**Оглавление**

[**ВВЕДЕНИЕ 2**](#_Toc150896849)

[**МЕТОД ПАРЕТО 3**](#_Toc150896850)

[1.1 Выбор Парето-оптимального множества 3](#_Toc150896851)

[**1.2 Указание верхних/нижних границ критериев. 4**](#_Toc150896852)

[**МЕТОД ЭЛЕКТРА II 6**](#_Toc150896856)

[1.1 Выбор лучшего варианта 6](#_Toc150896857)

[1.2 Веса предпочтений 8](#_Toc150896858)

[1.3 Вывод 14](#_Toc150896859)

[**МЕТОД АНАЛИЗА ИЕРАРХИЙ 15**](#_Toc150896860)

[1.1 Постановка задачи 15](#_Toc150896861)

[1.2 Построение шкалы относительной важности 15](#_Toc150896862)

[1.3 Синтез приоритетов 17](#_Toc150896863)

[1.4 Согласованность локальных приоритетов 25](#_Toc150896864)

[1.5 Синтез альтернатив 32](#_Toc150896865)

[**Графический метод** Error! Bookmark not defined.](#_Toc150896866)

[1.1 Постановка задачи 34](#_Toc150896867)

[1.2 Данные индивидуального варианта 34](#_Toc150896868)

[1.3 Подготовка данных 34](#_Toc150896869)

[1.4 Построение графика 35](#_Toc150896870)

[1.5 Выделение области допустимых решений 35](#_Toc150896871)

[1.6 Максимум функции 37](#_Toc150896872)

[1.7 Минимум функции 39](#_Toc150896873)

[**СИМПЛЕКСНЫЙ МЕТОД 41**](#_Toc150896874)

[1.1 Постановка задачи 41](#_Toc150896875)

[1.2 Математическая модель задачи 41](#_Toc150896876)

[**ЗАКЛЮЧЕНИЕ 49**](#_Toc150896877)

[**СПИСОК ИНФОРМАЦИОННЫХ ИСТОЧНИКОВ 50**](#_Toc150896878)

ВВЕДЕНИЕ

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.

Для выбора одной оптимальной стратегии из множества эффективных решений в каждой конкретной многокритериальной задаче необходимо использовать дополнительную информацию.

МЕТОД ПАРЕТО

Состояние А (множество параметров) называется Парето-оптимальным, если не существует другого состояния В (множества других параметров), доминирующего состояние А относительно целевой функции. Состояние А доминирует состояние В, если хотя бы по одному параметру А лучше В, а по остальным не хуже.

Применительно к задаче переговоров этот принцип утверждает, что, если для ситуации В существует такая ситуация А, что выигрыш каждого из участников переговоров при реализации ситуации А не меньше, чем при реализации ситуации В и, по крайней мере, один переговорщик получит выигрыш строго больший, то они предпочтут ситуацию А ситуации В.

Если относительно пары альтернатив-решений одной и той же многокритериальной задачи нельзя сказать, какая из них лучше, то их называют несравнимыми. Множество таких альтернатив называются множеством Парето.

## **Выбор Парето-оптимального множества**

Стремления указаны в таблице 1.

Таблица 1. - Список альтернатив, критериев и стремлений.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| № | Варианты  Решений | Критерии | | | | |
| Цена  (-) | Рейтинг (+) | Мин. Баллы  (-) | Время поездки  (-) | Кол-во бюджетных мест  (+) |
| 1 | РАНХиГС | 250000 | 4.4 | 144 | 35 | 50 |
| 2 | РТУ МИРЭА | 200000 | 4.9 | 100 | 20 | 100 |
| 3 | МГТУ | 270000 | 4.6 | 115 | 120 | 70 |
| 4 | МГУ | 226000 | 4.9 | 100 | 30 | 120 |
| 5 | МТУСИ | 230000 | 4.5 | 120 | 100 | 40 |
| 6 | МАИ | 210000 | 3.6 | 90 | 90 | 120 |
| 7 | МЭИ | 220000 | 4.1 | 130 | 95 | 90 |
| 8 | Политех | 180000 | 4.0 | 100 | 150 | 100 |
| 9 | Синергия | 200000 | 3.5 | 90 | 80 | 130 |

Таблица 2. Таблица сравнений

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 1 | x | x | x | x | x | x | x | x | x |
| 2 | A2 | x | x | x | x | x | x | x | x |
| 3 | н | н | x | x | x | x | x | x | x |
| 4 | A4 | н | A4 | x | x | x | x | x | x |
| 5 | н | н | н | н | x | x | x | x | x |
| 6 | н | н | н | н | н | x | x | x | x |
| 7 | н | н | н | н | н | н | x | x | x |
| 8 | н | н | н | н | н | н | н | x | x |
| 9 | н | н | н | н | н | н | н | н | x |

# Указание верхних/нижних границ критериев.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| № | Варианты  Решений | Критерии | | | | |
| Цена  (-) | Рейтинг (+) | Мин. Баллы  (-) | Время поездки (мин)  (-) | Кол-во бюджетных мест  (+) |
| 1 | РАНХиГС | 250000 | 4.4 | 144 | 35 | 50 |
| 2 | РТУ МИРЭА | 200000 | 4.9 | 100 | 20 | 100 |
| 4 | МГУ | 226000 | 4.9 | 100 | 30 | 120 |

Установим границы для критерий "Рейтинг" и "Время поездки". Рейтинг не должен быть ниже 3.9, а время поездки не более 60. Согласно данному условию таблица 1 трансформируется в таблцу 3.

Таблица 3. Список альтернатив с учетом границ

Варианты, удовлетворяющие ограничениям: {1, 2, 4}. Оптимальными из них являются: {2, 4}.

**1.3 Субоптимизация**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| № | Варианты  Решений | Критерии | | | | |
| Цена  (-) | Рейтинг (+) | Мин. Баллы  (-) | Время поездки (мин)  (-) | Кол-во бюджетных мест  (+) |
| 2 | РТУ МИРЭА | 200000 | 4.9 | 100 | 20 | 100 |
| 4 | МГУ | 226000 | 4.9 | 100 | 30 | 120 |

Субоптимизацию производят следующим образом: выделяют один из критериев, а по всем остальным критериям назначают нижние границы. Оптимальным при этом считается исход, максимизирующий выделенный критерий на множестве исходов, оценки которых по остальным критериям не ниже назначенных. Пусть в примере главным критерием выступает цена; ограничения: рейтиг не меньше 4.6, время поездки не больше 30 минут. Отбросим варианты, которые не удовлетворяют данным ограничениям и составим таблицу 4.

Остаются варианты 2 и 4. Цена ниже во втором варианте. Этот вариант и будет оптимальным.

**1.4 Лексикографическая оптимизация**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| № | Варианты  Решений | Критерии | | | | |
| Цена  (-) | Рейтинг (+) | Мин. Баллы  (-) | Время поездки (мин)  (-) | Кол-во бюджетных мест  (+) |
| 2 | РТУ МИРЭА | 200000 | 4.9 | 100 | 20 | 100 |

Лексикографическая оптимизация основана на упорядочении критериев по их относительной важности. После этого процедуру нахождения оптимального решения проводят следующим образом. На первом шаге отбирают исходы, которые имеют максимальную оценку по важнейшему критерию. Если такой исход единственный, то его и считают оптимальным. Если же таких исходов несколько, то среди них отбирают те, которые имеют максимальную оценку по следующему за важнейшим критерию. В результате такой процедуры всегда остается (по крайней мере, в случае конечного множества исходов) единственный исход — он и будет оптимальным.

Упорядочим критерии в примере по относительной важности, например, следующим образом: важнейший критерий – цена, следующий за ним по важности – время поездки. Отбросим варианты, которые не удовлетворяют данным ограничениям и составим таблицу 5.

# МЕТОД ЭЛЕКТРА II

## **1.1 Выбор лучшего варианта**

Составлена таблица критериев, по которым оцениваются проекты (Таблица 1).

*Таблица 1 – Таблица критериев для оценки альтернатив*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Критерии | Вес критерия | Шкала | Код | Стремление |
| Цена | 3 | До 205 тыс.руб.  От 205 до 230 тыс. руб.  От 230 тыс.руб. | 5  10  20 | min |
| Рейтинг | 4 | До 4.3  От 4.3 до 4.6  От 4.6 | 5  10  15 | max |
| Мин. баллы | 2 | До 100  От 100 до 120  От 120 | 5  10  15 | min |
| Время поездки | 5 | До 30 минут  От 30 до 60 минут  От 60 минут | 5  15  25 | min |

Составлена таблица оценок выбора лучшего ВУЗа. Для 6-ти альтернатив заполнена Таблицу 2.

