | Количество услуг (+) | Рейтинг (от 1 до 5) (+) |
| A1 | OldBoy | 15 | 15 | 10 | 5 |
| A2 | Метод | 10 | 5 | 10 | 5 |
| A3 | FIDEL | 10 | 10 | 15 | 5 |
| A4 | Чёрная кость | 5 | 5 | 10 | 4 |
| A5 | Бритый Ёж | 5 | 15 | 5 | 3 |
| A6 | БородаВайб | 15 | 5 | 10 | 5 |
| A7 | Чёлочка | 5 | 15 | 5 | 2 |
| A8 | Бритва | 15 | 5 | 15 | 5 |
| A9 | Baradach | 10 | 15 | 10 | 5 |
| A10 | BomboKlak | 15 | 10 | 15 | 4 |
| Вес | | 5 | 4 | 4 | 5 |
| Стремление | | min | min | max | max |

## 2.2 Веса предпочтений

P12 = 0 + 0 + 0 + 0 = 0
N12 = 5 + 4 + 0 + 0 = 9
D12 0/9 = 0 ≤ 1 – отб.

P13 = 0 + 0 + 0 + 0 = 0
N13 = 5 + 4 + 4 + 0 =13
D13 0/13 = 0 ≤ 1 – отб.

P14 = 0 + 0 + 0 + 5 = 5
N14 = 5 + 4 + 0 + 0 = 9
D14 = 5/9 ≤ 1 – отб.

P15 = 0 + 0 + 4 + 5 = 9
N15 = 5 + 0 + 0 + 0 = 5
D15 = 9/5 = 1.8

P16 = 0 + 0 + 4 + 0 = 4
N16 = 0 + 4 + 0 + 0 = 4
D16 = 4/4=1 ≤ 1 – отб.

P17 = 0 + 0 + 4 + 5 = 9
N17 = 5 + 0 + 0 + 0 = 5
D17 = 9/5 = 1.8

P18 = 0 + 0 + 0 + 0 = 0
N18 = 0 + 4 + 4 + 0 = 8
D18 = 0/8 = 0 ≤ 1 – отб.

P19 = 0 + 0 + 0 + 0 = 0
N19 = 5 + 0 + 0 + 0 = 5
D19 =0/5 = 0 ≤ 1 – отб.

P110 = 0 + 0 + 0 + 5 = 5
N110 = 0 + 4 + 4 + 0 = 8
D110 = 5/8 ≤ 1 – отб.

P23 = 0 + 4 + 0 + 0 = 4
N23 = 0 + 0 + 4 + 0 = 4
D23 = 4/4 = 1 ≤ 1 – отб.

P21 = 5 + 4 + 0 + 0 = 9
N21 = 0 + 0 + 0 + 0 = 0
D21 = 9/0 = inf

P31 = 5 + 4 + 4 + 0 =13
N31 = 0 + 0 + 0 + 0 = 0
D31 = 13/0 = inf

P41 = 5 + 4 + 0 + 0 = 9
N41 = 0 + 0 + 0 + 5 = 5
D41 = 9/5 = 1.8

P51 = 5 + 0 + 0 + 0 = 5
N51 = 0 + 0 + 4 + 5 = 9
D51 = 5/9 ≤ 1 – отб.

P61 = 0 + 4 + 0 + 0 = 4
N61 = 0 + 0 + 4 + 0 = 4
D61 = 4/4=1 ≤ 1 – отб.

P71 = 5 + 0 + 0 + 0 = 5
N71 = 0 + 0 + 4 + 5 = 9
D71 = 5/9 ≤ 1 – отб.

P81 = 0 + 4 + 4 + 0 = 8
N81 = 0 + 0 + 0 + 0 = 0
D81 = 8/0 = inf

P91 = 5 + 0 + 0 + 0 = 5
N91 = 0 + 0 + 0 + 0 = 0
D91 = 5/0 = inf

P101 = 0 + 4 + 4 + 0 = 8
N101= 0 + 0 + 0 + 5 = 5
D101 = 8/5 = 1.6

P32 = 0 + 0 + 4 + 0 = 4
N32 = 0 + 4 + 0 + 0 = 4
D32 = 4/4 = 1 ≤ 1 – отб.

P24 = 0 + 0 + 0 + 5 = 5
N24 = 5 + 0 + 0 + 0 = 5
D24 = 5/5 = 1 ≤ 1 – отб.

P25 = 0 + 4 + 4 + 5 =13
N25 = 5 + 0 + 0 + 0 =5
D25 = 13/5 = 2.6

P26 = 5 + 0 + 0 + 0 = 5
N26 = 0 + 0 + 0 + 0 = 0
D26 = 5/0 = inf

P27 = 0 + 4 + 4 + 5 = 13
N27 = 5 + 0 + 0 + 0 = 5
D27 = 13/5 = 2.6

P28 = 5 + 0 + 0 + 0 = 5
N28 = 0 + 0 + 4 + 0 = 4
D28 = 5/4 = 1.25

P29 = 0 + 4 + 0 + 0 = 4
N29 = 0 + 0 + 0 + 0 = 0
D29 = 4/0 = inf

P210 = 5 + 4 + 0 + 5 = 14

N210 = 0 + 0 + 4 + 0 = 4

D210 = 14/4 = 3.5

P42 = 5 + 0 + 0 + 0 = 5
N42 = 0 + 0 + 0 + 5 = 5
D42 = 5/5 = 1 ≤ 1 – отб.

P52 = 5 + 0 + 0 + 0 = 5
N52 = 0 + 4 + 4 + 5 = 13
D52 = 5/13 ≤ 1 – отб.

P62 = 0 + 0 + 0 + 0 = 0
N62 = 5 + 0 + 0 + 0 = 5
D62 = 0/5 ≤ 1 – отб.

P72 = 5 + 0 + 0 + 0 = 5
N72 = 0 + 4 + 4 + 5 = 13
D72 = 5/13 ≤ 1 – отб.

P82 = 0 + 0 + 4 + 0 = 4
N82 = 5 + 0 + 0 + 0 = 5
D82 = 4/5 ≤ 1 – отб.

P92 = 0 + 0 + 0 + 0 = 0
N92 = 0 + 4 + 0 + 4 = 4
D92 = 0/4 ≤ 1 – отб.

P102 = 0 + 0 + 4 + 0 = 4

N102 = 5 + 4 + 0 + 5 = 14

D102 = 4/14 ≤ 1 – отб.

P34 = 0 + 0 + 4 + 5 = 9

N34 = 5 + 4 + 0 + 0 = 9

D34 = 9/9 ≤ 1 – отб.

P43 = 5 + 4 + 0 + 0 = 9

N43 = 0 + 0 + 4 + 5 = 9

D43 = 9/9 ≤ 1 – отб.

P35 = 0 + 4 + 4 + 5 = 13

N35 = 5 + 0 + 0 + 0 = 5

D35 = 13/5 = 2.6

P53 = 5 + 0 + 0 + 0 = 5

N53 = 0 + 4 + 4 + 5 = 13

D53 = 5/13 ≤ 1 – отб.

$P36 = 5 + 0 + 4 + 0 = 9$

$N36 = 0 + 4 + 0 + 0 = 4$

$D36 = 9/4 = 2.25$

$P37 = 0 + 4 + 4 + 5 = 13$

$N37 = 5 + 0 + 0 + 0 = 5$

$D37 = 13/5 = 2.6$

$P38 = 5 + 0 + 0 + 0 = 5$

$N38 = 0 + 4 + 0 + 0 = 4$

$D38 = 5/4 = 1.25$

$P39 = 0 + 4 + 4 + 0 = 8$

$N39 = 0 + 0 + 0 + 0 = 0$

$D39 = 8/0 = inf$

$P310 = 5 + 0 + 0 + 5 = 10$

$N310 = 0 + 0 + 0 + 0 = 0$

$D310 = 10/0 = inf$

$P45 = 0 + 4 + 4 + 5 = 13$

$N45 = 0 + 0 + 0 + 0 = 0$

$D45 = 13/0 = inf$

$P46 = 5 + 0 + 0 + 0 = 5$

$N46 = 0 + 0 + 0 + 5 = 5$

$D46 = 5/5 = 1 \leq 1$ – отб.

$P63 = 0 + 4 + 0 + 0 = 4$

$N63 = 5 + 0 + 4 + 0 = 9$

$D63 = 4//9 \leq 1$ – отб.

$P73 = 5 + 0 + 0 + 0 = 5$

$N73 = 0 + 4 + 4 + 5 = 13$

$D73 = 5/13 \leq 1$ – отб.

$P83 = 0 + 4 + 0 + 0 = 4$

$N83 = 5 + 0 + 0 + 0 = 5$

$D83 = 4/5 \leq 1$ – отб.

$P93 = 0 + 0 + 0 + 0 = 0$

$N93 = 0 + 4 + 4 + 0 = 8$

$D93 = 0/8 \leq 1$ – отб.

$P103 = 0 + 0 + 0 + 0 =0$

$N103 = 5 + 0 + 0 + 5 = 10$

$D103 = 0 /10 \leq 1$ – отб.

$P54 = 0 + 0 + 0 + 0 = 0$

$N54 = 0 + 4 + 4 + 5 =13$

$D54 = 0/13 \leq 1$ – отб.

$P64 = 0 + 0 + 0 + 5 = 5$

$N64 = 5 + 0 + 0 + 0 = 5$

$D64 = 5/5 = 1 \leq 1$ – отб.

$P47 = 0 + 4 + 4 + 0 = 8$

$N47 = 0 + 0 + 0 + 5 = 5$

$D47 = 8/5 = 1.6$

$P48 = 5 + 0 + 0 + 0 = 5$

$N48 = 0 + 0 + 4 + 5 = 9$

$D48 = 5/9 \leq 1$ – отб.

$P49 = 5 + 4 + 0 + 0 = 9$

$N49 = 0 + 0 + 0 + 5 = 5$

$D49 = 9/5 = 1.8$

$P410 = 5 + 4 + 0 + 0 = 9$

$N410 = 0 + 0 + 4 + 0 = 4$

$D410 = 9/4 = 2.25$

$P56 = 5 + 0 + 0 + 0 = 5$

$N56 = 0 + 4 + 4 + 5 = 13$

$D56 = 5/13 \leq 1$ – отб.

$P57 = 0 + 0 + 0 + 5 = 5$

$N57 = 0 + 0 + 0 + 0 = 0$

$D57 = 5/0 = inf$

$P58 = 5 + 0 + 0 + 0 = 5$

$N58 = 0 + 4 + 4 + 5 = 13$

$D58 = 5/13 \leq 1$ – отб.

$P74 = 0 + 0 + 0 + 5 = 5$

$N74 = 0 + 4 + 4 + 0 = 8$

$D74 = 5/8 \leq 1$ – отб.

$P84 = 0 + 0 + 4 + 5 = 9$

$N84 = 5 + 0 + 0 + 0 = 5$

$D84 = 9/5 = 1.8$

$P94 = 0 + 0 + 0 + 5 = 5$

$N94 = 5 + 4 + 0 + 0 = 9$

$D94 = 5/9 \leq 1$ – отб.

$P104 = 0 + 0 + 4 + 0 = 4$

$N104 = 5 + 4 + 0 + 0 = 9$

$D104 = 4/9 \leq 1$ – отб.

$P65 = 0 + 4 + 4 + 5 = 13$

$N65 = 5 + 0 + 0 + 0 = 5$

$D65 = 13/5 =2.6$

$P75 = 0 + 0 + 0 + 0$

$N75 = 0 + 0 + 0 + 5 = 5$

$D75 = 0/5 \leq 1$ – отб.

$P85 = 0 + 4 + 4 + 5 = 13$

$N85 = 5 + 0 + 0 + 0 = 5$

$D85 = 13/5 = 2.6$

P59 = 5 + 0 + 0 + 0 = 5

N59 = 0 + 0 + 4 + 5 = 9

D59 = 5/9 ≤ 1 – отб.

P510 = 5 + 0 + 0 + 9 = 5

N510 = 0 + 4 + 4 + 5 = 13

D510 = 5/13 ≤ 1 – отб.

P67 = 0 + 4 + 4 + 5 = 13

N67 = 5 + 0 + 0 + 0 = 5

D67 = 13/5 = 2.6

P68 = 0 + 0 + 0 + 0 = 0

N68 = 0 + 0 + 4 + 0 = 4

D68 = 0/4 ≤ 1 – отб.

P69 = 0 + 4 + 0 + 0 = 4

N69 = 5 + 0 + 0 + 0 = 5

D69 = 4/5 ≤ 1 – отб.

P610 = 0 + 4 + 0 + 5 = 9

N610 = 0 + 0 + 4 + 0 = 4

D610 = 9/4 = 2.25

P78 = 5 + 0 + 0 + 0 = 5

N78 = 0 + 4 + 4 + 5 =13

D78 = 5/13 ≤ 1 – отб.

P95 = 0 + 0 + 4 + 5 = 9

N95 = 5 + 0 + 0 + 0 = 5

D95 = 9/5 = 1.8

P105 = 0 + 4 + 4 + 5 = 13

N105 = 5 + 0 + 0 + 9 = 5

D105 = 13/5 = 2.6

P76 = 5 + 0 + 0 + 0 = 5

N76 = 0 + 4 + 4 + 5 = 13

D76 = 5/13 ≤ 1 – отб.

P86 = 0 + 0 + 4 + 0 = 4

N86 = 0 + 0 + 0 + 0 = 0

D86 = 4/0 = inf

P96 = 5 + 0 + 0 + 0 = 5

N96 = 0 + 4 + 0 + 0 = 4

D96 = 5/4 = 1.25

P106 = 0 + 0 + 4 + 0 = 4

N106 = 0 + 4 + 0 + 5 = 9

D106 = 4/9 ≤ 1 – отб.

P87 = 0 + 4 + 4 + 5 =13

N 87 = 5 + 0 + 0 + 0 = 5

D87 = 13/5 = 2.6

P79 = 5 + 0 + 0 + 0 = 5
N79 = 0 + 0 + 4 + 5 = 9
D79 = 5/9 ≤ 1 – отб.