*Таблица 2 – Таблица оценок по критериям*

|  |  |
| --- | --- |
| № | Варианты решений |
| Критерии | | | |
| Цена | Рейтинг | Мин. Баллы | Время поездки |
| 1 | РАНХиГС | 250000 | 4.3 | 144 | 35 |
| 2 | РТУ МИРЭА | 200000 | 4.9 | 100 | 20 |
| 3 | МГУ | 226000 | 4.6 | 100 | 30 |
| 4 | МТУСИ | 230000 | 4.4 | 120 | 100 |
| 5 | МЭИ | 220000 | 4.1 | 130 | 95 |
| 6 | Политех | 180000 | 4.0 | 100 | 150 |
| Вес | | 3 | 4 | 2 | 5 |
| Стремление | | min | max | min | min |

**1.2 Веса предпочтений**

P12 = 0.0 + 0.0 + 0.0 + 0.0 = 0.0

N12 = 3.0 + 4.0 + 2.0 + 5.0 = 14.0

D12 = P12 / N12 = 0.0 - отбрасываем

D21 = N21 / P21 = inf - не подходит

P13 = 0.0 + 0.0 + 0.0 + 0.0 = 0.0

N13 = 3.0 + 4.0 + 2.0 + 5.0 = 14.0

D13 = P13 / N13 = 0.0 - отбрасываем

D31 = N31 / P31 = inf - не подходит

P14 = 0.0 + 0.0 + 0.0 + 5.0 = 5.0

N14 = 3.0 + 4.0 + 2.0 + 0.0 = 9.0

D14 = P14 / N14 = 0.5555555555555556 - отбрасываем

D41 = N41 / P41 = 1.8 - принимаем

P15 = 0.0 + 4.0 + 0.0 + 5.0 = 9.0

N15 = 3.0 + 0.0 + 2.0 + 0.0 = 5.0

D15 = P15 / N15 = 1.8 - принимаем

D51 = N51 / P51 = 0.5555555555555556 - отбрасываем

P16 = 0.0 + 4.0 + 0.0 + 5.0 = 9.0

N16 = 3.0 + 0.0 + 2.0 + 0.0 = 5.0

D16 = P16 / N16 = 1.8 - принимаем

D61 = N61 / P61 = 0.5555555555555556 - отбрасываем

P23 = 3.0 + 0.0 + 0.0 + 5.0 = 8.0

N23 = 0.0 + 0.0 + 0.0 + 0.0 = 0.0

D23 = P23 / N23 = inf - не подходит

D32 = N32 / P32 = 0.0 - отбрасываем

P24 = 3.0 + 4.0 + 2.0 + 5.0 = 14.0

N24 = 0.0 + 0.0 + 0.0 + 0.0 = 0.0

D24 = P24 / N24 = inf - не подходит

D42 = N42 / P42 = 0.0 - отбрасываем

P25 = 3.0 + 4.0 + 2.0 + 5.0 = 14.0

N25 = 0.0 + 0.0 + 0.0 + 0.0 = 0.0

D25 = P25 / N25 = inf - не подходит

D52 = N52 / P52 = 0.0 - отбрасываем

P26 = 0.0 + 4.0 + 0.0 + 5.0 = 9.0

N26 = 3.0 + 0.0 + 0.0 + 0.0 = 3.0

D26 = P26 / N26 = 3.0 - принимаем

D62 = N62 / P62 = 0.3333333333333333 - отбрасываем

P34 = 3.0 + 4.0 + 2.0 + 5.0 = 14.0

N34 = 0.0 + 0.0 + 0.0 + 0.0 = 0.0

D34 = P34 / N34 = inf - не подходит

D43 = N43 / P43 = 0.0 - отбрасываем

P35 = 0.0 + 4.0 + 2.0 + 5.0 = 11.0

N35 = 3.0 + 0.0 + 0.0 + 0.0 = 3.0

D35 = P35 / N35 = 3.6666666666666665 - принимаем

D53 = N53 / P53 = 0.2727272727272727 - отбрасываем

P36 = 0.0 + 4.0 + 0.0 + 5.0 = 9.0

N36 = 3.0 + 0.0 + 0.0 + 0.0 = 3.0

D36 = P36 / N36 = 3.0 - принимаем

D63 = N63 / P63 = 0.3333333333333333 - отбрасываем

P45 = 0.0 + 4.0 + 2.0 + 0.0 = 6.0

N45 = 3.0 + 0.0 + 0.0 + 5.0 = 8.0

D45 = P45 / N45 = 0.75 - отбрасываем

D54 = N54 / P54 = 1.3333333333333333 - принимаем

P46 = 0.0 + 4.0 + 0.0 + 5.0 = 9.0

N46 = 3.0 + 0.0 + 2.0 + 0.0 = 5.0

D46 = P46 / N46 = 1.8 - принимаем

D64 = N64 / P64 = 0.5555555555555556 - отбрасываем

P56 = 0.0 + 4.0 + 0.0 + 5.0 = 9.0

N56 = 3.0 + 0.0 + 2.0 + 0.0 = 5.0

D56 = P56 / N56 = 1.8 - принимаем

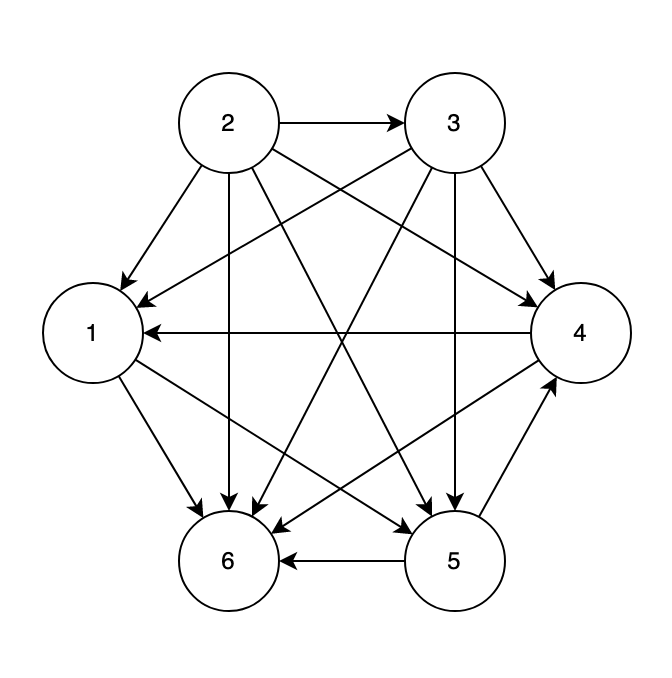
D65 = N65 / P65 = 0.5555555555555556 - отбрасываем

Составлена матрица предпочтений с внесенными и принятыми значениями D (Таблица 3).

*Таблица 3 – Полная матрица предпочтений альтернатив.*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 |
| 1 | х | - | - | - | 1.8 | 1.8 |
| 2 | inf | х | inf | inf | inf | 3.0 |
| 3 | inf | - | х | inf | 3.67 | 3.0 |
| 4 | 1.8 | - | - | х | - | 1.8 |
| 5 | - | - | - | 1.33 | х | 1.8 |
| 6 | - | - | - | - | - | x |

По матрице построен граф предпочтений (Рисунок 1).



**Рисунок 1 – Вид графа предпочтений**

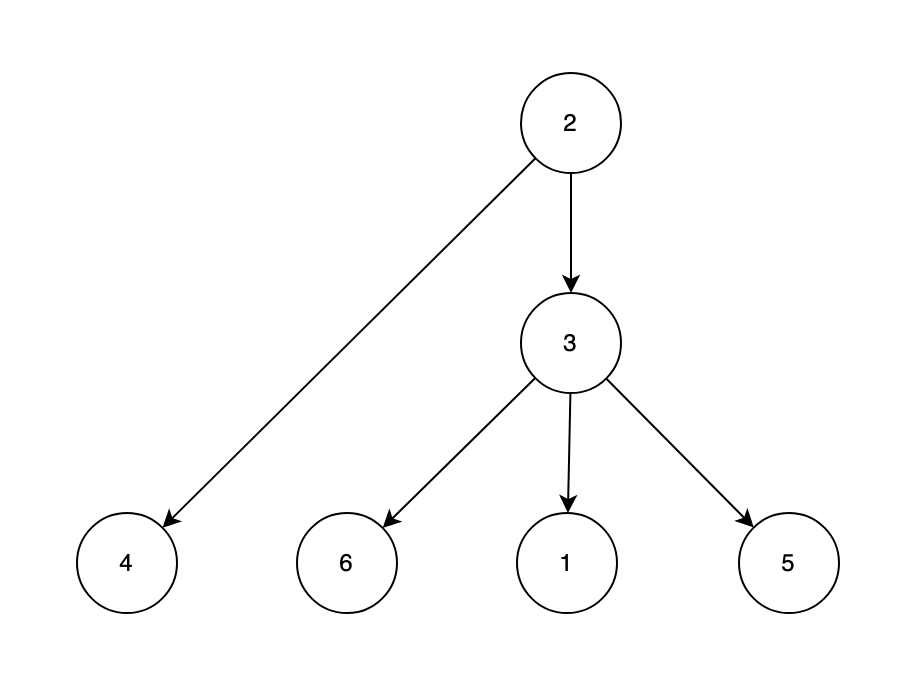
Назначен порог отбора предпочтений C = 2.0 (это соответствует тому, что учитываются только более сильные связи в графе).