P97 = 0 + 0 + 4 + 5 = 9
N97 = 5 + 0 + 0 + 0 = 5
D97 = 9/5 = 1.8

P710 = 5 + 0 + 0 + 0 = 5
N710 = 0 + 4 + 4 + 5 = 13
D710 = 5/13 ≤ 1 – отб.

P107 = 0 + 4 + 4 + 5 = 13
N107 = 5 + 0 + 0 + 0 = 5
D107 = 13/5 2.6

P89 = 0 + 4 + 4 + 0 = 8
N89 = 5 + 0 + 0 + 0 = 5
D89 = 8/5 = 1.6

P98 = 5 + 0 + 0 + 0 = 5
N98 = 0 + 4 + 4 + 0 = 8
D98 = 5/8 ≤ 1 – отб.

P810 = 0 + 4 + 0 + 5 + 9
N810 = 0 + 0 + 0 + 0 = 0
D810 = 9/0 = inf

P108 = 0 + 0 + 0 + 0 = 0
N108 = 0 + 4 + 0 + 5 + 9
D108 = 0/9 ≤ 1 – отб.

P910 = 5 + 0 + 0 + 5 = 10
N910 = 0 + 4 + 4 + 0 = 8
D910 = 10/8 = 1.25

P109 = 0 + 4 + 4 + 0 = 8
N109 = 5 + 0 + 0 + 5 = 10
D109 = 8/10 ≤ 1 – отб.

Составлена матрица предпочтений с внесенными и принятыми значениями.

Составляем матрицу, внося вычисленные (и принятые) значения D. Матрица имеет смысл предпочтений проектов между собой. Для нашего случая матрица выглядит следующим образом (см. табл. 2.4).

*Таблица 2.4 – Полная матрица предпочтений альтернатив*

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | x | - | - | - | 1.8 | - | 1.8 | - | - | - |
| 2 | inf | x | - | - | 2.6 | inf | 2.6 | 1.25 | inf | 3.5 |
| 3 | inf | - | x | | 2.6 | 2.25 | 2.6 | 1.25 | inf | inf |
| 4 | 1.8 | - | - | x | inf | - | inf | - | 1.8 | 2.25 |
| 5 | - | - | - | - | x | - | inf | - | - | - |
| 6 | inf | - | - | - | 2.6 | x | 2.6 | - | - | 2.25 |
| 7 | - | - | - | - | - | - | x | - | - | - |
| 8 | inf | - | - | 1.8 | 2.6 | inf | 2.6 | x | 1.6 | inf |
| 9 | inf | - | - | - | 1.8 | 1.25 | 1.8 | - | x | 1.25 |
| 10 | 1.6 | - | - | - | 2.6 | - | 2.6 | - | - | x |

По матрице построен граф предпочтений (Рисунок 1).

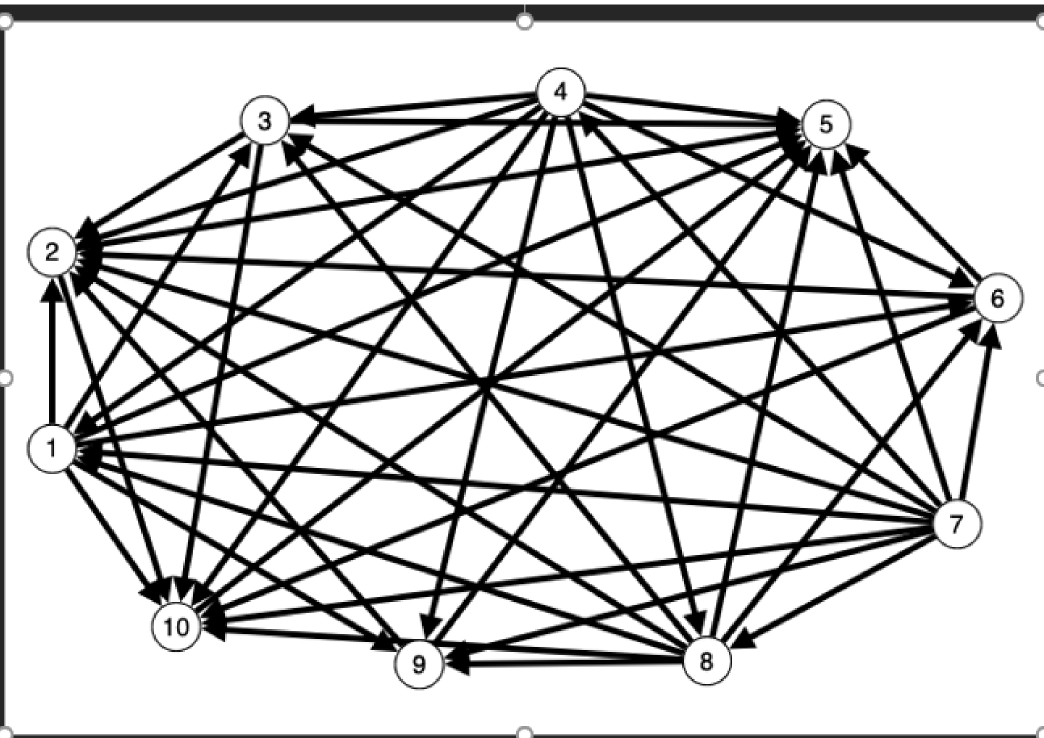


**Рисунок 2.1 – Вид графа предпочтений**

Петли в графе нет, поэтому оставляем граф без изменений.

21

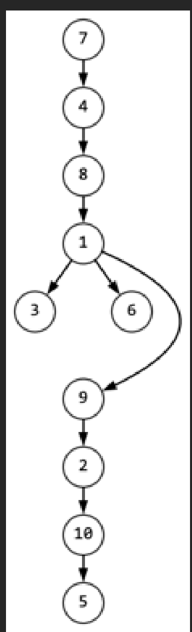


**Рисунок 2.2 – Вид графа предпочтений без дополнительных связей**

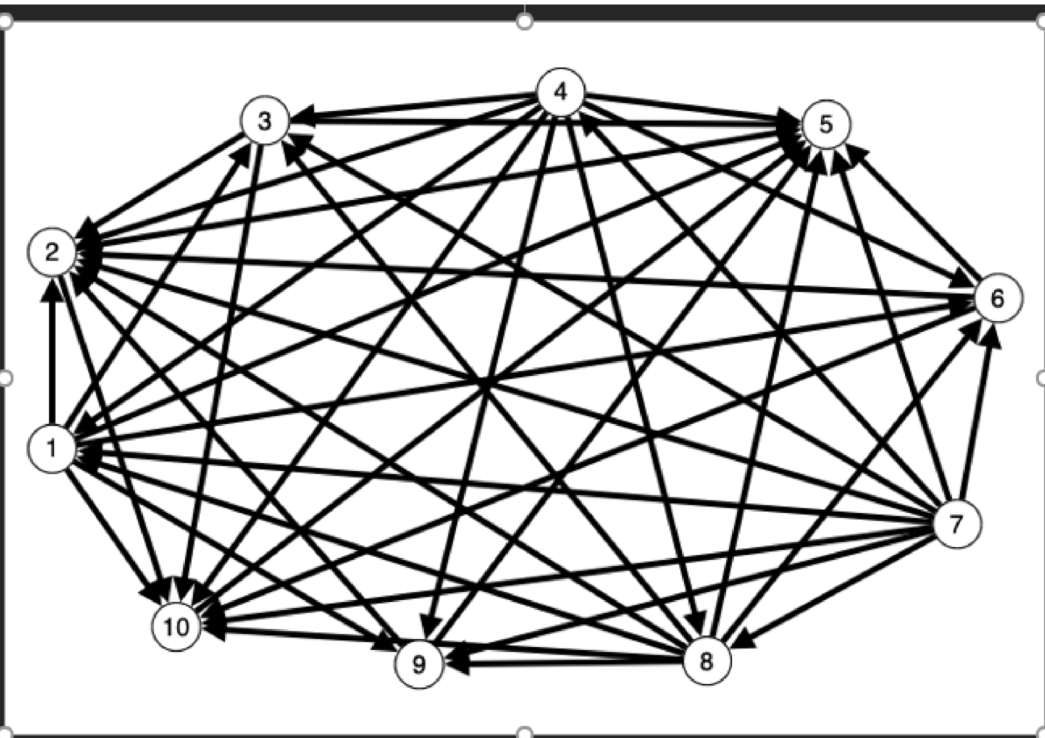


**Рисунок 2.1 – Вид графа предпочтений**

Петли в графе нет, поэтому оставляем граф без изменений.

21

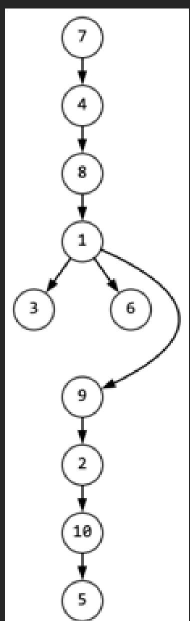


**Рисунок 2.2 – Вид графа предпочтений без дополнительных связей**

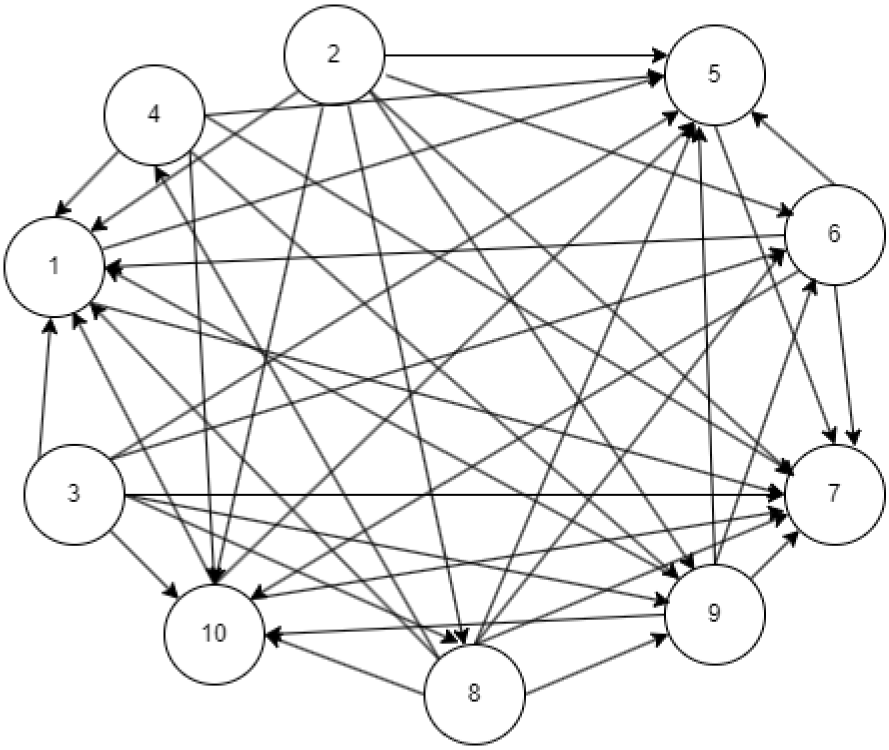


**Рисунок 2.1 – Вид графа предпочтений**

Петли в графе нет, поэтому оставляем граф без изменений.

## 2.3 Результат работы программы

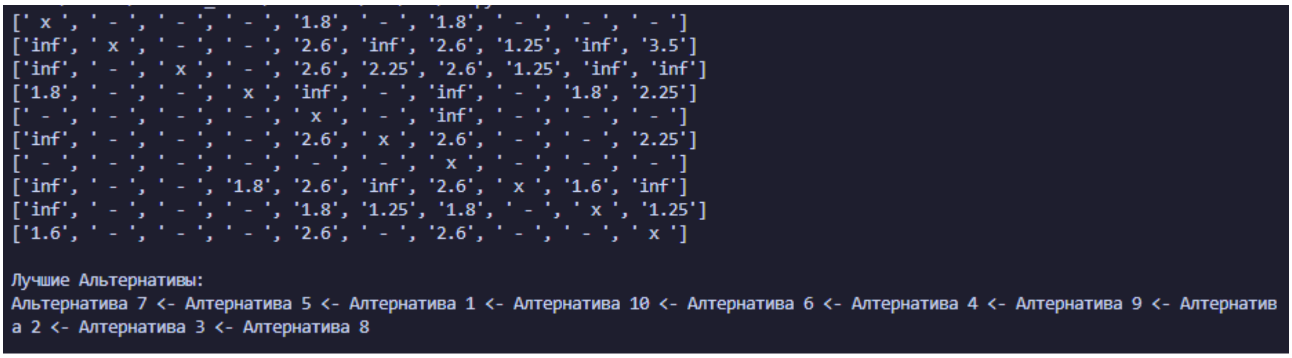


**Рисунок 2.3 – Результат работы программы. Вывод матрицы предпочтений.**

# 3 МЕТОД АНАЛИЗА ИЕРАРХИЙ

## 3.1 Постановка задачи

Задача практической работы: выбор сотового оператора.

## 3.1 Представление проблемы в виде иерархии

Первый этап – представление проблемы в виде иерархии или сети. В простейшем случае, иерархия строится, начиная с цели, которая помещается в вершину иерархии. Через промежуточные уровни, на которых располагаются критерии и от которых зависят последующие уровни, к самому низкому уровню, который содержит перечень альтернатив.

Иерархия считается полной, если каждый элемент заданного уровня является критерием для всех элементов нижнего уровня

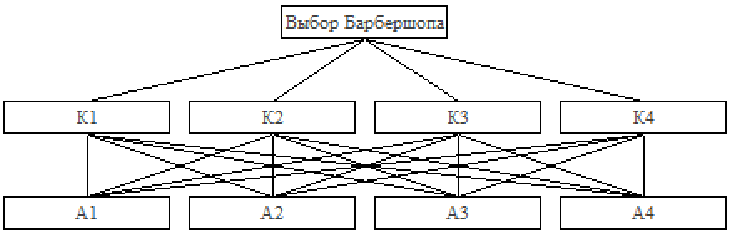


**Рисунок 3.1 – Полная доминантная иерархия.**

Критерии:

К 1 – Средний чек;

К 2 – Рейтинг;

К 3 – Количество услуг;

К 4 – Удалённость локации.