Таким образом, матрица разрежается. В ней остаются только самые сильные связи (Таблица 4).

*Таблица 4* **–** *Матрица предпочтений проектов, при пороге С=2.0*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 |
| 1 | х | - | - | - | - | - |
| 2 | inf | х | inf | inf | inf | 3.0 |
| 3 | inf | - | х | - | 3.67 | 3.0 |
| 4 | - | - | - | х | - | - |
| 5 | - | - | - | - | х | - |
| 6 | - | - | - | - | - | x |

По этой матрице построен граф предпочтений (Рисунок 2).



**Рисунок 2 – Вид графа предпочтений для случая порога принятия решений C = 2.0**

## **1.3 Вывод**

Выводы работы по Электро II позволяют отметить, что метод предоставляет возможность принимать коллективные решения на основе мнения участников, учитывая их предпочтения и взаимоотношения. В результате были получены таблицы предпочтений и графики, которые позволили более полно представить предпочтения и представительность группы в выборе решений.  
 Однако, метод имеет и некоторые ограничения, связанные с вычислительной сложностью и временными затратами. Также возможен риск выбора неоптимального решения, если в группе доминирует определенная подгруппа или некоторые участники не могут выразить свое мнение достаточно четко.  
Таким образом, можно сделать вывод, что метод Электра II может быть эффективным инструментом для принятия коллективных решений в различных сферах деятельности, однако его применение требует тщательного анализа и оценки результатов.

**МЕТОД АНАЛИЗА ИЕРАРХИЙ**

**1.1 Постановка задачи**

Предметная область: Высшие учебные заведения.

Задача: выбрать лучшее высшее учебное заведение.

В данной практической работе исследуется и используется метод Анализа Иерархий, позволяющий делать оптимальный выбор, на примере выбранной предметной области.

**1.2 Построение шкалы относительной важности**

Построим полную доминантную иерархию (см рис. 1.1).

Изображение выглядит как текст

Автоматически созданное описание

Выбор лучшего ВУЗа

**Рисунок 1.4 – Полная доминантная иерархия**

Критерии:

К1 – Проходной балл

K2 – Кол-во бюджетных мест

K3 – Цена

K4 – Рейтинг

K5 – Время поездки

Альтернативы:

A1 – Синергия

A2 - РАНХиГС

A3 - МГТУ им. Баума

A4 - РТУ МИРЭА

A5 - МГУ

После иерархического представления задачи необходимо установить приоритеты критериев и оценить каждую из альтернатив по критериям, определив наиболее важную их них.

В методе анализа иерархий элементы сравниваются попарно по отношению к их влиянию на общую для них характеристику.

Для облегчения работы введем шкалу относительной важности (табл. 1.1).

*Таблица 1.1 – Шкала относительности 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 |

**1.3 Синтез приоритетов**

Из групп парных сравнений формируется набор локальных критериев, которые выражают относительное влияние элементов на элемент, расположенный на уровне выше.

Составим обратно симметричную матрицу для парного сравнения критериев (табл. 1.2).

*Таблица 1.2 – Таблица парного сравнения критериев*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Цель | K1 | K2 | K3 | K4 | K5 | Vi | W2i |
| K1 | 1 | 3 | 3 | 5 | 5 | 2.954 | 0.452 |
| K2 | 1/3 | 1 | 3 | 3 | 3 | 1.551 | 0.237 |
| K3 | 1/3 | 1/3 | 1 | 3 | 5 | 1.107 | 0.169 |
| K4 | 1/5 | 1/3 | 1/3 | 1 | 3 | 0.581 | 0.088 |
| K5 | 1/5 | 1/3 | 1/5 | 1/3 | 1 | 0.338 | 0.051 |
| ΣVi | | | | | | 6.531 |

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

Строка № 1

V1 = (1х3х3х5х 5)1/5 = 2.954

Строка № 2

V2 = (1/3х1х3х3х3)1/5 = 1.551

Строка № 3

V3 = (1/3х1/3х1х3х5)1/5 = 1,107

Строка № 4

V4 = (1/5х1/3х1/3х1х7)1/5 = 0.581

Строка № 5

V5 = (1/5х1/3х1/5х1/7х1)1/5 = 0.338

Находим сумму Vi и можем все вычисления переписывать в Таблицу 1.2.

∑Vi = V1+V2+V3+V4+V5 = 2,954+1,551+1,107+0,581+0,338 = 6.531

Далее каждое из чисел Vi делим на ∑Vi, в результате найдем важность приоритетов.

Строка № 1

W21= 2,954/6.531= 0.452

Строка № 2

W22= 1,551/6.531= 0.237

Строка № 3

W23= 1,107/6.531= 0.169

Строка № 4

W24= 0.581/6.531= 0.088

Строка № 5

W25= 0.338/6.531= 0.051

В результате получаем вектор приоритетов:

W2i = (0.452; 0.237; 0.169; 0.088; 0.051)

*Таблица 1.3 – Таблица критерия “Проходной балл”*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| K1 | A1 | A2 | A3 | A4 | A5 | VK1Y | W3K1Y |
| A1 | 1 | 1/3 | 1/3 | 1/9 | 1/9 | 0.267 | 0.031 |
| A2 | 3 | 1 | 1/3 | 1/9 | 1/9 | 0.415 | 0.048 |
| A3 | 3 | 3 | 1 | 1/7 | 1/7 | 0.712 | 0.083 |
| A4 | 9 | 9 | 7 | 1 | 1 | 3.553 | 0.418 |
| A5 | 9 | 9 | 7 | 1 | 1 | 3.553 | 0.418 |
| ΣVK1Y | | | | | | 8.5 |

Строка № 1

VK11 = (1x1/3x1/3x1/9x1/9)1/5 = 0.267

Строка № 2

VK12 = (3x1x1/3x1/9x1/9)1/5 = 0.415

Строка № 3

VK13 = (3x3x1x1/7x1/7)1/5 = 0.712

Строка № 4

VК14=(9х9х7х1х1)1/5= 3.553

Строка № 5

VК15=(9х9х7х1х1)1/5= 3.55

Находим сумму VК1Y и можем переписывать вычисления в Таблицу 1.3.

∑VК1Y= VК11+VК12+VК13+VК14+VК15 = 0.267+0.415+0.712+3,553+3,553 = 8,5

И каждое из чисел VК1Y делим на ∑ VК1Y, в результате найдем важность приоритетов.

Строка № 1

W3К11= 0.267/8,5= 0.031

Строка № 2

W3К12= 0.415/8,5= 0.048

Строка № 3

W3К13= 0.712/8,5= 0.083

Строка № 4

W3К14= 3,553/8,5= 0.418

Строка № 5

W3К15= 3,553 /8,5= 0.418

В результате получаем вектор приоритетов:

W3К1Y = (0.031; 0.048; 0.083; 0.418; 0.418)

*Таблица 1.4 - Таблица критерия “Кол-во бюджетных мест”*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| K2 | A1 | A2 | A3 | A4 | A5 | VK2Y | W3K2Y |
| A1 | 1 | 1/5 | 1/7 | 1/9 | 1/9 | 0.203 | 0,025 |
| A2 | 5 | 1 | 1/7 | 1/9 | 1/9 | 0.388 | 0,048 |
| A3 | 7 | 7 | 1 | 1/3 | 1/3 | 1.403 | 0,1755 |
| A4 | 9 | 9 | 3 | 1 | 1 | 3 | 0,375 |
| A5 | 9 | 9 | 3 | 1 | 1 | 3 | 0,375 |
| ΣVK2Y | | | | | | 7.995 |

Строка № 1

VK21 = (1х1/5х1/7х1/9х1/9)1/5= 0.203

Строка № 2

VK22 = (5х1х1/7х1/9х1/9)1/5= 0.388

Строка № 3

VK23 = (7х7х1х1/3х1/3)1/5= 1.403

Строка № 4

VК24=(9х9х3х1х1)1/5= 3

Строка № 5

VК25=(9х9х3х1х1)1/5= 3

Находим сумму VК2Y и можем переписывать вычисления в Таблицу 1.4.

∑VК2Y= 0.203+0.388+1.403+3+3 = 7,994.

И каждое из чисел VК2Y делим на ∑ VК2Y, в результате найдем важность приоритетов.

Строка № 1

W3К21= 0.203/7,995= 0,025

Строка № 2

W3К22= 0.388/7,995 = 0,048

Строка № 3

W3К23= 1.403/7,995= 0,1755

Строка № 4

W3К24= 3/7,995= 0,375

Строка № 5

W3К25= 3/7,995= 0,375

В результате получаем вектор приоритетов:

W3К2Y = (0,025; 0,048; 0,1755; 0,375; 0,375)

*Таблица 1.5 - Таблица критерия “Цена”*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| K2 | A1 | A2 | A3 | A4 | A5 | VK2Y | W3K2Y |
| A1 | 1 | 1/5 | 1/7 | 1/9 | 1/9 | 0.203 | 0,025 |
| A2 | 5 | 1 | 1/7 | 1/9 | 1/9 | 0.388 | 0,048 |
| A3 | 7 | 7 | 1 | 1/3 | 1/3 | 1.403 | 0,1755 |
| A4 | 9 | 9 | 3 | 1 | 1 | 3 | 0,375 |
| A5 | 9 | 9 | 3 | 1 | 1 | 3 | 0,375 |
| ΣVK2Y | | | | | | 7.995 |

Строка № 1

VK31 = (1х1/3х1/5х1/9х1/7)1/5= 0.254

Строка № 2

VK32 = (3х1х1/3х1/9х1/7)1/5= 0.4366

Строка № 3

VK33 = (5х3х1х1/7х1/3)1/5= 0.9349

Строка № 4

VК34=(9х9х7х1х2)1/5= 4.082

Строка № 5

VК35=(7х7х3х1/2х1)1/5= 2.3618

Находим сумму VК3Y и можем переписывать вычисления в Таблицу 1.5.

∑VК3Y= 0.254+0.4366+0.9349+4.082+2.3618= 8,0693

И каждое из чисел VК3Y делим на ∑ VК3Y, в результате найдем важность приоритетов.

Строка № 1

W3К31= 0.254/8,0693= 0,0314

Строка № 2

W3К32= 0.4366/8,0693= 0,0541

Строка № 3

W3К33= 0.9349/8,0693= 0,1158

Строка № 4

W3К34= 4.082/8,0693= 0,5058

Строка № 5

W3К35= 2.3618/8,0693= 0,2926

В результате получаем вектор приоритетов:

W3К3Y = (0,0314; 0,0541; 0,1158; 0,5058; 0,2926)

*Таблица 1.6 – Таблица критерия “Рейтинг”*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| K4 | A1 | A2 | A3 | A4 | A5 | VK4Y | W3K4Y |
| A1 | 1 | 1/3 | 1/5 | 1/9 | 1/9 | 0.2415 | 0.0293 |
| A2 | 3 | 1 | 1/7 | 1/9 | 1/7 | 0.3685 | 0.0448 |
| A3 | 5 | 7 | 1 | 1/5 | 1/3 | 1.1846 | 0.144 |
| A4 | 9 | 9 | 5 | 1 | 3 | 4.1391 | 0.5032 |
| A5 | 7 | 7 | 3 | 1/5 | 1 | 2.290 | 0.2784 |
| ΣVK4Y | | | | | | 8.224 |

Строка № 1

VK41 = (1х1/3х1/5х1/9х1/9)1/5= 0.2415

Строка № 2

VK42 = (3х1х1/7х1/9х1/7)1/5= 0.3685

Строка № 3

VK43 = (5х7х1х1/5х1/3)1/5= 1.1846

Строка № 4

VК44=(9х9х5х1х3)1/5= 4.1391

Строка № 5

VК45=(7х7х3х1/5х1)1/5= 2.290

Находим сумму VК4Y и можем переписывать вычисления в Таблицу 1.6.

∑VК4Y= 0.2415+0.3685+1.1846+4.1391+2.290 = 8.224

И каждое из чисел VК4Y делим на ∑ VК4Y, в результате найдем важность приоритетов.

Строка № 1

W3К41= 0.2415/8.224= 0.029

Строка № 2

W3К42= 0.3685/8.224= 0.0448

Строка № 3

W3К43= 1.1846/8.224= 0.144

Строка № 4

W3К44= 4.139/8.224= 0.5032

Строка № 5

W3К45= 2.290/8.224= 0.2784

В результате получаем вектор приоритетов:

W3К4Y = (0.029; 0.0448; 0.144; 0.5032; 0.2784)

*Таблица 1.7 – Таблица критерия “Время поездки”*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| K5 | A1 | A2 | A3 | A4 | A5 | VK5Y | W3K5Y |
| A1 | 1 | 3 | 5 | 7 | 9 | 3.9362 | 0.5021 |
| A2 | 1/3 | 1 | 3 | 5 | 9 | 2.1411 | 0.2731 |
| A3 | 1/5 | 1/3 | 1 | 3 | 5 | 1 | 0.1275 |
| A4 | 1/7 | 1/5 | 1/3 | 1 | 5 | 0.5439 | 0.0693 |
| A5 | 1/9 | 1/9 | 1/5 | 1/5 | 1 | 0.2181 | 0.0278 |
| ΣVK5Y | | | | | | 7.839 |

Строка № 1

VK51 = (1x3x5x7x9)1/5= 3.9362

Строка № 2

VK52 = (1/3х1х3х5х9)1/5= 2.1411

Строка № 3

VK53 = (1/5х1/3х1х3х5)1/5= 1

Строка № 4

VК54=(1/7х1/5х1/3х1х5)1/5= 0.5439

Строка № 5

VК55=(1/9х1/9х1/5х1/5х1)1/5= 0.2181

Находим сумму VК5Y и можем переписывать вычисления в Таблицу 1.7.

∑VК5Y= 3.9362+2.1411+1+0.5439+0.2181= 7.839.

И каждое из чисел VК4Y делим на ∑ VК5Y, в результате найдем важность приоритетов.

Строка № 1

W3К51= 3.9362/7.839= 0.5021

Строка № 2

W3К52= 2.1411/7.839= 0.2731

Строка № 3

W3К53= 1/7.839= 0.1275

Строка № 4

W3К54= 0.5439/7.839= 0.0693

Строка № 5

W3К55= 0.2181/7.839= 0.0278

В результате получаем вектор приоритетов:

W3К5Y = (0.5021; 0.2731; 0.1275; 0.0693; 0.0278)

**1.4 Согласованность локальных приоритетов**

В таблице приведены средние значения индекса случайной согласованности (СИ) для случайных матриц суждений разного порядка. Ниже приведена Таблица 1.8.

*Таблица 1.8 – таблица СИ*

|  |  |
| --- | --- |
| Размер матрицы | Среднее значение  индекса случайной  согласованности  (СИ) |
| 1 | 0.00 |
| 2 | 0.00 |
| 3 | 0.58 |
| 4 | 0.90 |
| 5 | 1.12 |
| 6 | 1.24 |
| 7 | 1.32 |
| 8 | 1.41 |
| 9 | 1.45 |
| 10 | 1.49 |
| 11 | 1.51 |
| 12 | 1.48 |
| 13 | 1.56 |
| 14 | 1.57 |

В нашей задаче размерность матрицы n=5, тогда среднее значение индекса случайной согласованности СИ = 1,12.

Определим индекс согласованности и отношение согласованности для матрицы «Цель» (табл. 1.2).

Определяется сумма каждого столбца матрицы суждений.

S1 = 1 + 1/3 + 1/3 + 1/5 + 1/5 = 31/15

S2 = 3 + 1 + 1/3 + 1/3 + 1/3 = 5

S3 = 3 + 3 + 1 + 1/3 + 1/5 = 113/15

S4 = 5 + 3 + 3 + 1 + 1/3 = 37/3

S5 = 5 + 3 + 5 + 3 + 1 = 17

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

P1 = S1 x W21 = 31/15 х 0,452 = 0,934

P2 = S2 x W22 = 5 х 0,237 = 1,185

P3 = S3 x W23 = 113/15 х 0,169 = 1,2769

P4 = S4 x W24 = 37/3 х 0.089= 1,098

P5 = S5  x W25 = 17 х 0.0518 = 0,8807

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

λmax = P1 + P2 + P3 + P4 + P5 = 0,934 + 1,185+ 1,2769+ 1,098 + 0,8807= 5,377

Отклонение от согласованности выражается индексом согласованности.

ИС = (λmax – n)/(n-1) = (5,377 – 5)/(5-1) = 0.09445

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

ОС = ИС/СИ = 0.09445/1.12 = 0.0843

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

Определим индекс согласованности и отношение согласованности для матрицы К1 «Проходной балл» (табл. 1.3).

Определяется сумма каждого столбца матрицы суждений.

S1K1 = 1 + 3 + 3 + 9 + 9 = 25

S2K1 = 1/3 + 1 + 3 + 9 + 9 = 22,33

S3K1 = 1/3 + 1/3 + 1 + 7 + 7 = 15.66

S4K1 = 1/9 + 1/9 + 1/7 + 1 + 1 = 2,365

S5K1 = 1/9 + 1/9 + 1/7 + 1 + 1 = 2,365

Затем полученный результат умножается на компоненту нормализованного вектора приоритетов.

P­1K1 = S1 x W3K11 = 25 х 0,031 = 0,786

P­2K1 = S2 x W3K12 = 22,33х 0,048 = 1,09

P­3K1 = S3 x W3K13 = 15.66 х 0,083 = 1,312

P­4K1 = S4 x W3K14 = 2,365х 0,418 = 0,988

P­5K1 = S5 x W3K15 = 2,365х 0,418 = 0,988

Находим пропорциональность предпочтений.