Альтернативы:

А 1 - Метод;

А 2 -FIDEL;

А 3 - БородаВайб;

А 4 – Бритва.

## 3.3 Установка приоритетов критериев

После иерархического представления задачи установлены приоритеты критериев и оценена каждая из альтернатив по критериям, определена наиболее важная их них. В методе анализа иерархий элементы сравниваются попарно по отношению к их влиянию на общую для них характеристику. Парные сравнения приводят к записи характеристик сравнений в виде квадратной таблицы чисел, которая называется матрицей. Для облегчения работы введена шкала относительной важности (Таблица 3.1).

*Таблица 3.1 – Шкала относительной важности*

| Интенсивность относительной важности | Определение | Объяснение |
|---|---|---|
| 1 | Равная важность | Равный вклад двух критериев в цель. |
| 3 | Слабое превосходство | Дают легкое превосходство одной альтернативы над другой |
| 5 | Умеренное превосходство | Опыт и суждения дают умеренное превосходство |
| 7 | Сильное превосходство | Одному из критериев дается настолько сильное предпочтение. |
| 9 | Абсолютное превосходство | Очевидность превосходства одного критерия над другим |
| 2,4,6,8 | Промежуточные решения между двумя соседними суждениями | Применяется в компромиссных случаях |

Шкала содержит соответствующие обратные значения.

## 3.4 Синтез приоритетов

После построения иерархии и определения величин парных субъективных суждений следует этап, на котором иерархическая декомпозиция и относительные суждения объединяются для получения осмысленного решения многокритериальной задачи принятия решений. Из групп парных сравнений формируется набор локальных критериев, которые выражают относительное влияние элементов на элемент, расположенный на уровне выше. Составлена обратно симметричная матрица для парного сравнения критериев (Таблица 3.2).

*Таблица 3.2 – Матрица парного сравнения критериев*

| Цель | К 1 | К 2 | К 3 | К 4 | Vi | $W_{2i}$ |
|---|---|---|---|---|---|---|
| К 1 | 1 | 1/3 | 4 | 7 | 1.747 | 0.313 |
| К 2 | 3 | 1 | 3 | 7 | 2.817 | 0.508 |
| К 3 | 1/4 | 1/3 | 1 | 4 | 0.759 | 0.132 |
| К 4 | 1/7 | 1/7 | 1/4 | 1 | 0.267 | 0.046 |
| ∑Vi | | | | | 5.590 | |

Для определения относительной ценности каждого элемента необходимо найти геометрическое среднее и с этой целью перемножить n элементов каждой строки и из полученного результата извлечь корни n-й степени (размерность матрицы n=5).

Строка № 1

$V_1=(1x1/3x4x7)_{1/4} = 1.747;$

Строка № 2

$V_2=(3x1x3x7)_{1/4} = 2.817;$

Строка № 3

$V_3=(1/4x1/3x1x4)_{1/4} = 0.759;$

Строка № 4

$V_4=(1/7x1/7x1/4x1)_{1/4} = 0.267;$

Проведена нормализация полученных чисел. Для этого определен нормирующий коэффициент ∑Vi.

$\sum Vi = V_1 + V_2 + V_3 + V_4 = 1.747 + 2.817 + 0.759 + 0.267 = 5.590$.

Найдена важность приоритетов $W_{2i}$, для этого каждое из чисел Vi разделено на ∑Vi.

***Строка № 1***

$W_{21} = 1.747 / \sum Vi = 0,313$;

Строка № 2

$W_{22} = 2.817 / \sum Vi = 0,506$;

***Строка № 3***

$W_{23} = 0.759 / \sum Vi = 0,132$;

***Строка № 4***

$W_{24} = 0.267 / \sum Vi = 0,046$;

В результате получен вектор приоритетов:

$W_{2i}$ = (0,313; 0,506; 0,132; 0,046), где индекс 2 означает, что вектор приоритетов относится ко второму уровню иерархии.

К 1 – Средний чек (Таблица 3.2.1);

*Таблица 3.2.1 – Матрица сравнения по критерию 1*

| К1 | А1 | А2 | А3 | А4 | $V_{K1Y}$ | $W_{3K1Y}$ |
|---|---|---|---|---|---|---|
| А1 | 1 | 1/3 | 3 | 7 | 1.62658 | 0.292 |
| А2 | 3 | 1 | 3 | 7 | 2.81731 | 0.511 |
| А3 | 1/3 | 1/3 | 1 | 5 | 0.86334 | 0.152 |
| А4 | 1/7 | 1/7 | 1/5 | 1 | 0.25276 | 0.045 |
| $\sum V_{K1Y}$ | | | | | 5.55999 | |

Определена относительная ценность каждого элемента.

Строка № 1

$V_{K11}$=(1x1/3x3x7)1/4= 1.62658;

Строка № 2

$V_{K12}$=(3x1x3x7)1/4= 2.81731;

Строка № 3

$V_{K13}$=(1/3x1/3x1x5)1/4= 0.86334;

Строка № 4

$V_{K14}$=(1/7x1/7x1/5x1)1/4= 0.25276;

Проведена нормализация полученных чисел. Для этого определен нормирующий коэффициент $\sum V_{K1Y}$.

∑VK1Y = VK11 + VK12 + VK13 + VK14 = 5.55999.

Найдена важность приоритетов $W_{3K1Y}$, для этого каждое из чисел $V_{K1Y}$ разделено на $\sum V_{K1Y}$.

Строка № 1

$W_{3K11}$= 1.62658/ ∑Vi = 0.292;

Строка № 2

$W_{3K12}$= 2.81731/ ∑Vi = 0.511;

Строка № 3

$W_{3K13}$= 0.86334/ ∑Vi = 0.152;

Строка № 4

$W_{3K14}$= 0.25276/ ∑Vi = 0.045;

В результате получаем вектор приоритетов:

$W_{3K1Y}$ = (0.292; 0.511; 0.152; 0.045), где индекс 3 означает, что вектор приоритетов относится к третьему уровню иерархии критерия К1.

К 2 – Рейтинг (Таблица 3.2.2);

*Таблица 3.2.2 – Матрица сравнения по критерию 2*

| К2 | А1 | А2 | А3 | А4 | $V_{K2Y}$ | $W_{3K2Y}$ |
|---|---|---|---|---|---|---|
| А1 | 1 | 1/3 | 1/2 | 3 | 0.840 | 0.170 |
| А2 | 3 | 1 | 3 | 3 | 2.279 | 0.481 |
| А3 | 2 | 1/3 | 1 | 4 | 1.277 | 0.262 |
| А4 | 1/3 | 1/3 | 1/4 | 1 | 0.408 | 0.087 |
| | | | $\sum V_{K2Y}$ | 4.3094 | | |

Определена относительная ценность каждого элемента.

Строка № 1

$V_{K21}$=(1x1/3x1/2x3)1/4=0.840;

Строка № 2

$V_{K22}$=(3x1x3x3)1/4= 2.279;

Строка № 3

$V_{K23}$=(2x1/3x1x4)1/4= 1.277;

Строка № 4

$V_{K24}$=(1/3x1/3x1/4x1)1/4= 0.408;

Проведена нормализация полученных чисел. Для этого определен нормирующий коэффициент $\sum V_{K2Y}$.

∑VK2Y = VK21 + VK22 + VK23 + VK24 = 4.804.

Найдена важность приоритетов $W_{3K2Y}$, для этого каждое из чисел $V_{K2Y}$ разделено на $\sum V_{K2Y}$.

Строка № 1

$W_{3K21}$= 0.840 / ∑Vi = 0.170;

Строка № 2

$W_{3K22}$= 2.279 / ∑Vi = 0.481;

Строка № 3

$W_{3K23}$= 1.277/ ∑Vi = 0.262;

Строка № 4

$W_{3K24}$= 0.408/ ∑Vi = 0.087;

В результате получаем вектор приоритетов:

$W_{3K2Y}$ = (0.170; 0.481; 0.262; 0.087), где индекс 3 означает, что вектор приоритетов относится к третьему уровню иерархии критерия К2.

К 3 – Количество услуг (Таблица 3.2.3);

*Таблица 3.2.3 – Матрица сравнения по критерию 3*

| К3 | А1 | А2 | А3 | А4 | $V_{K3Y}$ | $W_{3K3Y}$ |
|---|---|---|---|---|---|---|
| А1 | 1 | 1/3 | 3 | 1/7 | 0.614788 | 0.105 |
| А2 | 3 | 1 | 5 | 1/3 | 1.49535 | 0.249 |
| А3 | 1/3 | 1/5 | 1 | 1/7 | 0.312394 | 0.054 |
| А4 | 7 | 3 | 7 | 1 | 3.482 | 0.592 |
| $\sum V_{K35}$ | | | | | 5.9045 | |

Определена относительная ценность каждого элемента.

Строка № 1

$V_{K31}$=(1x1/3x3x1/7)1/4= 0.614788;

Строка № 2

$V_{K32}$=(3x1x5x1/3)1/4=1.49535;

Строка № 3

$V_{K33}$=(1/3x1/5x1x1/7)1/4= 0.312394;

Строка № 4

$V_{K34}$=(7x3x7x1)1/5= 3.482;

Проведена нормализация полученных чисел. Для этого определен нормирующий коэффициент $\sum V_{K3Y}$.

∑VК3Y = VК31 + VК32 + VК33 + VК34 = 5.9045.

Найдена важность приоритетов $W_{3K2Y}$, для этого каждое из чисел $V_{K2Y}$ разделено на $\sum V_{K2Y}$.

Строка № 1

$W_{3K31}$= 0.614788/ ∑Vi = 0.105;

Строка № 2

$W_{3K32}$= 1.49535/ ∑Vi = 0.249;

Строка № 3

$W_{3K33}$= 0.312394/ ∑Vi = 0.054;

Строка № 4

$W_{3K34}$= 3.482/ ∑Vi = 0.592;

В результате получаем вектор приоритетов:

$W_{3K3Y}$ = (0.105; 0.249; 0.054; 0.592), где индекс 3 означает, что вектор приоритетов относится к третьему уровню иерархии критерия К3.

К 4 – Удалённость локации (Таблица 3.2.4);

*Таблица 3.2.4 – Матрица сравнения по критерию 4*

| К4 | А1 | А2 | А3 | А4 | $V_{K4Y}$ | $W_{3K4Y}$ |
|---|---|---|---|---|---|---|
| А1 | 1 | 3 | 1/3 | 1 | 1 | 0.200 |
| А2 | 1/3 | 1 | 1/5 | 1/3 | 0.386097 | 0.078 |
| А3 | 3 | 5 | 1 | 3 | 2.59002 | 0.522 |
| А4 | 1 | 3 | 1/3 | 1 | 1 | 0.200 |
| $\sum V_{K4Y}$ | | | | | 4.976117 | |

Определена относительная ценность каждого элемента.

Строка № 1

$V_{K41}$=(1x3x1/3x1)1/4 = 1;

Строка № 2

$V_{K42}$=(1/3x1x1/5x1/3)1/4 = 0.386097;

Строка № 3

$V_{K43}$=(3x5x1x3)1/4 = 2.59002;

Строка № 4

$V_{K44}$=(1x3x1/3x1)1/4 = 1;

Проведена нормализация полученных чисел. Для этого определен нормирующий коэффициент $\sum V_{K4Y}$.

∑VK4Y = VK41 + VK42 + VK43 + VK44 = 4.976117.

Найдена важность приоритетов $W_{3K4Y}$, для этого каждое из чисел $V_{K4Y}$ разделено на $\sum V_{K4Y}$.

Строка № 1

$W_{3K41}$= 1/ $\sum$Vi = 0.200;

Строка № 2

$W_{3K42}$= 0.386097/ $\sum$Vi = 0.078;

Строка № 3

$W_{3K43}$= 2.59002/ $\sum$Vi = 0.522;

Строка № 4

$W_{3K44}$= 1/ $\sum$Vi = 0.200;

В результате получаем вектор приоритетов:

$W_{3K4Y}$ = (0.200; 0.078; 0.522; 0.200), где индекс 3 означает, что вектор приоритетов относится к третьему уровню иерархии критерия К4.

## 3.5 Согласованность локальных приоритетов

Любая матрица суждений в общем случае не согласована, так как суждения отражают субъективные мнения ЛПР, а сравнение элементов, которые имеют количественные эквиваленты, может быть несогласованным из-за присутствия погрешности при проведении измерений. Совершенной согласованности парных сравнений даже в идеальном случае на практике достичь трудно. Нужен способ оценки степени согласованности при решении конкретной задачи.

Метод анализа иерархий дает возможность провести такую оценку.

Вместе с матрицей парных сравнений есть мера оценки степени отклонения от согласованности. Когда такие отклонения превышают установленные пределы тем, кто проводит решение задачи, необходимо их пересмотреть.

В таблице приведены средние значения индекса случайной согласованности (СИ) для случайных матриц суждений разного порядка.

В нашей задаче размерность матрицы n=5, тогда среднее значение индекса случайной согласованности СИ =0.90.

Определены индекс согласованности и отношение согласованности для матрицы «Выбор Барбершопа» (Таблица 3.3.1).

*Таблица 3.3.1 – Матрица «Выбор сотового оператора»*

| Цель | К 1 | К 2 | К 3 | К 4 | $W_{2i}$ |
|---|---|---|---|---|---|
| К 1 | 1 | 1/3 | 4 | 7 | 0.313 |
| К 2 | 3 | 1 | 3 | 7 | 0.508 |
| К 3 | 1/4 | 1/3 | 1 | 4 | 0.132 |
| К 4 | 1/7 | 1/7 | 1/4 | 1 | 0.046 |

Определена сумма каждого столбца матрицы суждений.