λmax K1 = 0,786 + 1,09 + 1,312 + 0,988 + 0,988 = 5.167

Отклонение от согласованности выражается индексом согласованности.

ИСK1 = (λmax K1 – n)/(n-1) = (5.167– 5)/(5-1) = 0,0417

Найдем отношением согласованности ОС.

ОС­K1 = ИС/СИ = 0,0417/1,12 = 0,0372

Значение ОС меньше или равное 0.10 считается приемлемым, значит матрица К1 «Проходной балл» согласована.

Определим индекс согласованности и отношение согласованности для матрицы К2 «Кол-во бюджетных мест» (табл. 1.4).

Определяется сумма каждого столбца матрицы суждений.

S1K2 = 1 + 5 + 7 + 9 + 9 = 31

S2K2 = 1/5 + 1 + 7 + 9 + 9 = 26,2

S3K2 = 1/7 + 1/7 + 1 + 3 + 3 = 7.285

S4K2 = 1/9 + 1/9 + 1/3 + 1 + 1 = 2.55

S5K2 = 1/9 + 1/9 + 1/3 + 1 + 1 = 2,55

Затем полученный результат умножается на компоненту нормализованного вектора приоритетов.

P­1K2 = S1 x W3K21 =31 х 0,025 = 0,7906

P­2K2 = S2 x W3K22 = 26,2х 0,048 = 1,272

P­3K2 = S3 x W3K23 = 7.285х 0,1755= 1,2788

P­4K2 = S4 x W3K24 = 2.55х 0,375= 0,9588

P­5K2 = S5 x W3K25 = 2.55х 0,375= 0,9588

Находим пропорциональность предпочтений.

λmax K2 = 0,7906+ 1,272+ 1,2788+ 0,9588 + 0,9588 = 5,259.

Отклонение от согласованности выражается индексом согласованности.

ИСK2 = (λmax K2 – n)/(n-1) = (5,259 – 5)/(5-1) = 0,0648

Найдем отношением согласованности ОС.

ОС­K2 = ИС/СИ = 0,0648/1,12 = 0,05789

Значение ОС меньше или равное 0.10 считается приемлемым, значит матрица К2 «Кол-во бюджетных мест» согласована.

Определим индекс согласованности и отношение согласованности для матрицы К3 «Цена» (табл. 1.5).

Определяется сумма каждого столбца матрицы суждений.

S1K3 = 1 + 3 + 5 + 9 + 7 = 25

S2K3 = 1/3 + 1 + 3 + 9 + 7 = 20.33

S3K3 = 1/5 + 1/3 + 1 + 7 + 3 = 11.533

S4K3 = 1/9 + 1/9 + 1/7 + 1 + 1/2 = 1.865

S5K3 = 1/7 + 1/7 + 1/3 + 2 + 1 = 3.619

Затем полученный результат умножается на компоненту нормализованного вектора приоритетов.

P­1K3 = S1 x W3K31 =25х 0,0314 = 0,787

P­2K3 = S2 x W3K32 = 20.33 х 0,0541= 1.1

P­3K3 = S3 x W3K33 = 11.533 х 0,1158 = 1,336

P­4K3 = S4 x W3K34 = 1.865 х 0,5058= 0.943

P­5K3 = S5 x W3K35 = 3.619 х 0,2926= 1,059

Находим пропорциональность предпочтений.

λmax K3 = 0,787 + 1.1 + 1,336 + 0.943 + 1,059 = 5,226

Отклонение от согласованности выражается индексом согласованности.

ИСK3 = (λmax K3 – n)/(n-1) = (5,226 – 5)/(5-1) = 0,056

Найдем отношением согласованности ОС.

ОС­K3 = ИС/СИ = 0,056/1,12 = 0,0504

Значение ОС меньше или равное 0.10 считается приемлемым, значит матрица К3 «Цена» согласована.

Определим индекс согласованности и отношение согласованности для матрицы К4 «Рейтинг» (табл. 1.6).

Определяется сумма каждого столбца матрицы суждений.

S1K4 = 1 + 3 + 5 + 9 + 7 = 27

S2K4 = 1/3 + 1 + 7 + 9 + 7 = 24,33

S3K4 = 1/5 + 1/7 + 1 + 5 + 3 = 9.342

S4K4 = 1/9 + 1/9 + 1/5 + 1 + 1/5 = 1,755

S5K4 = 1/9 + 1/7 + 1/3 + 3 + 1 = 4.587

Затем полученный результат умножается на компоненту нормализованного вектора приоритетов.

P­1K4 = S1 x W3K41 =27 х 0.0293= 0,7931

P­2K4 = S2 x W3K42 = 24,33х 0.0448= 1.0905

P­3K4 = S3 x W3K43 = 9.342х 0.144= 1,3458

P­4K4 = S4 x W3K44 = 1,755х 0.5032= 0,8835

P­5K4 = S5 x W3K45 = 4.587х 0.2784= 1,277

Находим пропорциональность предпочтений.

λmax K4 = 0,7931+ 1.0905 + 1,3458 + 0,8835 + 1,277= 5,3904.

Отклонение от согласованности выражается индексом согласованности.

ИСK4 = (λmax K4 – n)/(n-1) = (5,3904 – 5)/(5-1) = 0,0976

Найдем отношением согласованности ОС.

ОС­K4 = ИС/СИ = 0,0976/1,12 = 0,0871

Значение ОС меньше или равное 0.10 считается приемлемым, значит матрица К4 «Рейтинг» согласована.

Определим индекс согласованности и отношение согласованности для матрицы К5 «Время поездки».

Определяется сумма каждого столбца матрицы суждений.

S1K5 = 1 + 1/3 + 1/5 + 1/7+ 1/9 = 1,7873

S2K5 = 3 + 1 + 1/3 + 1/5 + 1/9 = 1.268

S3K5 = 5 + 3 + 1 + 1/3 + 1/5 = 9.533

S4K5 = 7 + 5 + 3 + 1 + 1/5 = 16.2

S5K5 = 9 + 9 + 5 + 5 + 1 = 29

Затем полученный результат умножается на компоненту нормализованного вектора приоритетов.

P­1K5 = S1 x W3K51 =1,7873х 0.5021= 0,8974

P­2K5 = S2 x W3K52 = 1.268 х 0.2731= 0.3462

P­3K5 = S3 x W3K53 = 9.533х 0.1275= 1,216

P­4K5 = S4 x W3K54 = 16.2 х 0.0693= 1.126

P­5K5 = S5 x W3K55 = 29 х 0,0278 = 0,806

Находим пропорциональность предпочтений.

λmax K5 = P1K5 + P2K5 + P3K5 + P4K5 + P5K5 = 0,8974 + 0.3462 + 1,216 + 1.126 + 0,806 = 5,3129

Отклонение от согласованности выражается индексом согласованности.

ИСK5 = (λmax K5 – n)/(n-1) = (5,3129 – 5)/(5-1) = 0,078

Найдем отношением согласованности ОС.

ОС­K5 = ИС/СИ = 0,078/1,12 = 0,069

Значение ОС меньше или равное 0.10 считается приемлемым, значит матрица К5 «Время поездки» согласована.

**1.5 Синтез альтернатив**

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

W2i = (0.452; 0.237; 0.169; 0.088; 0.051)

W3К1Y = (0.031; 0.048; 0.083; 0.418; 0.418)

W3К2Y = (0,025; 0,048; 0,1755; 0,375; 0,375)

W3К3Y = (0,0314; 0,0541; 0,1158; 0,5058; 0,2926)

W3К4Y = (0.029; 0.0448; 0.144; 0.5032; 0.2784)

W3К5Y = (0.5021; 0.2731; 0.1275; 0.0693; 0.0278)

Приоритеты альтернатив получим следующим образом:

W1 = W21 х W3К11 + W22 х W3К21 + W23 х W3К31 + W24 х W3К41 + W25 х W3К51 = 0,452 х 0,031 + 0,237 х 0,025 + 0,169 х 0,0314 + 0,088 х 0,029 + 0,051 х 0,5021 = 0,05425

W2 = W21 х W3К12 + W22 х W3К22 + W23 х W3К32 + W24 х W3К42 + W25 х W3К52 = 0,452 х 0.048 + 0,237 х 0.048 + 0,169 х 0,0541 + 0,088 х 0.0448+ 0,051 х 0.2731 = 0.0609

W3 = W21 х W3К13 + W22 х W3К23 + W23 х W3К33 + W24 х W3К43 + W25 х W3К53 = 0,452 х 0.083 + 0,237 х 0,1755+ 0,169 х 0,1158 + 0,088 х 0.144 + 0,051 х 0.1275 = 0.1186

W4 = W21 х W3К14 + W22 х W3К24 + W23 х W3К34 + W24 х W3К44 + W25 х W3К54 = 0,452 х 0.418 + 0,237 х 0,375 + 0,169 х 0,5058 + 0,088 х 0.5032 + 0,051 х 0.0693 = 0.4122

W5 = W21 х W3К15 + W22 х W3К25 + W23 х W3К35 + W24 х W3К45 + W25 х W3К55 = 0,452 х 0.418 + 0,237 х 0,375 + 0,169 х 0,2926 + 0,088 х 0.2784 + 0,051 х 0.0278 = 0.3539

Таким образом, приоритеты альтернатив равны:

Альтернатива А1 (Проходной балл) - W1 приоритет равен 0,05425;

Альтернатива А2 (Кол-во бюджетных мест) – W2 приоритет равен 0.0609;

Альтернатива А3 (Мощности двигателя) – W3 приоритет равен 0.1186;

Альтернатива А4 (Рейтинг) – W4 приоритет равен 0.4122;

Альтернатива А5 (Время поездки) – W5 приоритет равен 0.3539;

Наиболее перспективным с позиции метода анализа иерархий признается выбор ВУЗа А4 - РТУ МИРЭА. Однако видно, что выбор А5 - МГУ, оказывается тоже неплохим.

# ГРАФИЧЕСКИЙ МЕТОД

## 1.1 Постановка задачи

Решить задачу линейного программирования с двумя переменными графическим методом.

## 1.2 Данные индивидуального варианта

## 1.3 Подготовка данных

В среде Microsoft Excel добавим 4 столбца:

1. – значения от 0 до 10 с шагом 0,5;
2. – значения ограничения ;
3. – значения ограничения ;
4. – значения = 0.

*Таблица 1.1 – Данные для графика*

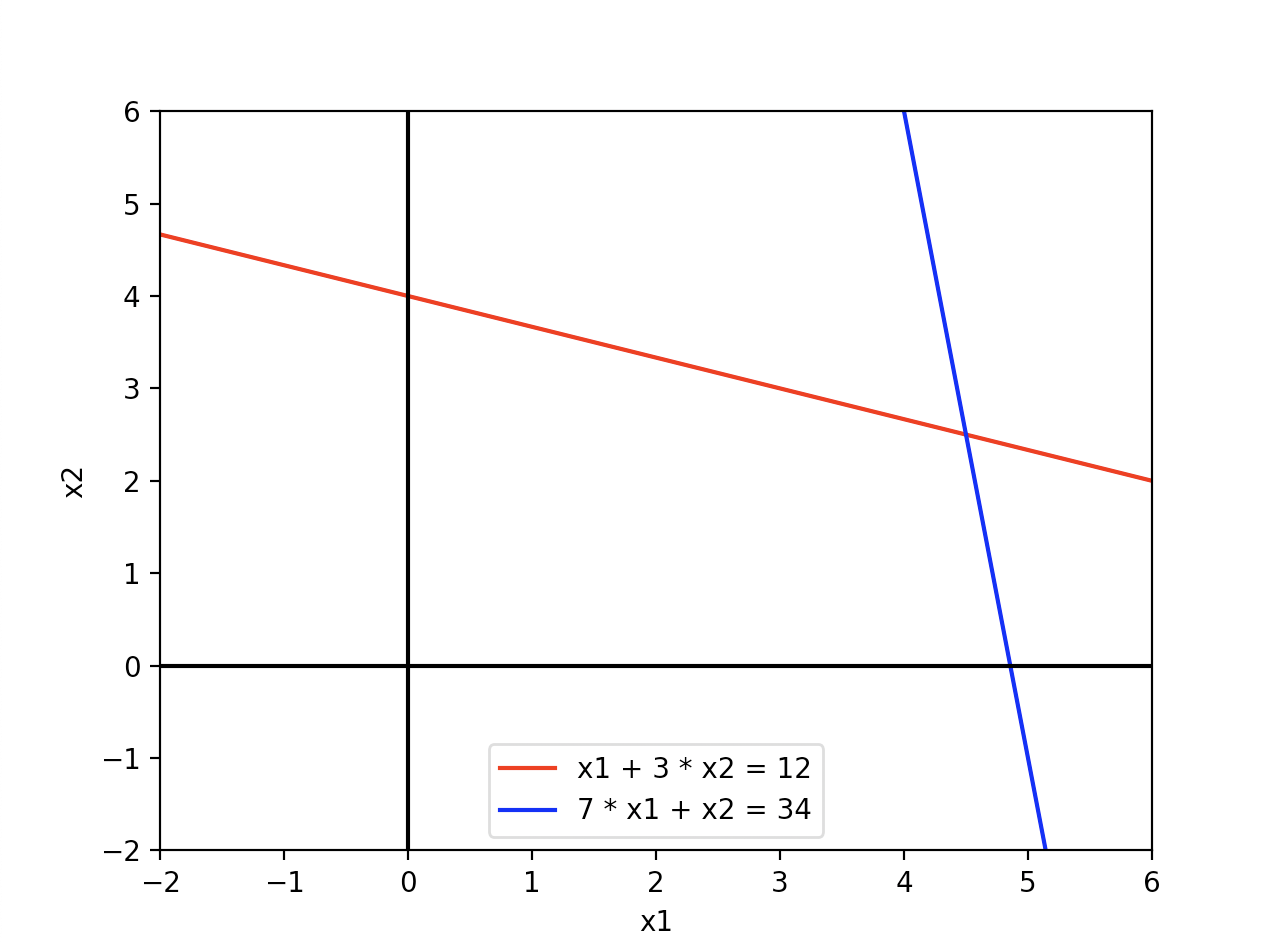
|  |  |  |  |
| --- | --- | --- | --- |
| x1 |  |  | f(x) = |
| 0 | 4 | 34 | 0 |
| 0,5 | 3,8 | 30,5 | -0,5 |
| 1 | 3,6 | 27 | -1 |
| 1,5 | 3,5 | 23,5 | -1,5 |
| 2 | 3,3 | 20 | -2 |
| 2,5 | 3,16 | 16,5 | -2,5 |
| 3 | 3 | 13 | -3 |
| 3,5 | 2,8 | 9,5 | -3,5 |
| 4 | 2,6 | 6 | -4 |
| 4,5 | 2,5 | 2,5 | -4,5 |
| 5 | 2,3 | -1 | -5 |
| 5,5 | 2,1 | -4,5 | -5,5 |

*Продолжение Таблицы 1.1*

|  |  |  |  |
| --- | --- | --- | --- |
| 6 | 2 | -8 | -6 |
| 6,5 | 1,8 | -11,5 | -6,5 |
| 7 | 1,6 | -15 | -7 |
| 7,5 | 1,5 | -18,5 | -7,5 |
| 8 | 1,3 | -22 | -8 |
| 8,5 | 1,1 | -25,5 | -8,5 |
| 9 | 1 | -29 | -9 |
| 9,5 | 0,8 | -32,5 | -9,5 |
| 10 | 0,6 | -36 | -10 |

## 1.4 Построение графика

Выделим таблицу подготовленных данных и построим гладкий график. Произведём настройку шага координатной оси x1 и получим следующие графики (Рисунок 1.1).