$S_1 = 1 +3 + 1/4 + 1/7 = 4.3928$;

$S_2 =1/3 + 1 + 1/3 + 1/7 = 1.8095$;

$S_3 =4+ 3 + 1 + 1/4 = 8.25$;

$S_4 = 7 + 7 + 4 + 1 = 19$;

Полученный результат умножен на компоненту нормализованного вектора приоритетов, т.е. сумму суждений первого столбца на первую компоненту, сумму суждений второго столбца - на вторую и т.д.

$P_1 = S_1 \text{ x } W_{21} = 1.3749$;

$P_2 = S_2 \text{ x } W_{22} = 0.919$;

$P_3 = S_3 \text{ x } W_{23} = 1.089$;

$P_4 = S_4 \text{ x } W_{24} = 0.874$;

Сумма чисел Pj отражает пропорциональность предпочтений, чем ближе эта величина к n (числу объектов и видов действия в матрице парных сравнений), тем более согласованны суждения.

$$\lambda_{max} = P_1 + P_2 + P_3 + P_4 = 4.266.$$

Отклонение от согласованности выражается индексом согласованности.

$$ИС = (\lambda_{max} - n)/(n - 1) = 0.089.$$

Отношение индекса согласованности ИС к среднему значению случайного индекса согласованности СИ называется отношением согласованности ОС.

$$ОС = ИС/СИ = 0.098.$$

Значение ОС меньше или равное 0.10 считается приемлемым, значит матрица «Выбор Барбершопа» согласована.

Определены индекс согласованности и отношение согласованности для матрицы К 1 – Средний чек (Таблица 3.3.2).

*Таблица 3.3.2 – Матрица сравнения по критерию 1*

| К1 | А1 | А2 | А3 | А4 | $W_{3K1Y}$ |
|---|---|---|---|---|---|
| А1 | 1 | 1/3 | 3 | 7 | 0.292 |
| А2 | 3 | 1 | 3 | 7 | 0.511 |
| А3 | 1/3 | 1/3 | 1 | 5 | 0.152 |
| А4 | 1/7 | 1/7 | 1/5 | 1 | 0.045 |

Определяется сумма каждого столбца матрицы суждений.

$S_{1\,K1} = 4.4761;$

$S_{2\,K1} = 1.80952;$

$S_{3\,K1} = 7.2;$

$S_{4\,K1} = 20;$

Затем полученный результат умножен на компоненту нормализованного вектора приоритетов.

$P_{1\,K1} = S_1 \times W_{3K11} = 1.30704;$

$P_{2\,K1} = S_2 \times W_{3K12} = 0.92466;$

$P_{3\,K1} = S_3 \times W_{3K13} = 1.0944;$

$P_{4\,K1} = S_1 \times W_{3K14} = 0.89999;$

Найдена пропорциональность предпочтений.

$$\lambda_{max\,K1} = P_{1K1} + P_{2K1} + P_{3K1} + P_{4K1} = 4.226.$$

Отклонение от согласованности выражается индексом согласованности.

$$ИС_{К1} = (\lambda_{max\ К1} - n)/(n - 1) = 0.075.$$

Найдено отношение согласованности ОС.

$$ОС_{К1} = ИС/СИ = 0.08.$$

Значение ОС меньше или равное 0.10 считается приемлемым, значит матрица К 1 (Средний чек) согласована.

Определены индекс согласованности и отношение согласованности для матрицы К 2 – Рейтинг (Таблица 3.3.3).

*Таблица 3.3.3 – Матрица сравнения по критерию 2*

| К2 | А1 | А2 | А3 | А4 | $W_{3К2Y}$ |
|---|---|---|---|---|---|
| А1 | 1 | 1/3 | 1/2 | 3 | 0.170 |
| А2 | 3 | 1 | 3 | 3 | 0.481 |
| А3 | 2 | 1/3 | 1 | 4 | 0.262 |
| А4 | 1/3 | 1/3 | 1/4 | 1 | 0.087 |

Определена сумма каждого столбца матрицы суждений.

$S_{1\ К2} = 6.333$;

$S_{2\ К2} = 1.999$;

$S_{3\ К2} = 4.75$;

$S_{4\ К2} = 11$.

Затем полученный результат умножен на компоненту нормализованного вектора приоритетов.

$P_{1\ К2} = S_1 \times W_{3\ К21} = 1.076$;

$P_{2\ К2} = S_2 \times W_{3\ К22} = 0.9619$;

$P_{3\ К2} = S_3 \times W_{3\ К23} = 1.2445$;

$P_{4\ К2} = S_4 \times W_{3\ К24} = 0.957$;

Найдена пропорциональность предпочтений.

$\lambda_{max\ К2} = P_{1К2} + P_{2К2} + P_{3К2} + P_{4К2} = 4.244$.

Отклонение от согласованности выражается индексом согласованности.

$ИС_{К2} = (\lambda_{max\ К2} - n)/(n - 1) = 0.081$.

Найдено отношение согласованности ОС.

ОС $_{К2}$ = ИС/СИ = 0.090.

Значение ОС меньше или равное 0.10 считается приемлемым, значит матрица К 2 (Рейтинг) согласована.

Определены индекс согласованности и отношение согласованности для матрицы К 3 – Количество услуг (Таблица 3.3.4).

*Таблица 3.3.4 – Матрица сравнения по критерию 3*

| К3 | А1 | А2 | А3 | А4 | $W_{3К3Y}$ |
|---|---|---|---|---|---|
| А1 | 1 | 1/3 | 3 | 1/7 | 0.105 |
| А2 | 3 | 1 | 5 | 1/3 | 0.249 |
| А3 | 1/3 | 1/5 | 1 | 1/7 | 0.054 |
| А4 | 7 | 3 | 7 | 1 | 0.592 |

Определена сумма каждого столбца матрицы суждений.

$S_{1К3}$ = 11.333;

$S_{2\ К3}$ = 4.533;

$S_{3\ К3}$ = 16;

$S_{4\ К3}$ = 1.619.

Затем полученный результат умножен на компоненту нормализованного вектора приоритетов.

$P_{1\ К3}$ = $S_1$ x $W_{3\ К31}$ = 1.189;

$P_{2\ К3}$ = $S_2$ x $W_{3\ К32}$ = 1.1288;

$P_{3\ К3}$ = $S_3$ x $W_{3\ К33}$ = 0.864;

$P_{4\ К3}$ = $S_4$ x $W_{3\ К34}$ = 0.958 .

Найдем пропорциональность предпочтений.

$$\lambda_{max\ К3} = P_{1К3} + P_{2К3} + P_{3К3} + P_{4К3} = 4.143.$$

Отклонение от согласованности выражается индексом согласованности.

$$ИС_{К3} = (\lambda_{max\ К3} - n)/(n - 1) = 0.048.$$

Найдено отношение согласованности ОС.

$$ОС_{К3} = ИС/СИ = 0.053.$$

Значение ОС меньше или равное 0.10 считается приемлемым, значит матрица К 3 согласована.

Определены индекс согласованности и отношение согласованности для матрицы К 4 – удаленность от локации (Таблица 3.3.5).

*Таблица 3.3.5 – Матрица сравнения по критерию 4*

| К4 | А1 | А2 | А3 | А4 | $W_{3К4Y}$ |
|---|---|---|---|---|---|
| А1 | 1 | 3 | 1/3 | 1 | 0.200 |
| А2 | 1/3 | 1 | 1/5 | 1/3 | 0.078 |
| А3 | 3 | 5 | 1 | 3 | 0.522 |
| А4 | 1 | 3 | 1/3 | 1 | 0.200 |

Определена сумма каждого столбца матрицы суждений.

$S_{1К4} = 5.333;$

$S_{2К4} = 12;$

$S_{3К4} = 1.866;$

$S_{4К4} = 5.333.$

Затем полученный результат умножен на компоненту нормализованного вектора приоритетов.

$P_{1К4} = S_1 \text{ x } W_{3\,К41} = 1.066;$

$P_{2К4} = S_2 \text{ x } W_{3\,К42} = 0.935;$

$P_{3К4} = S_3 \text{ x } W_{3\,К43} = 0.974;$

$P_{4К4} = S_4 \text{ x } W_{3\,К44} = 1.066.$

Найдена пропорциональность предпочтений.

$$\lambda_{max\ К4} = P_{1К4} + P_{2К4} + P_{3К4} + P_{4К4} = 4.043.$$

Отклонение от согласованности выражается индексом согласованности.

$$ИС_{К4} = (\lambda_{max\ К4} - n)/(n - 1) = 0.014.$$

Найдено отношение согласованности ОС.

$$ОС_{К4} = ИС/СИ = 0.015.$$

Значение ОС меньше или равное 0,10 считается приемлемым, значит матрица К 4 (удаленность от ближайшей станции метро) согласована.

## 3.6 Синтез альтернатив

Векторы приоритетов и отношения согласованности определяются для всех матриц суждений, начиная со второго уровня.

Для определения приоритетов альтернатив локальные приоритеты умножены на приоритет соответствующего критерия на высшем уровне и найдены суммы по каждому элементу в соответствии с критериями, на которые воздействует этот элемент.

$W_{2i}$ = (0.313; 0.508; 0.132; 0.046);

$W_{3K1Y}$ = (0.292; 0.511; 0.152; 0.045);

$W_{3K2Y}$ = (0.170; 0.481; 0.262; 0.087);

$W_{3K3Y}$ = (0.105; 0.249; 0.054; 0.592);

$W_{3K4Y}$ = (0.200; 0.078; 0.522; 0.200).

Приоритеты альтернатив получены следующим образом:

$W_1 = W_{21} \times W_{3K11} + W_{22} \times W_{3K21} + W_{23} \times W_{3K31} + W_{24} \times W_{3K41} = 0.203701.$

$W_2 = W_{21} \times W_{3K12} + W_{22} \times W_{3K22} + W_{23} \times W_{3K32} + W_{24} \times W_{3K42} = 0.435739.$

$W_3 = W_{21} \times W_{3K13} + W_{22} \times W_{3K23} + W_{23} \times W_{3K33} + W_{24} \times W_{3K43} = 0.214747.$

$W_4 = W_{21} \times W_{3K14} + W_{22} \times W_{3K24} + W_{23} \times W_{3K34} + W_{24} \times W_{3K44} = 0.147126.$

Таким образом, приоритеты альтернатив равны:

Альтернатива А1 (Метод) - $W_1$ приоритет равен = 0.203701;

Альтернатива А2(FIDEL) - $W_2$ приоритет равен =0.435739;

Альтернатива А3 (БородаВайб) - $W_3$ приоритет равен =0.214747;

Альтернатива А4 (Бритва) – $W_4$ приоритет равен = 0.147126.

## 3.7 Вывод метода анализа иерархий

С помощью метода анализа иерархий мы получаем, что Альтернатива А1, является лучшей, т.к. её приоритет имеет наибольшее значение.

## 3.8 Результат работы программы

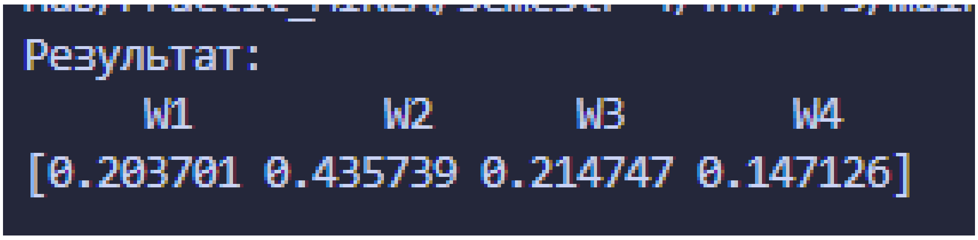


**Рисунок 3.2 – Вывод программы**

# 4 ГРАФИЧЕСКИЙ МЕТОД

## 4.1 Постановка задачи

Решить задачу линейного программирования с двумя переменными графическим методом.