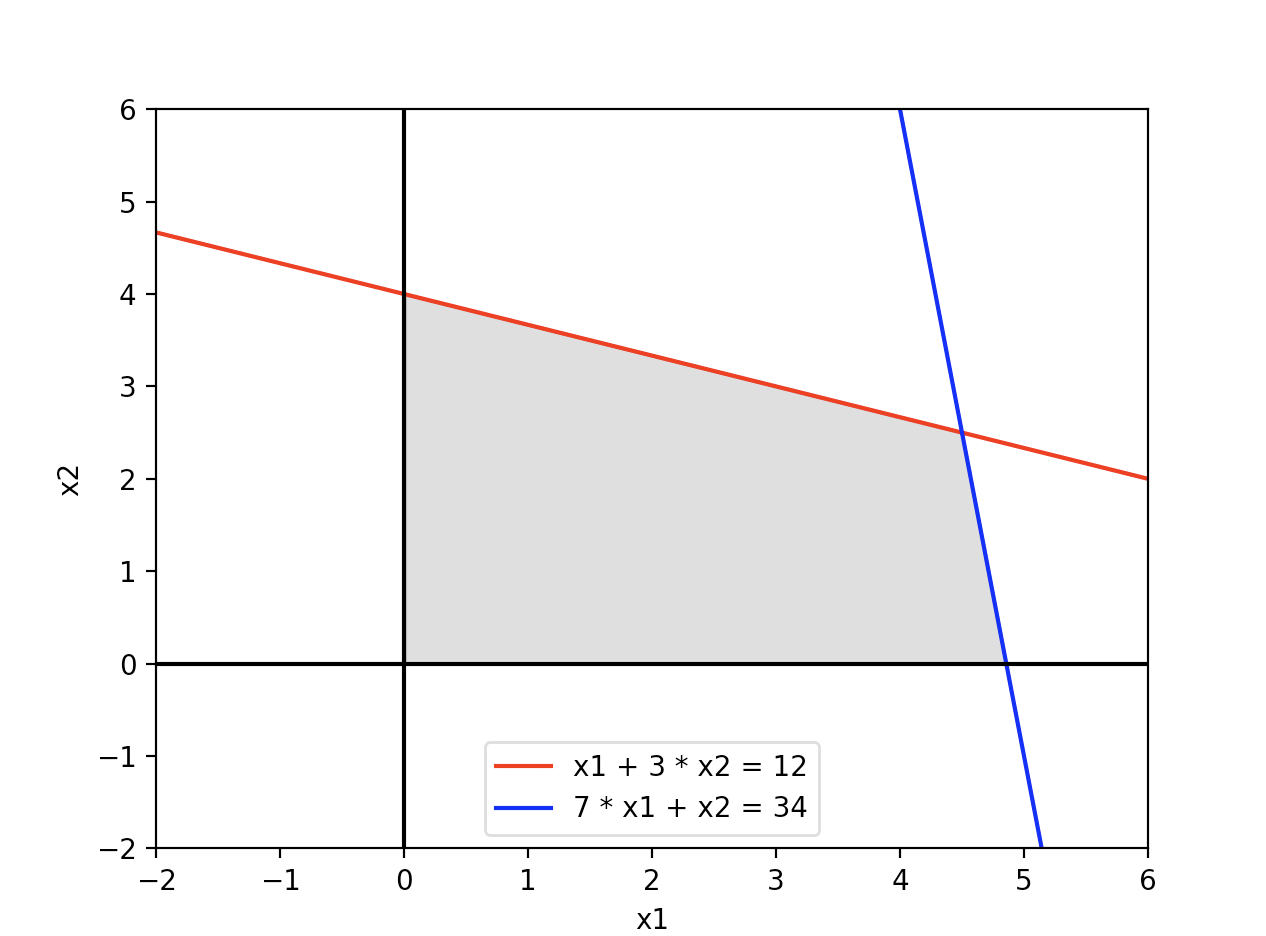


**Рисунок 1.1 – Построение графиков по данным**

## 1.5 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 1.2



**Рисунок 1.2 – Выделение области допустимых значений**

## 1.6 Максимум функции

Для нахождения максимума функции найдем её градиент по формуле 1.1:

(1.1)

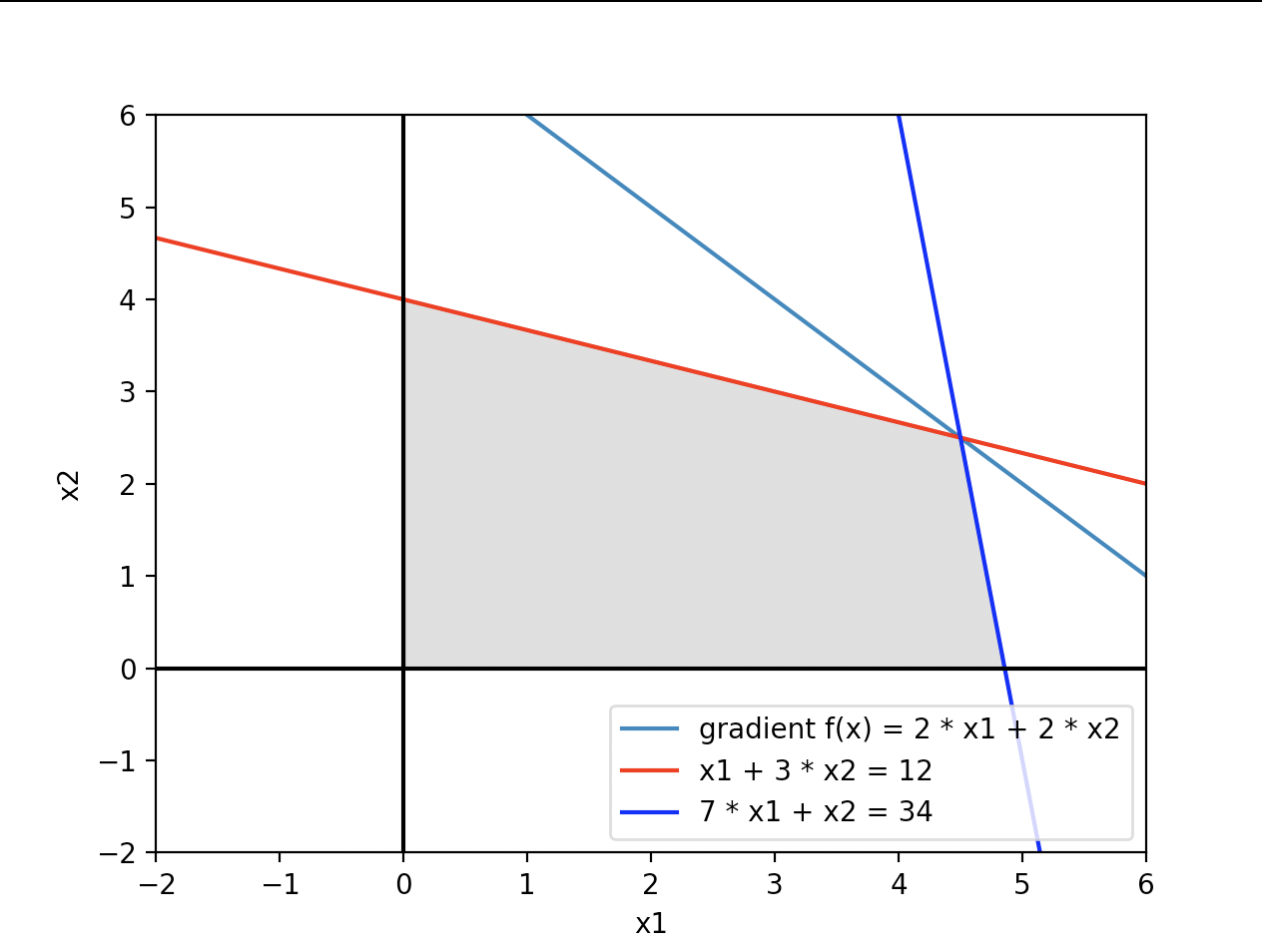
Для нахождения минимума функции найдем её градиент по формуле 1.1:

(1.2)

Градиент функции будет равен {2, 2}, а антиградиент функции будет равен {-2, -2}. Изобразим эти вектора на графике (Рисунок 1.4).

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

Координаты точки максимума: {4,5; 2,5}



**Рисунок 1.4 – Точка максимума функции**

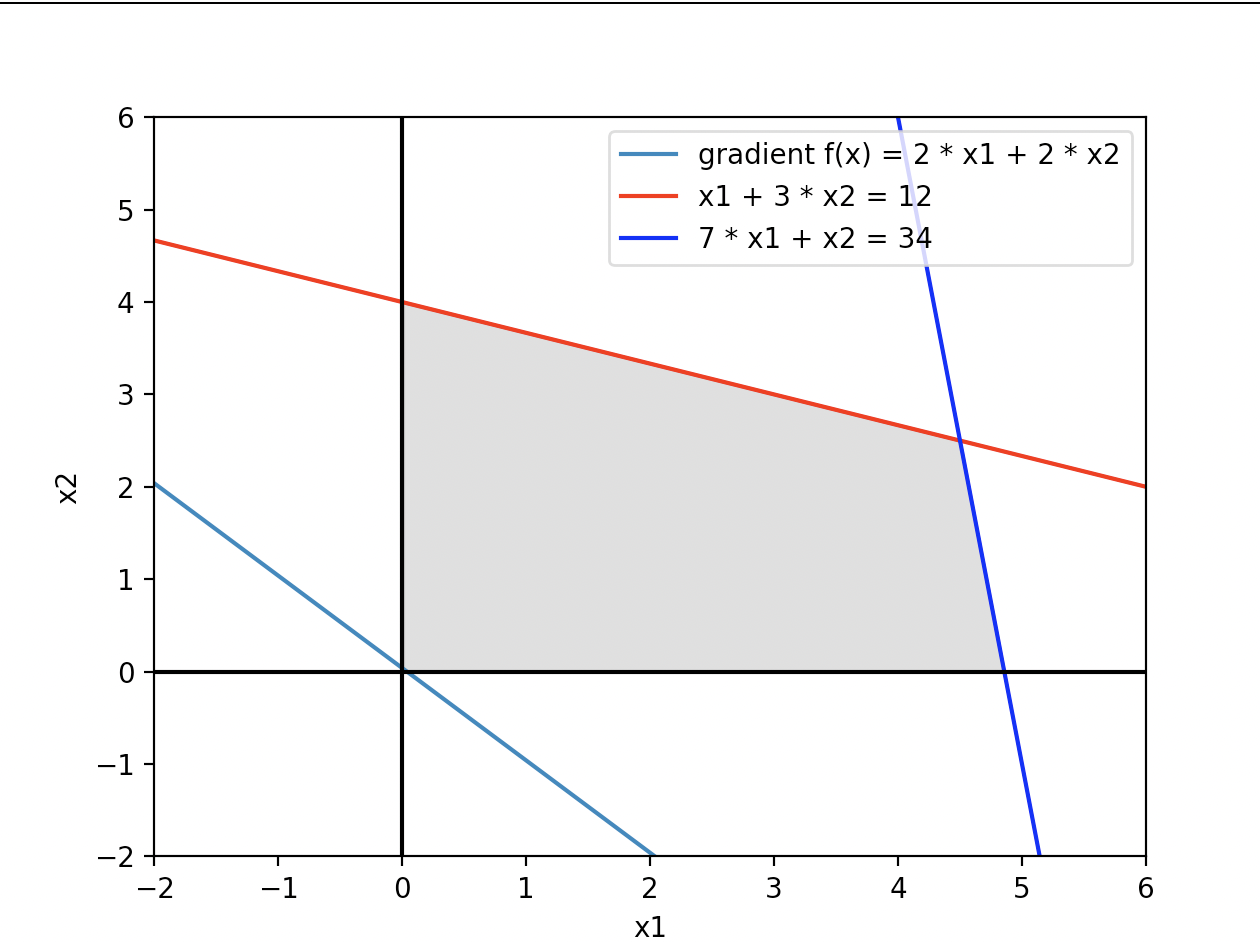
Найдем значение функции в точке максимума.