## 4.2 Данные индивидуального варианта

$$f(x) = 5x_1 + 6x_2 \rightarrow min/max$$

$$\begin{cases} x_1 + 2x_2 \leq 13 \\ 6x_1 + x_2 \leq 34 \\ x_1, x_2 \geq 0 \end{cases}$$

## 4.3 Подготовка данных

В среде Microsoft Excel составим 4 столбца:

1. $x_1$ – значения от 0 до 10 с шагом 0,5;
2. $x_2 = \frac{13-x_1}{2}$ – значения ограничения $x_1 + 2x_2 \leq 13$
3. $x_2 = 34 - 6x_1$ – значения ограничения $6x_1 + x_2 \leq 34$
4. $x_2 = \frac{5x_1}{6}$ – значения $f(x) = 5x_1 + 6x_2$

*Таблица 4.1 – Данные для графика*

| $x_1$ | $x_2 = \frac{13 - x_1}{2}$ | $x_2 = 34 - 6x_1$ | $x_2 = \frac{5x_1}{6}$ |
|---|---|---|---|
| 0 | 6,5 | 34 | 0,00 |
| 0,5 | 6,25 | 31 | 0,42 |
| 1 | 6 | 28 | 0,83 |
| 1,5 | 5,75 | 25 | 1,25 |
| 2 | 5,5 | 22 | 1,67 |
| 2,5 | 5,25 | 19 | 2,08 |
| 3 | 5 | 16 | 2,50 |
| 3,5 | 4,75 | 13 | 2,92 |
| 4 | 4,5 | 10 | 3,33 |
| 4,5 | 4,25 | 7 | 3,75 |
| 5 | 4 | 4 | 4,17 |
| 5,5 | 3,75 | 1 | 4,58 |
| 6 | 3,5 | -2 | 5,00 |
| 6,5 | 3,25 | -5 | 5,42 |
| 7 | 3 | -8 | 5,83 |
| 7,5 | 2,75 | -11 | 6,25 |
| 8 | 2,5 | -14 | 6,67 |
| 8,5 | 2,25 | -17 | 7,08 |
| 9 | 2 | -20 | 7,50 |
| 9,5 | 1,75 | -23 | 7,92 |
| 10 | 1,5 | -26 | 8,33 |

## 4.4 Построение графика

Выделим таблицу подготовленных данных и построим гладкий график. Произведем настройку шага координатной оси x1 и получим следующий график (Рисунок 4.1)

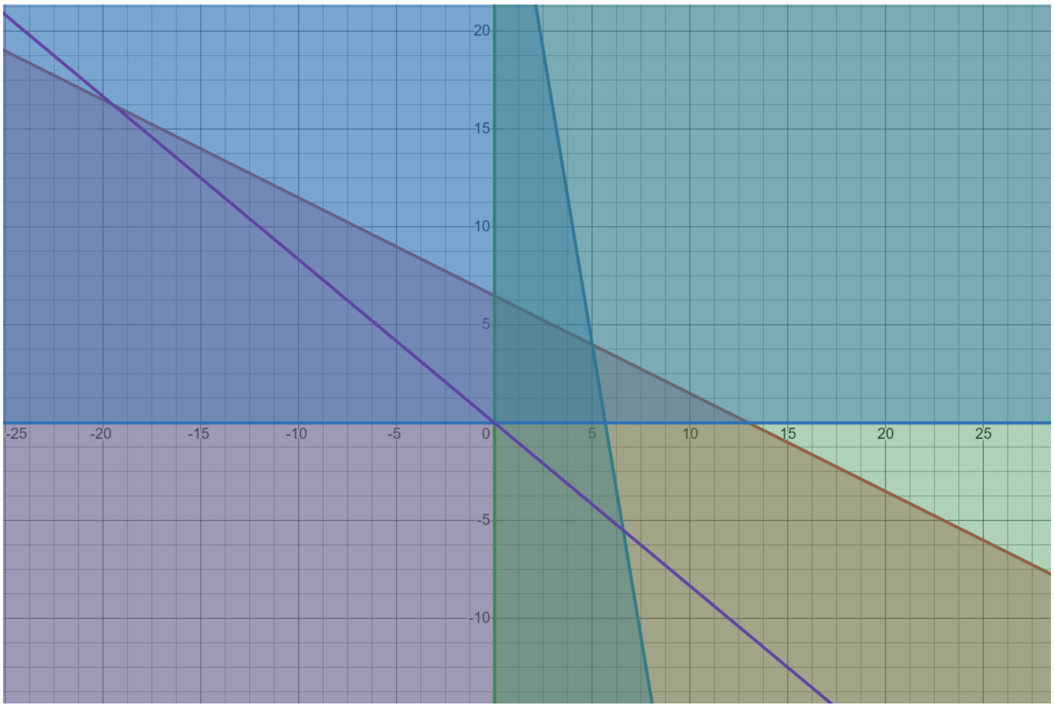


**Рисунок 4.1 – Построение графиков по данным**

## 4.5 Выделение области допустимых решений

Чтобы определить форму ОДР надо рассмотреть каждую из построенных прямых по отдельности и, заменив мысленно в соответствующем уравнении знак равенства на исходное неравенство, определить, с какой стороны от рассматриваемой прямой лежит ОДР. Для этого необходимо решить соответствующее неравенство относительно точки (0,0). Если неравенство истинно, то ОДР лежит в полуплоскости, которой принадлежит точка (0,0), если ложно – то в полуплоскости, которая не содержит точку (0,0). ОДР будет являться областью пересечения всех полуплоскостей, задаваемых неравенствами-ограничителями.

В результате получим область допустимых решений, представленную на Рисунке 4.2.

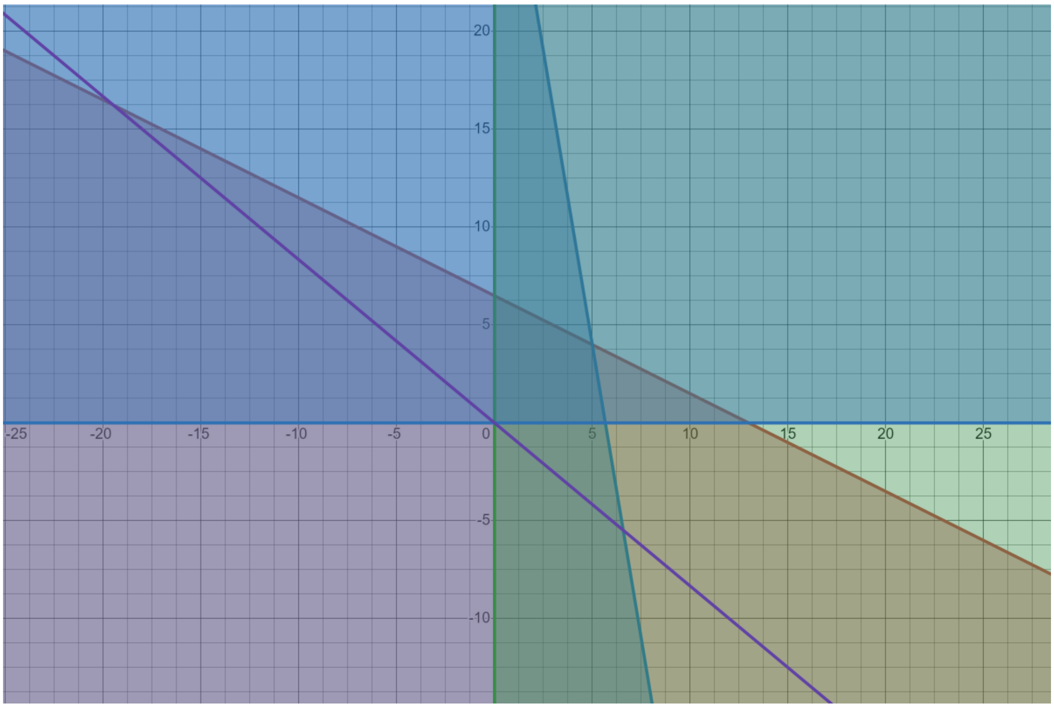


**Рисунок 4.2 – Выделение области допустимых решений**

## 4.6 Максимум функции

Для нахождения максимума функции найдем её градиент по формуле 1.1:

$$\overline{gradf(x)} = \left\{\frac{df(x)}{dx_1}, \frac{df(x)}{dx_2}\right\} \tag{1.1}$$

Для нахождения минимума функции найдем её градиент по формуле 1.2:

$$-\overline{gradf(x)} = \left\{-\frac{df(x)}{dx_1}, -\frac{df(x)}{dx_2}\right\} \tag{1.2}$$

Градиент функции будет равен {2, 1}, а антиградиент функции будет равен {-2, -1}. Изобразим эти вектора на графике (Рисунок 4.3).

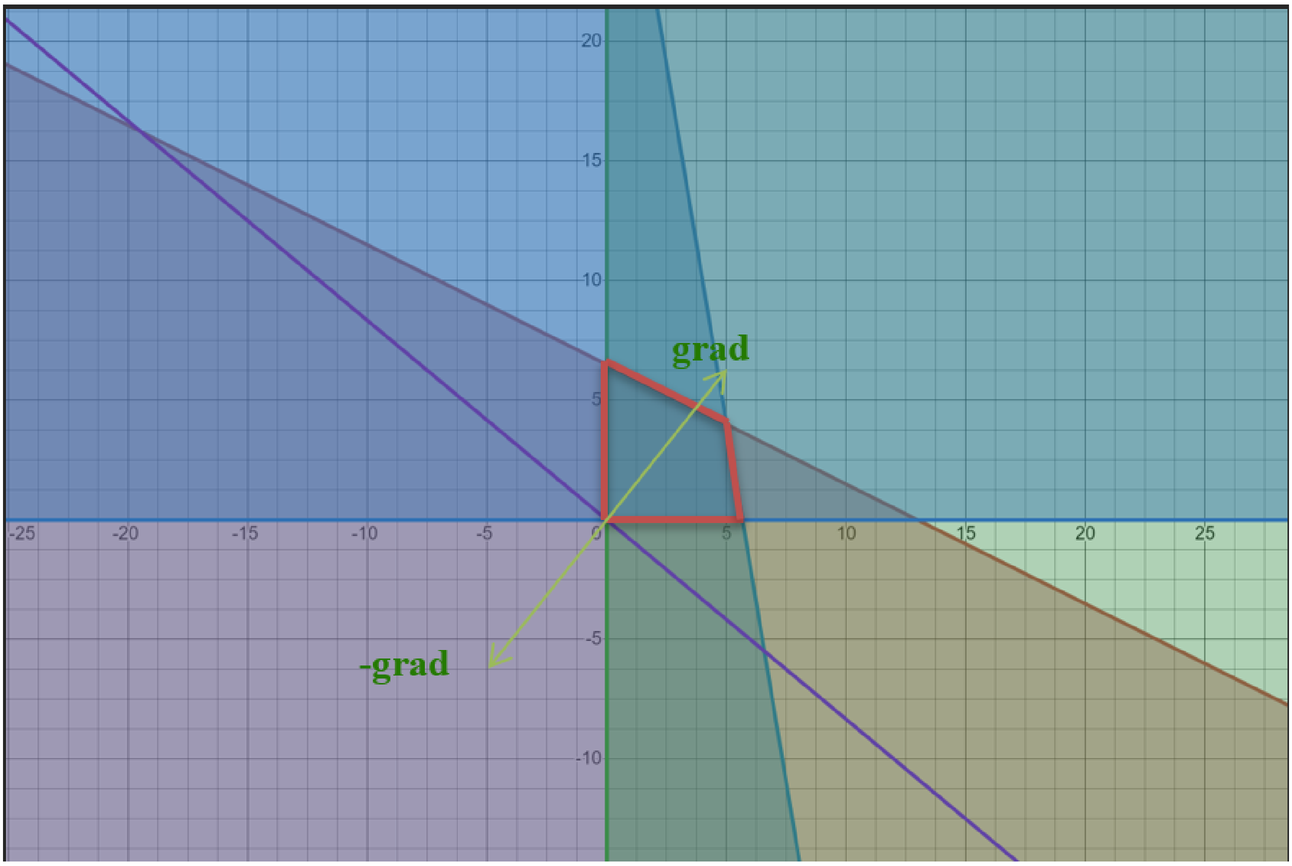


**Рисунок 4.3 – Построение векторов градиента и антиградиента**

Теперь начинаем мысленно сдвигать прямую целевой функции в направлении градиента, и определяем последнюю точку ОДР, которая лежит на пути прямой. Найдем её координаты:

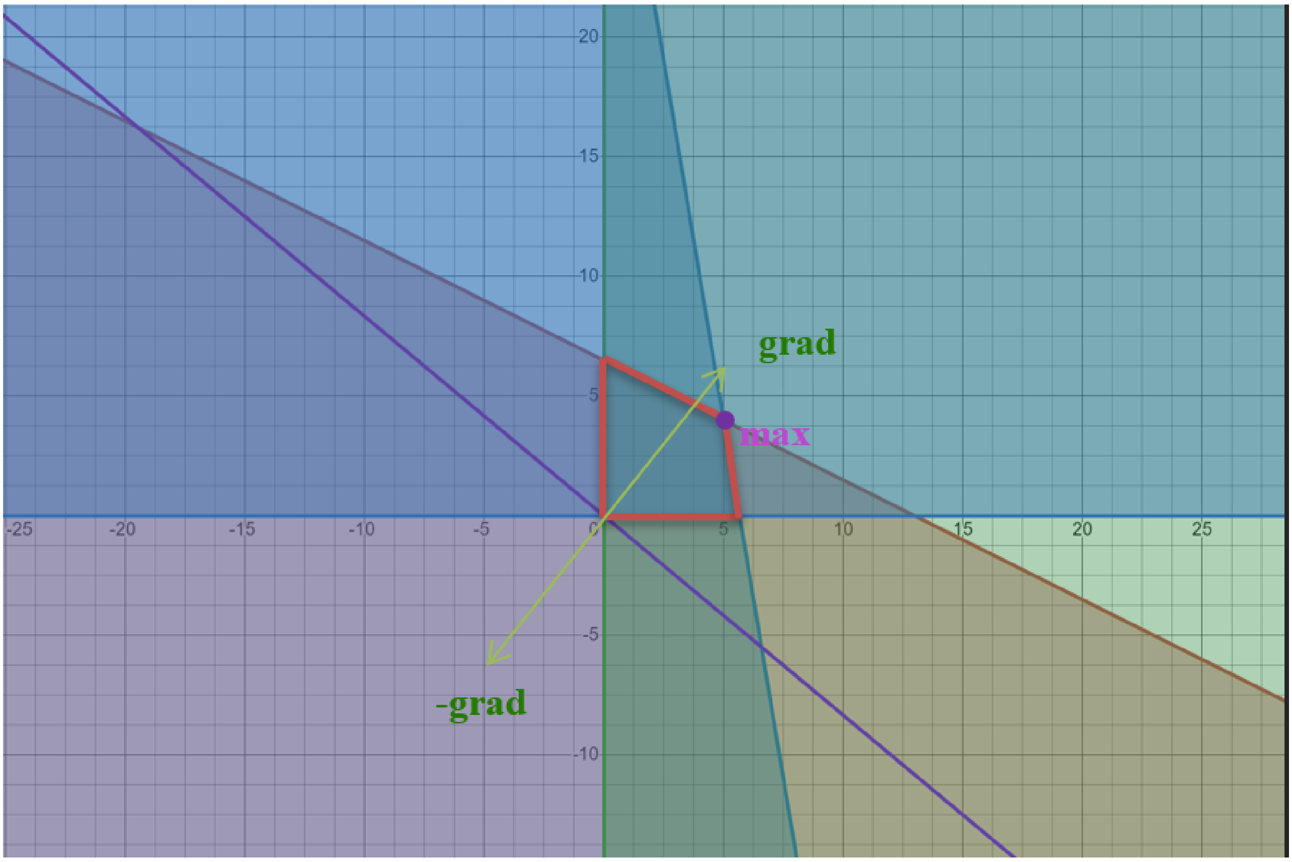

**Рисунок 4.4 – Точка максимума функции**

Найдем значение функции в точке максимума.

Координаты точки максимума: {5;4}

Подставим координаты найденных точек (максимума) в систему уравнений и убедимся, что точки принадлежат области ОДР:

$$\begin{cases} x_1 + 2x_2 \le 13 \\ 6x_1 + x_2 \le 34 \\ x_1, x_2 \ge 0 \end{cases}$$

Получим значение равное F(x)max = 48

## 4.7 Минимум функции

Для нахождения минимума функции будем перемещать прямую в сторону антиградиента. Отметим на графике найденную точку (Рисунок 4.5).

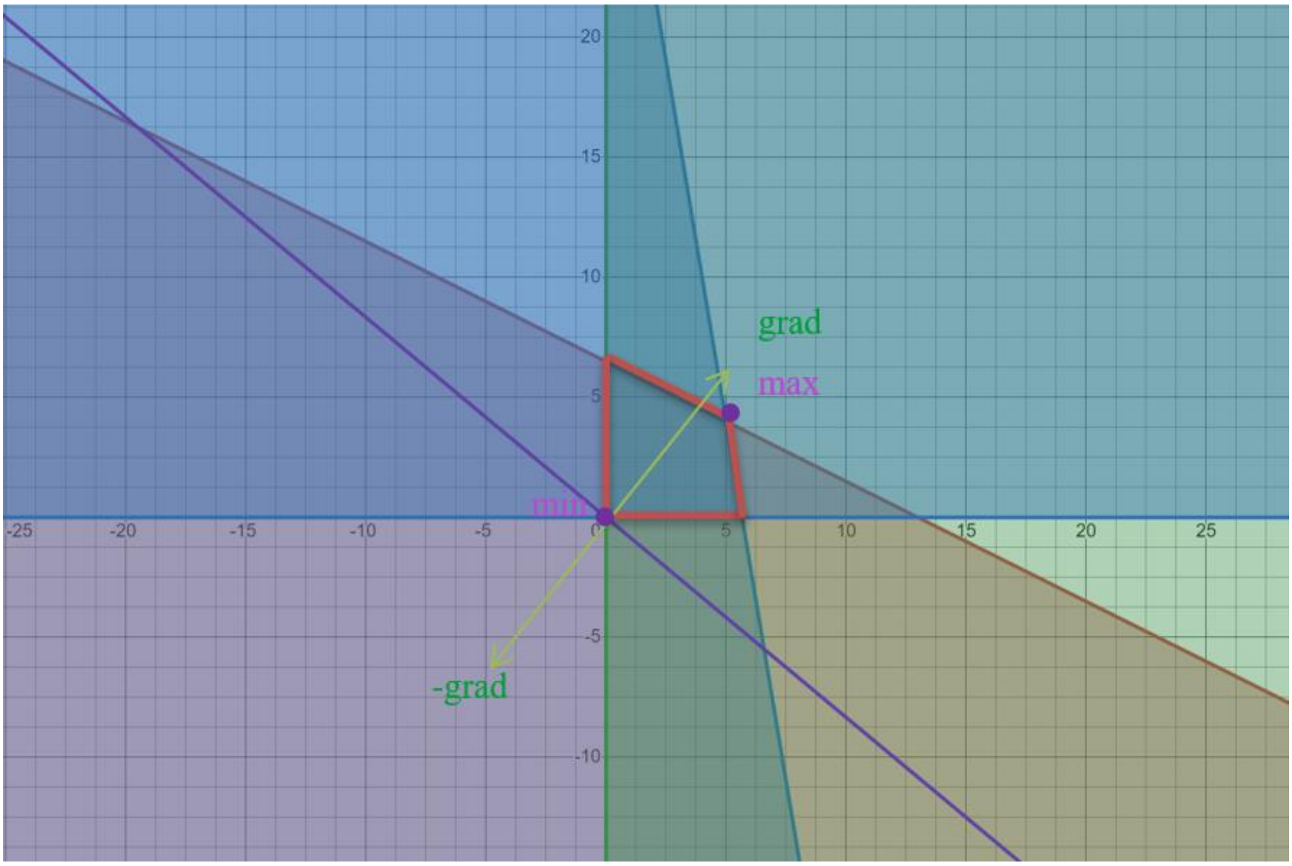


**Рисунок 4.5 – Точка минимума функции**

Найдем координаты точки минимума:

Координаты точки минимума: {0, 0}

Подставив координаты найденных точек (минимума) в систему уравнения и убедимся, что точки принадлежать к области ОДР:

$$\begin{cases} x_1 + 2x_2 \leq 13 \\ 6x_1 + x_2 \leq 34 \\ x_1, x_2 \geq 0 \end{cases}$$

Получим результат F(x)min = 0

Ответ:

F(x)max = 48.

F(x)min = 0.

# 5 СИМПЛЕКСНЫЙ МЕТОД

## 5.1 Постановка задачи

Решить прямую ЗЛП с помощью симплексного метода и обратную с помощью теорем двойственности. Определить интервалы устойчивости.

***Задача.*** Фабрика может производить тарелки и кружки. На производство тарелки идет 5 единиц материала, на производство кружки – 20 единиц (керамики). Тарелка требует 10 человеко-часа, кружка – 15. На производство тарелки тратится 0,5 кВатт электроэнергии, кружки – 0,3. Расходы при производстве тарелки равны 1 рубль, а кружки – 2 рубля. Имеется 400 единиц материала и 450 человеко-часов, 25 кВатт энергии и объем накладных расходов равен 300 рублей. Эти данные представлены в таблице 5.1.

*Таблица 5.1. Исходные данные задачи.*

| Ресурс | Товар | | Объем ресурса |
|---|---|---|---|
| | Тарелки | Кружки | |
| Материал | 5 | 20 | 400 |
| Человеко-часы | 10 | 15 | 450 |
| Электроэнергия | 0,5 | 0,3 | 25 |
| Расходы | 1 | 2 | 300 |

Прибыль при производстве тарелки – 20 рублей, при производстве кружки – 50 рублей. Сколько надо сделать тарелок и кружек, чтобы получить максимальную прибыль?

## 5.2 Математическая модель задачи

Пусть x1 – тарелки, x2 – кружки. Максимальный выпуск продукции составит $20x_1 + 50x_2$.

Ограничения задачи:

$$\begin{cases} 5x_1 + 20x_2 \leq 400 \\ 10x_1 + 15x_2 \leq 450 \\ 0{,}5x_1 + 0{,}3x_2 \leq 25 \\ x_1 + 2x_2 \leq 300 \\ x_1, x_2 \geq 0 \end{cases}$$

Таким образом, переходим к задаче линейного программирования:

$$f(x) = 20x_1 + 50x_2 \rightarrow max$$

$$\begin{cases} 5x_1 + 20x_2 \leq 400 \\ 10x_1 + 15x_2 \leq 450 \\ 0{,}5x_1 + 0{,}3x_2 \leq 25 \\ x_1 + 2x_2 \leq 300 \\ x_1, x_2 \geq 0 \end{cases}$$

Приведем задачу к канонической форме. Для этого в левые части ограничений вводим дополнительные переменные: x3 ≥ 0, x4 ≥ 0, x5 ≥ 0, x6 ≥ 0. Эти переменные выбираются так, чтобы они обращали неравенства в равенства.

$$\begin{cases} 5x_1 + 20x_2 + \ x_3 = 400 \\ 10x_1 + 15x_2 + \ x_4 = 450 \\ 0{,}5x_1 + 0{,}3x_2 + \ x_5 = 25 \\ x_1 + 2x_2 + \ x_6 = 300 \\ x_1, x_2 \geq 0 \end{cases}$$

$$f(x) = 20x_1 + 50x_2 + 0x3 \ + \ 0x4 + 0x5 + 0x6$$

Построим начальную симплекс-таблицу. Запишем систему в векторной форме:

$$\mathrm{A1x1} \ + \ \mathrm{A2x2} \ + \mathrm{A3x3} \ + \ \mathrm{A4x4} \ + \ \mathrm{A5x5} \ + \ \mathrm{A6x6} \ = \ \mathrm{A0},$$

$$\mathrm{A1} = \begin{pmatrix} 5 \\ 10 \\ 0{,}5 \\ 1 \end{pmatrix}, \mathrm{A2} = \begin{pmatrix} 20 \\ 15 \\ 0{,}3 \\ 2 \end{pmatrix}, \mathrm{A3} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \mathrm{A4} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}, \mathrm{A5} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}, \mathrm{A6} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix},$$

$$\mathrm{A0} = \begin{pmatrix} 400 \\ 450 \\ 25 \\ 300 \end{pmatrix}$$

Векторы $A3$, $A4$, $A5$, $A6$ являются линейно независимыми единичными векторами 3х-мерного пространства и образуют базис этого пространства.

Поэтому за базисные переменные выбираем переменные $x3$, $x4$, $x5$, $x6$. Небазисными переменными являются $x1$, $x2$. Разложение позволяет найти первое базисное допустимое решение.

Для этого свободные переменные $x1$, $x2$ приравниваем нулю. В результате получим разложение

$$\mathrm{A3x3} + \mathrm{A4x4} + \mathrm{A5x5} + \mathrm{A6x6} = \mathrm{A0},$$

Которому соответствует первоначальный опорный план

$$\mathrm{x}^{(0)} = (\mathrm{x1}, \mathrm{x2}, \mathrm{x3}, \mathrm{x4}, \mathrm{x5}, \mathrm{x6}) = (0{,}0{,}0{,}0{,}20{,}50),$$

$$f\left(x^{(0)}\right) = 0.$$

Для проверки плана $x^{(0)}$ на оптимальность построим первую симплекс-таблицу. Введем в рассмотрение вектор коэффициентов целевой функции при базисных переменных.

$$\overline{C_B} = (c3, c4, c5, c6)^T = (0{,}0{,}0{,}0)^T.$$

В левый столбец Таблицы 1.2 запишем переменные $x3$, $x4$, $x5$, $x6$ образующие базис, в верхней строке – небазисные переменные $x1$, $x2$. В строке $c_j$ запишем коэффициенты целевой функции, соответствующие небазисным переменным c1 = 20, c2 = 50. В столбце $\overline{C_B}$ запишем коэффициенты целевой функции, соответствующие базисным переменным Столбец, определяемый переменной $x1$, состоит из коэффициентов вектора $\overline{A_1}$. Аналогично, столбец, определяемый переменной $x2$, состоит из коэффициентов вектора $\overline{A_2}$. Крайний правый столбец заполняется элементами столбца $\overline{A_0}$, в нем же в результате вычислений получаем оптимальный план.

Заполнение f-строки (Таблица 1.3). Найдем относительные оценки Δ1, Δ2 и значение целевой функции $Q$.

$$\Delta_1 = (\overline{C_B} * \overline{A_1}) - c_1 = 0 * 5 + 0 * 10 + 0 * 0{,}5 + 0 * 1 - 20 = -20;$$

$$\Delta_2 = (\overline{C_B} * \overline{A_2}) - c_2 = 0 * 20 + 0 * 15 + 0 * 0{,}3 + 0 * 2 - 50 = -50;$$

$$\mathrm{Q} = (\overline{C_B} * \overline{A_0}) = 0 * 20 + 0 * 50 = 0.$$

*Таблица 5.2 – Начальная симплекс-таблица задачи о максимальном доходе*

| | $c_j$ | 20 | 50 | |
|---|---|---|---|---|
| $\overline{C_B}$ | | X1 | X2 | $\overline{A_0}$ |
| 0 | X3 | 5 | 20 | 400 |
| 0 | X4 | 10 | 15 | 450 |
| 0 | X5 | 0,5 | 0,3 | 25 |
| 0 | X6 | 1 | 2 | 300 |
| | f | | | |
| | | $\Delta_1$ | $\Delta_2$ | Q |

*Таблица 5.3 – Заполнение f-строки*

| | $c_j$ | 20 | 50 | | |
|---|---|---|---|---|---|
| $\overline{C_B}$ | | X1 | X2 | $\overline{A_0}$ | |
| 0 | X3 | 5 | 20 | 400 | 400 / 20 = 20 *min* |
| 0 | X4 | 10 | 15 | 450 | 450 / 15 = 30 |
| 0 | X5 | 0,5 | 0,3 | 25 | 25 / 0,3 = 32 |
| 0 | X6 | 1 | 2 | 300 | 300 / 2 = 150 |
| | f | -20 | -50 | 0 | |
| | | $\Delta_1$ | $\Delta_2$ | Q | |

Для оптимальности опорного решения в задаче на максимум требуется выполнение не отрицательности всех относительных оценок $\Delta i \geq 0$. Так как оценки $\Delta 1= -20$ и $\Delta 2= -50$ в f-строке отрицательны, то это свидетельствуют о возможности улучшения полученного решения. Наибольшая по модулю отрицательная оценка $\Delta 2= -50$. В базис будет включена соответствующая ей небазисная переменная $x2$. Составим отношения свободных членов к положительным элементам разрешающего столбца. Данные отношения приведены справа от таблицы. Наименьшему частному соответствует строка с переменной $x5$. Эта переменная исключается из базиса. В Таблице 1.3 разрешающий столбец и разрешающая строка выделены. Разрешающим элементом является число $a13 = 20$.

Далее построим новую симплекс-таблицу. Ниже поэтапно демонстрируется процесс заполнения новой симплекс-таблицы (Таблицы 5.4, 5.5).

*Таблица 5.4 – Новая симплекс-таблица*

| $\overline{C_B}$ | $c_j$ | 20 | 0 | |
|---|---|---|---|---|
| | | X1 | X3 | $\overline{A}_0$ |
| 50 | X2 | | 0,05 | |
| 0 | X4 | | | |
| 0 | X5 | | | |
| 0 | X6 | | | |
| | f | | | |
| | | $\Delta_1$ | $\Delta_2$ | Q |

В Таблице 1.4 переменные $x2$ и $x3$ меняются местами вместе с коэффициентами $cj$. Разрешающий элемент заменяется на обратный. В Таблице 1.5 элементы разрешающей строки делятся на разрешающий элемент. Элементы разрешающего столбца делятся на разрешающий элемент и меняют знак.

*Таблица 5.5 – Симплекс преобразования*

| $\overline{C_B}$ | $c_j$ | 20 | 0 | |
|---|---|---|---|---|
| | | X1 | X3 | $\overline{A}_0$ |
| 50 | X2 | 0.25 | 0.05 | 20 |
| 0 | X4 | | -0.75 | |
| 0 | X5 | | -0.02 | |
| 0 | X6 | | -0.1 | |
| | f | | 2.5 | |
| | | $\Delta_1$ | $\Delta_2$ | Q |

*Таблица 5.6 – Итерация 1*

| $\overline{C_B}$ | $c_j$ | 20 | 0 | | |
|---|---|---|---|---|---|
| | | X1 | X3 | $\overline{A}_0$ | |
| 50 | X2 | 0.25 | 0.05 | 20 | 20 / 0.25 = *80* |
| 0 | X4 | 6.25 | -0.75 | 150 | 150 / 6.25 = 24 *min* |
| 0 | X5 | 0.42 | -0.02 | 19 | 19 / 0.42 = 45.23 |
| 0 | X6 | 0.5 | -0.1 | 260 | 260 / 0.5 = 520 |
| | f | -7.5 | 2.5 | 1000 | |
| | | $\Delta_1$ | $\Delta_2$ | Q | |

Остальные элементы (Таблица 5.6) рассчитываются по «правилу прямоугольника».

$$a_{21} = \frac{(20*10)-(5*15)}{20} = 6.25;\ a_{31} = \frac{(20*0.5)-(5*0.3)}{20} = 0.42;$$

$$a_{41} = \frac{(20*1)-(5*2)}{20} = 0.5;\ \Delta_1 = \frac{(20*(-20))-(5*(-50))}{20} = -7.5;$$

$$a_{23} = \frac{(20*450)-(400*15)}{20} = 150;\ a_{33} = \frac{(20*25)-(400*0.3)}{20} = 19;$$

$$a_{43} = \frac{(20*300)-(400*2)}{20} = 260;$$

Базисное решение, которое дает последняя таблица

$$x^{(1)} = (x1, x2, x3, x4, x5, x6) = (0, 150, 20, 0, 19, 260),$$

$$f\left(x^{(1)}\right) = (\overline{C_B} * \overline{A_0}) = 50*20 + 0*150 + 0*19 + \ 0*260 = 1000.$$

Это решение не является оптимальным, так как в f-строке имеется отрицательная оценка Δ1.

В Таблице 5.6 разрешающий столбец и разрешающая строка выделены. Разрешающим элементом является число $a21$ = 6.25.

Построим новую симплекс-таблицу. Ниже поэтапно демонстрируется процесс заполнения новой симплекс-таблицы (Таблицы 5.7, 5.8).

*Таблица 5.7 – Новая симплекс-таблица*

| | $c_j$ | 0 | 0 | |
|---|---|---|---|---|
| $\overline{C_B}$ | | X4 | X3 | $\overline{A_0}$ |
| 50 | X1 | | | |
| 20 | X2 | 0.16 | | |
| 0 | X5 | | | |
| 0 | X6 | | | |
| | f | | | |
| | | $\Delta_1$ | $\Delta_2$ | Q |

В Таблице 5.7 переменные $x1$ и $x3$ меняются местами вместе с коэффициентами $cj$. Разрешающий элемент заменяется на обратный. В Таблице 5.8 элементы разрешающей строки делятся на разрешающий элемент. Элементы разрешающего столбца делятся на разрешающий элемент и меняют знак.

*Таблица 5.8 – Симплекс преобразования*

| | | $c_j$ | 0 | 0 | |
|---|---|---|---|---|---|
| | $\overline{C_B}$ | | X4 | X3 | $\overline{A_0}$ |
| | 50 | X1 | -0.04 | | |
| | 20 | X2 | 0.16 | -0.12 | 24 |
| | 0 | X5 | -0.07 | | |
| | 0 | X6 | -0.08 | | |
| | | f | 1.2 | | |
| | | | $\Delta_1$ | $\Delta_2$ | Q |

*Таблица 5.9 – Итерация 2*

| | | $c_j$ | 0 | 0 | |
|---|---|---|---|---|---|
| | $\overline{C_B}$ | | X4 | X3 | $\overline{A_0}$ |
| | 50 | X1 | -0.04 | 0.08 | 14 |
| | 20 | X2 | 0.16 | -0.12 | 24 |
| | 0 | X5 | -0.07 | 0.04 | 8.8 |
| | 0 | X6 | -0.08 | -0.04 | 248 |
| | | f | 1.2 | 1.6 | 1180 |
| | | | $\Delta_1$ | $\Delta_2$ | Q |

Остальные элементы (Таблица 5.9) рассчитываются по «правилу прямоугольника».

$$a_{12} = \frac{(6.25 * 0.05) - \left(0.25 * (-0.75)\right)}{6.25} = 0.08;$$

$$a_{13} = \frac{(6.25 * 20) - (150 * 0.25))}{6.25} = 14;$$

$$a_{32} = \frac{\left(6.25 * (-0.02)\right) - \left(0.42 * (-0.75)\right)}{6.25} = 0.04;$$

$$a_{33} = \frac{(6.25 * 19) - (0.42 * 150)}{6.25} = 8.8;$$

$$a_{42} = \frac{(6.25 * (-0.1)) - \left(0.5 * (-0.75)\right)}{6.25} = -0.04;\ a_{43} = \frac{(6.25 * 260) - (0.5 * 150)}{6.25} = 248;$$

$$\Delta_2 = \frac{(6.25 * 2.5) - \left(-7.5 * (-0.75)\right)}{6.25} = 2.5;$$

Если в последней таблице f-строке не содержит отрицательных оценок, то это свидетельствует об оптимальности полученного решения:

Подставляем базисное решение, которое дает последняя таблица

$$x^{(2)} = (x1, x2, x3, x4, x5, x6) = (14, 24, 0,0, 8.8, 248),$$

Где n – количество итераций

$$f\left(x^{(2)}\right) = (\overline{C_B} * \overline{A_0}) = 14 * 50 + 24 * 20 + 0 * 8.8 + 0 * 248 = 1180.$$

Проверим решение по «правилу прямоугольника».

$$f_{max} = Q = (\overline{C_B} * \overline{A_0}) = \frac{(6.25 * 1000) - (-7.5 * 150)}{6.25} = 1180.$$

Таким образом, предприятие должно выпускать в течении недели $x1$ = 24 шт. тарелок и $x2$ = 14 шт. кружек. Тогда предприятие получит программу по максимальному доходу - 1180 [шт.].

## 5.3 Результат работы программы

Так же, ссылаясь на Приложение Г, реализован симплексный метод в виде программы и для самопроверки можно оценить правильность расчётов. На рисунке 5.1 приведён результат работы программы.

```
 Итерация : 1
ind A0          x_1     x_2     b_1     b_2     b_3     b_4
F       1000.0  -7.5    0.0     2.5     0.0     0.0     0.0
x_2     20.0    0.25    1.0     0.05    0.0     0.0     0.0
x_4     150.0   6.25    0.0     -0.75   1.0     0.0     0.0
x_5     19.0    0.42    0.0     -0.02   0.0     1.0     0.0
x_6     260.0   0.5     0.0     -0.1    0.0     0.0     1.0

 Разрешающий столбец: 1
 Разрешающая строка: 2
 Разрешающий элемент:  6.25

=====================


 Итерация : 2
ind A0          x_1     x_2     b_1     b_2     b_3     b_4
F       1180.0  0.0     0.0     1.6     1.2     0.0     0.0
x_2     14.0    0.0     1.0     0.08    -0.04   0.0     0.0
x_1     24.0    1.0     0.0     -0.12   0.16    0.0     0.0
x_5     8.8     0.0     0.0     0.04    -0.07   1.0     0.0
x_6     248.0   0.0     0.0     -0.04   -0.08   0.0     1.0

----------------------------------


 Финальная таблица была получена за 2 итерации
ind A0          x_1     x_2     b_1     b_2     b_3     b_4
F       1180.0  0.0     0.0     1.6     1.2     0.0     0.0
x_2     14.0    0.0     1.0     0.08    -0.04   0.0     0.0
x_1     24.0    1.0     0.0     -0.12   0.16    0.0     0.0
x_5     8.8     0.0     0.0     0.04    -0.07   1.0     0.0
x_6     248.0   0.0     0.0     -0.04   -0.08   0.0     1.0


 Коэффициенты:
 [24. 14.]
```

**Рисунок 5.1 – Результат работы программы**

# ЗАКЛЮЧЕНИЕ

В процессе написания курсовой работы были рассмотрены различные примеры принятия решений. Было изучено, как правильное принятие решений может привести к успеху в различных аспектах.

В заключение можно сказать, что курсовая работа по теории принятия решений помогла понять, что этот процесс является сложным и многоступенчатым. Он включает определение проблемы, сбор и анализ информации, оценку альтернатив, выбор наилучшего варианта и реализацию решения. Каждый из этих этапов требует тщательного подхода и внимательного анализа множества факторов.

В ходе работы удалось изучить и применить различные методы и инструменты, такие как метод Парето, ELECTRE II, метод анализа иерархий, графический метод, симплексный метод и анализ двойственной задачи. Было показано, что выбор метода зависит от конкретной ситуации и специфики задачи. Например, метод Парето полезен для нахождения оптимальных решений по нескольким критериям, а метод анализа иерархий помогает структурировать сложные решения.

Курсовая работа также позволила разработать стратегии для достижения различных целей и улучшить навыки принятия решений. Понимание теории и применение различных методов на практике способствуют более обоснованному и эффективному принятию решений в будущем.

Таким образом, изучение теории принятия решений и различных методов, применяемых в этой области, способствует повышению качества принимаемых решений, что в свою очередь ведет к достижению успеха в различных сферах деятельности.

# СПИСОК ИНФОРМАЦИОННЫХ ИСТОЧНИКОВ


1. Болотова Л. С. Многокритериальная оптимизация. Болотова Л. С., Сорокин А. Б. [Электронный ресурс] / Метод. указания по вып. курсовой работы — М.: МИРЭА, 2015.
2. Сорокин А. Б. Методы оптимизации: гибридные генетические алгоритмы. Сорокин А. Б. [Электронный ресурс] / Учебно-метод. пособие — М.: МИРЭА, 2016.
3. Сорокин А. Б. Линейное программирование: практикум. Сорокин А. Б., Бражникова Е. В., Платонова О. В. [Электронный ресурс] / Учебно-метод. пособие — М.: МИРЭА, 2017.
4. Видео о графической задаче — URL: https://youtu.be/xIQOpfuLCeI.
5. Подробный разбор симплекс-метода — URL: https://habr.com/ru/articles/474286/.
6. Статья о симплекс-методе — URL: https://ru.wikipedia.org/wiki/Симплекс-метод.

# ПРИЛОЖЕНИЯ

Приложение А – Код реализации метода Парето на языке Python.

Приложение Б – Код реализации метода Электра II на языке Python.

Приложение В – Код реализации метода анализа иерархий на языке Python.

Приложение Г – Код реализации симплексного метода на языке Python.

# Приложение А

Код реализации метода Парето на языке Python.

*Листинг А - Код реализации метода Парето*

```
import pandas as pd
import numpy as np

df = pd.DataFrame({
    'Средний чек (руб.)': [2300, 1800, 1700, 1200, 800, 1950, 500, 2600, 1400,
3500],
    'Удалённость локации (км)': [4.30, 2.30, 2.70, 1.60, 9.30, 2, 11.10, 2.30,
5.80, 3.20],
    'Количество услуг': [16, 17, 20, 13, 8, 16, 4, 22, 16, 19],
    'Рейтинг (от 1 до 5)': [4.80, 5, 4.90, 4.40, 3.80, 5, 2.70, 4.90, 4.80,
4.70]
    })

df.index = ['A1', 'A2', 'A3', 'A4', 'A5', 'A6', 'A7', 'A8', 'A9', 'A10']
print(df)

df['Средний чек (руб.)'] = 1 / df['Средний чек (руб.)']
df['Удалённость локации (км)'] = 1 / df['Удалённость локации (км)']

# Создаем массив для хранения результатов попарного сравнения
arr1 = np.zeros((10, 10), dtype=object)

# Попарное сравнение альтернатив
for i in range(10):
    for j in range(i + 1, 10):
        arr = df.iloc[i].values >= df.iloc[j].values  # Сравнение значений
всех столбцов для альтернативы i и j
        check = all(arr)  # Проверка, что все критерии для i лучше или равны j
        arr2 = df.iloc[i].values <= df.iloc[j].values  # Сравнение значений
всех столбцов для альтернативы i и j
        check2 = all(arr2)  # Проверка, что все критерии для i хуже или равны
j
        # Запись результата в массив arr1
        if check:
            arr1[j, i] = 'A' + str(i + 1)
        elif check2:
            arr1[j, i] = 'A' + str(j + 1)
        else:
            arr1[j, i] = 'н'

# Создаем новый DataFrame для результатов попарного сравнения
df_ = pd.DataFrame(arr1, columns=['A1', 'A2', 'A3', 'A4', 'A5', 'A6', 'A7',
'A8', 'A9', 'A10'],
                   index=['A1', 'A2', 'A3', 'A4', 'A5', 'A6', 'A7', 'A8',
'A9', 'A10'])

print("\n Таблица Попарное сравнение альтернатив: ")
print(df_)

print("\n Вывод парето-оптимальных альтернатив")
print(df.iloc[[1, 2, 5, 7]])

df['Средний чек (руб.)'] = 1 / df['Средний чек (руб.)']
df['Удалённость локации (км)'] = 1 / df['Удалённость локации (км)']
```

*Продолжение листинга А - Код реализации метода Парето*

```
print("\n Результат указания верхней/нижней границы: ('Средний чек (руб.)' >=
1700 , 'Удалённость локации (км)' < 4.30)")
print(df[(df['Средний чек (руб.)'] >= 1700) & (df['Удалённость локации (км)']
< 4.30)])

print(
    "\n Результат отбора вариантов, удовлетворяющих заданным критериям:
главный критерий: \n Средний чек (руб.) >= 1200, Рейтинг (от 1 до 5) >= 4.6,
Удалённость локации (км) < 2.70")
print(df[(df['Средний чек (руб.)'] >= 1200) & (df['Рейтинг (от 1 до 5)'] >=
4.6) & (df['Удалённость локации (км)'] < 2.70)])

df['Средний чек (руб.)'] = 1 / df['Средний чек (руб.)']
df['Удалённость локации (км)'] = 1 / df['Удалённость локации (км)']


def lex_optimization(df):
    max_crit = df['Количество услуг'].max()  # Нахождение максимального
значения в столбце 'Количество услуг'
    optimal_df = df[df['Количество услуг'] == max_crit]  # Фильтрация данных
по максимальному значению количества услуг

    if len(optimal_df) == 1:  # Проверка, если найден только один оптимальный
вариант
        return optimal_df

    next_crit = optimal_df['Удалённость локации (км)'].max()  # Нахождение
следующего критерия - максимального значения удаленности локации
    optimal_df = optimal_df[optimal_df['Удалённость локации (км)'] ==
next_crit]  # Дополнительная фильтрация данных по максимальному значению
удаленности локации

    return optimal_df


result = lex_optimization(df)

print("\n Результат лексикографической оптимизации: (Самая важная: Количество
услуг)")
print(result)
```

## Приложение Б

Код реализации метода Электра II на языке Python.

*Листинг Б - Код реализации метода Электра II*

```
a = [
    [15,15,10,5],
    [10,5,10,5],
    [10,10,15,5],
    [5,5,10,4],
    [5,15,5,3],
    [15,5,10,5],
    [5,15,5,2],
    [15,5,15,5],
    [10,15,10,5],
    [15,10,15,4]
]

b = [' x '] * 10

c = [5, 4, 4, 5]

for i in range(10):
    b[i] = [' x '] * 10

countdominant = 0
countdominanted = 0

for i in range(10):
    for m in range(i + 1, 10):
        for j in range(4):
            if j == 0 or j == 1:
                if a[i][j] < a[m][j]:
                    countdominant += c[j]
                elif a[i][j] > a[m][j]:
                    countdominanted += c[j]
            else:
                if a[i][j] > a[m][j]:
                    countdominant += c[j]
                elif a[i][j] < a[m][j]:
                    countdominanted += c[j]
        if countdominant != 0 and countdominanted == 0:
            b[i][m] = 'inf'
            b[m][i] = ' - '
        elif countdominant == 0 and countdominanted != 0:
            b[m][i] = 'inf'
            b[i][m] = ' - '
        else:
            if countdominant == 0 or countdominanted == 0:
                b[i][m] = ' - '
                b[m][i] = ' - '
            else:
                ratio = countdominant / countdominanted
                if ratio == 1:
                    b[i][m] = ' - '
                    b[m][i] = ' - '
                elif ratio < 1:
                    b[i][m] = ' - '
                    b[m][i] = str(round(1 / ratio, 2))
                else:
                    b[i][m] = str(round(ratio, 2))
```

*Продолжение листинга Б - Код реализации метода Электра II 2*

```
                        b[m][i] = ' - '
                countdominant = 0
                countdominanted = 0

for i in b:
    print(i)

alternative_count = [0] * 10

for i in range(10):
    for j in range(10):
        if b[i][j] != ' - ' and b[i][j]!= ' x ':
            alternative_count[i] += 1

# Сортировка по количеству вхождений
sorted_alternative = sorted(range(len(alternative_count)), key=lambda k:
alternative_count[k])

print("\nЛучшие Альтернативы:")
for i in range(10):
    if i == 0:
        print(f"Альтернатива {sorted_alternative[i] + 1}", end="")
    else:
        print(f" <- Алтернатива {sorted_alternative[i] + 1}", end="")
print()
```

## Приложение В

Код реализации метода анализа иерархий на языке Python

*Листинг В - Код реализации метода анализа иерархий*

```
k = [
    [1, 1/3, 4, 7],
    [3, 1, 3, 7],
    [1/4, 1/3, 1, 4],
    [1/7, 1/7, 1/4, 1],
]
a = [
    [1, 1/3, 3, 7],
    [3, 1, 3, 7],
    [1/3, 1/3, 1, 5],
    [1/7, 1/7, 1/5, 1],
]
b = [
    [1, 1/3, 1/2, 3],
    [3, 1, 3, 3],
    [2, 1/3, 1, 4],
    [1/3, 1/3, 1/4, 1],
]
c = [
    [1, 1/3, 3, 1/7],
    [3, 1, 5, 1/3],
    [1/3, 1/5, 1, 1/7],
    [7, 3, 7, 1],
]
d = [
    [1, 3, 1/3, 1],
    [1/3, 1, 1/5, 1/3],
    [3, 5, 1, 3],
    [1, 3, 1/3, 1],
]
def multiply_elements_and_raise(matrix):
    global summ_multiplied_and_raised
    multiplied_and_raised = []
    for row in matrix:
        row_product = 1
        for element in row:
            row_product *= element
        row_result = row_product ** 0.25
        row_result = round(row_result, 3)
        multiplied_and_raised.append(row_result)
        summ_multiplied_and_raised = round(sum(multiplied_and_raised), 3)

    divided_by_sum = []
    for element in multiplied_and_raised:
        division_result = round(element / summ_multiplied_and_raised, 3)
        divided_by_sum.append(division_result)
    return divided_by_sum
result_k = multiply_elements_and_raise(k)
result_a = multiply_elements_and_raise(a)
result_b = multiply_elements_and_raise(b)
result_c = multiply_elements_and_raise(c)
result_d = multiply_elements_and_raise(d)

combined_matrix = [list(row) for row in zip(result_a, result_b, result_c,
result_d)]
```

*Продолжение листинга В - Код реализации метода анализа иерархий*

```
combined_matrix_transposed = [[combined_matrix[j][i] for j in
range(len(combined_matrix))] for i in
                              range(len(combined_matrix[0]))]

array = result_k
array2 = combined_matrix_transposed
import numpy as np

array_np = np.array(array)
array2_np = np.array(array2)

result = np.dot(array_np, array2_np)

print("Результат:")
print("     W1         W2       W3        W4")
print(result)
```

# Приложение Г

## Код реализации симплексного метода на языке Python

*Листинг Г - Код реализации симплексного метода*

```
import numpy as np


class LinearModel:
    # Инициализация параметров модели
    def __init__(self, A=np.empty([0, 0]), b=np.empty([0, 0]), c=np.empty([0, 
0]), minmax="MAX"):
        self.A = A  # Матрица коэффициентов ограничений
        self.b = b  # Вектор правой части ограничений
        self.c = c  # Вектор коэффициентов целевой функции
        self.x = [float(0)] * len(c)  # Начальное решение (все переменные 
равны нулю)
        self.minmax = minmax  # Тип оптимизации (минимизация или максимизация)
        self.printIter = True  # Флаг для печати итераций
        self.optimalValue = None  # Оптимальное значение целевой функции
        self.transform = False  # Флаг для преобразования модели

    def addA(self, A):  # Установка матрицы коэффициентов ограничений
        self.A = A

    def addB(self, b):  # Установка вектора правой части ограничений
        self.b = b

    def addC(self, c):  # Установка вектора коэффициентов целевой функции
        self.c = c
        self.transform = False

    def setObj(self, minmax):  # Установка типа оптимизации
        self.minmax = minmax
        self.transform = False

    def setPrintIter(self, printIter):  # Установка флага для печати итераций
        self.printIter = printIter

    def printSoln(self):  # Печать решения и оптимального значения
        print(" Коэффициенты: ")
        print("", self.x)
        print("\n Оптимальное значение: ")
        print("", self.optimalValue)

    def getTableau(self):  # Создание симплекс-таблицы
        num_var = len(self.c)  # Получение количества переменных
        num_slack = len(self.A)  # Получение количества ограничений
        # Создание верхней строки таблицы
        t1 = np.hstack(([None], [0], self.c, [0] * num_slack))


        # Создание базисных переменных и расширение матрицы A, если необходимо
        basis = np.array([0] * num_slack)  # Создание массива для базисных 
переменных
        for i in range(0, len(basis)):
            basis[i] = num_var + i  # Установка индексов базисных переменных
        A = self.A
        if not ((num_slack + num_var) == len(self.A[0])):
            # Если матрица A не квадратная, добавляем единичную матрицу для 
расширения
```

*Продолжение листинга Г - Код реализации симплексного метода*

```
            B = np.identity(num_slack)
            A = np.hstack((self.A, B))

        # Создание нижних строк таблицы
        t2 = np.hstack((np.transpose([basis]), np.transpose([self.b]), A))

        # Объединение верхней и нижней частей таблицы
        tableau = np.vstack((t1, t2))  # Слияние верхней и нижней частей
таблицы
        tableau = np.array(tableau, dtype='float')  # Преобразование в массив
NumPy
        return tableau  # Возвращение симплекс-таблицы

    def optimize(self): # Оптимизация симплекс-методом
        tableau = self.getTableau()  # Получение симплекс-таблицы

        if self.printIter:
            print(" Стартовая таблица:")
            self.print_table(tableau, True)  # Печать начальной симплекс-
таблицы
        optimal = False  # Флаг для проверки на оптимальность
        iter = 0  # Счетчик итераций
        while 1:
            if self.printIter:
                if iter > 0:
                    print("\n====================\n")
                    print(" Итерация :", iter)
                    self.print_table(tableau, False)  # Печать текущей
симплекс-таблицы
            for profit in tableau[0, 2:]:
                if profit > 0:
                    optimal = False
                    break
                optimal = True
            if optimal:
                break
            n = tableau[0, 2:].tolist().index(np.amax(tableau[0, 2:])) + 2 #
Выбор разрешающего столбца
            minimum = 99999  # Инициализация минимального значения
            r = -1  # Инициализация разрешающей строки
            for i in range(1, len(tableau)):
                if tableau[i, n] > 0:
                    val = tableau[i, 1] / tableau[i, n]
                    if val != 0 and val < minimum:
                        minimum = val  # Обновление минимального значения
                        r = i  # Обновление разрешающей строки
            pivot = tableau[r, n]  # Получение разрешающего элемента
            print("\n Разрешающий столбец:", n - 1)
            print(" Разрешающая строка:", r)
            print(" Разрешающий элемент: ", pivot)
            tableau[r, 1:] = tableau[r, 1:] / pivot  # Деление строки на
разрешающий элемент
            for i in range(0, len(tableau)):
                if i != r:
                    mult = tableau[i, n] / tableau[r, n]  # Вычисление
множителя
                    tableau[i, 1:] = tableau[i, 1:] - mult * tableau[r, 1:]  #
Обновление строк
            tableau[r, 0] = n - 2  # Обновление индекса базисной переменной в
таблице
            iter += 1  # Увеличение счетчика итераций
```

*Продолжение листинга Г - Код реализации симплексного метода 2*

```
        if self.printIter:
            print("\n-----------------------------------\n")
            print(" Финальная таблица была получена за", iter, "итерации")
            self.print_table(tableau, False)  # Печать финальной симплекс-
таблицы
        else:
            print("Решено")
        self.x = np.array([0] * len(self.c), dtype=float)  # Создание массива
для решения
        for key in range(1, (len(tableau))):
            if tableau[key, 0] < len(self.c):
                self.x[int(tableau[key, 0])] = tableau[key, 1]  # Обновление
значений переменных
            self.optimalValue = -1 * tableau[0, 1]  # Установка оптимального
значения

    def print_table(self, tableau, start):  # Функция для печати симплекс-
таблицы
        print("ind A0\t\t ", end="")  # Печать заголовка столбца с индексом и
A0

        for i in range(1, len(self.c) + 1):  # Печать заголовков столбцов
переменных x
            print("x_" + str(i), end="\t ")

        for i in range(1, 5):  # Печать заголовков столбцов правой части
ограничений
            print("b_" + str(i), end="\t ")
        print()  # Переход на новую строку после печати заголовка

        for j in range(0, len(tableau)):  # Перебор строк таблицы
            for i in range(0, len(tableau[0])):  # Перебор элементов в строке
                if not np.isnan(tableau[j, i]):  # Проверка, что элемент не
NaN
                    if i == 0: # Если это первый столбец (индекс базисной
переменной)
                        print('x_' + str(int(tableau[j, i]) + 1), end="\t ")
                    else:
                        if j == 0 and start is False:  # Если это первая
строка и start равно False
                            if round(tableau[j, i], 2) == 0:
                                print(round(tableau[j, i], 2), end="\t ")  #
Если значение округленное до 2 знаков после запятой равно 0
                            else:
                                print((-1) * round(tableau[j, i], 2), end="\t
")  # В противном случае, печать отрицательного значения
                        else:
                            print(round(tableau[j, i], 2), end="\t ")  # Если
не первая строка или start равно True
                else:
                    print('F', end="\t ")  # Если элемент NaN, печать символа
'F' вместо значения
            print()  # Переход на новую строку после печати строки таблицы
if __name__ == '__main__':
    model1 = LinearModel()
    A = np.array(
        [
            [5, 20],
            [10, 15],
            [0.5, 0.3],
            [1, 2]
        ]
    )
```

*Продолжение листинга Г - Код реализации симплексного метода 3*

```
    b = np.array(
        [400, 450, 25, 300]
    )
    c = np.array(
        [20, 50]
    )
    model1.addA(A)
    model1.addB(b)
    model1.addC(c)
    print("\n Дано:")
    print("> A =\n", A, "\n")
    print("> A0 =\n", b, "\n")
    print("> C =\n", c, "\n\n")
    model1.optimize()
    print("\n")
    model1.printSoln()
```