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

Получим значение равное F(x)max =

## 1.7 Минимум функции

Для нахождения минимума функции будем перемещать прямую в сторону антиградиента. Отметим на графике найденную точку (Рисунок 1.5).



**Рисунок 1.5 – Точка минимума функции**

Найдем координаты точки минимума:

Координаты точки минимума: {0; 0}

В результате получим точку с координатами (0, 0). Найдем значение функции в этой точке.

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

Получим значение равное F(x)min =

Ответ:

F(x)max = 14

F(x)min = 0

**СИМПЛЕКСНЫЙ МЕТОД**

* 1. **Постановка задачи**

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

Для изготовления двух видов тары (бочек и ящиков) употребляется два вида древесины. Расход древесины каждого вида на каждое изделие, объём ресурсов и прибыль на единицу изделия заданы в таблице 1.

*Таблица 1 Условия задачи*

|  |  |  |  |
| --- | --- | --- | --- |
| Изделия | Расход древесины | | Прибыль на единицу продукции, ден. ед. |
| I | II |
| Бочки | 0.15 | 0.2 | 1.5 |
| Ящики | 0.2 | 0.1 | 1.2 |
| Объём ресурсов | 60 | 40 |  |

Требуется определить сколько ящиков и бочек должен изготовить завод, чтобы прибыль была максимальной.

**1.2 Математическая модель задачи**

Пусть х1 – количество выпускаемых бочек, х2 – количество выпускаемых ящиков. Прибыль от продажи сырья составит 1.5х1 + 1.2х2, прибыль требуется максимизировать.

Ограничения задачи:

Таким образом, переходим к задаче линейного программирования:

Приведем задачу к канонической форме. Для этого в левые части ограничений вводим дополнительные переменные: x3 ≥ 0, х4 ≥ 0. Эти переменные выбираются так, чтобы они обращали неравенства в равенства.

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

Векторы 𝐴3, 𝐴4 являются линейно независимыми единичными векторами 3х-мерного пространства и образуют базис этого пространства [2].

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

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

Которому соответствует первоначальный опорный план

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

В левый столбец Таблицы 1.2 запишем переменные 𝑥3, 𝑥4 образующие базис, в верхней строке – небазисные переменные 𝑥1, 𝑥2. В строке 𝑐j запишем коэффициенты целевой функции, соответствующие небазисным переменным с1 = 1.5, с2 = 1.2. В столбце запишем коэффициенты целевой функции, соответствующие базисным переменным Столбец, определяемый переменной 𝑥1, состоит из коэффициентов вектора . Аналогично, столбец, определяемый переменной 𝑥2, состоит из коэффициентов вектора . Крайний правый столбец заполняется элементами столбца , в нем же в результате вычислений получаем оптимальный план.

Заполнение f-строки (Таблица 1.3). Найдем относительные оценки ∆1, ∆2 и значение целевой функции 𝑄.

*Таблица 1.2 – Начальная симплекс-таблица задачи о максимальном доходе*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | сj | 1.5 | 1.2 |  |
|  |  | X1 | X2 |  |
| 0 | X3 | 0.15 | 0.12 | 60 |
| 0 | X4 | 0.2 | 0.1 | 40 |
|  | F | -1.5 | -1.2 | 0 |
|  |  | Δ1 | Δ2 | Q |

*Таблица 1.3 – Заполнение f-строки*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | сj | 1.5 | 1.2 |  |  |
|  |  | X1 | X2 |  |  |
| 0 | X3 | 0.15 | 0.2 | 60 | 60 / 0.15 = 400 |
| 0 | X4 | 0.2 | 0.1 | 40 | 40 / 0.2 = 200 *min* |
|  | f | -1.5 | -1.2 | 0 |  |
|  |  | Δ1 | Δ2 | Q |  |

Для оптимальности опорного решения в задаче на максимум требуется выполнение не отрицательности всех относительных оценок ∆i ≥ 0. Так как оценки ∆1= −1.5, ∆2= −1.2 в f-строке отрицательны, то это свидетельствуют о возможности улучшения полученного решения. Наибольшая по модулю отрицательная оценка ∆1= −1.5. В базис будет включена соответствующая ей небазисная переменная 𝑥1. Составим отношения свободных членов к положительным элементам разрешающего столбца. Данные отношения приведены справа от таблицы. Наименьшему частному соответствует строка с переменной 𝑥4. Эта переменная исключается из базиса. В Таблице 1.3 разрешающий столбец и разрешающая строка выделены. Разрешающим элементом является число 𝑎21 = 0.2.

Далее построим новую симплекс-таблицу. Ниже поэтапно демонстрируется процесс заполнения новой симплекс-таблицы (Таблицы 1.4 ).

*Таблица 1.4 – Новая симплекс-таблица*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | сj | 0 | 1.2 |  |
|  |  | X4 | X2 |  |
| 0 | X3 |  |  |  |
| 1.5 | X1 | 5 |  |  |
|  | f |  |  |  |
|  |  | Δ1 | Δ2 | Q |

В Таблице 1.4 переменные 𝑥1 и 𝑥4 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 1.5 – Симплекс преобразования*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | сj | 7.5 | 3 |  |
|  |  | X4 | X2 |  |
| 0 | X3 | -0.75 |  |  |
| 1.5 | X1 | 5 | 0.5 | 200 |
|  | F | 7.5 |  |  |
|  |  | Δ1 | Δ2 | Q |

*Таблица 1.6 – Итерация 0*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | сj | 0 | 1.2 |  |  |
|  |  | X4 | X2 |  |  |
| 0 | X4 | -0.75 | 0.125 | 30 | 30 / 0.125 = 240 |
| 1.5 | X6 | 5 | 0.5 | 200 | 200 / 0.5 = 400 |
|  | f | 7.5 | -0.45 | 300 |  |
|  |  | Δ1 | Δ2 | Q |  |

Остальные элементы (Таблица 1.6) рассчитываются по «правилу прямоугольника».

Базисное решение, которое дает последняя таблица

Это решение не является оптимальным, так как в f-строке имеются отрицательная оценка ∆2.

Выполняем следующую итерацию до тех пор, пока в таблице f-строка не будет отрицательных оценок.

Так как оценка ∆2= -0.45 в f-строке отрицательна, то это свидетельствуют о возможности улучшения полученного решения. Наибольшая по модулю отрицательная оценка ∆1= −0.45. 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 𝑥4. Эта переменная исключается из базиса. В Таблице 1.6 разрешающий столбец и разрешающая строка выделены. Разрешающим элементом является число 𝑎12 = 0.125.

Далее построим новую симплекс-таблицу. Ниже поэтапно демонстрируется процесс заполнения новой симплекс-таблицы (Таблицы 1.7 ).

*Таблица 1.7 – Новая симплекс-таблица*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | сj | 0 | 0 |  |
|  |  | X4 | X3 |  |
| 1.2 | X2 |  |  |  |
| 1.5 | X1 | 5 |  |  |
|  | F |  |  |  |
|  |  | Δ1 | Δ2 | Q |

В Таблице 1.7 переменные 𝑥3 и 𝑥7 меняются местами вместе с коэффициентами 𝑐𝑗. Разрешающий элемент заменяется на обратный. В Таблице 1.8 элементы разрешающей строки делятся на разрешающий элемент. Элементы разрешающего столбца делятся на разрешающий элемент и меняют знак.

*Таблица 1.8 – Симплекс преобразования*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | сj | 0 | 0 |  |
|  |  | X4 | X3 |  |
| 1.2 | X1 | -6 | 8 | 240 |
| 1.5 | X2 |  | -4 |  |
|  | F |  | 3.6 |  |
|  |  | Δ1 | Δ2 | Q |

*Таблица 1.9 – Итерация 1*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | сj | 0 | 0 |  |
|  |  | X4 | X3 |  |
| 1.2 | X1 | -6 | 8 | 240 |
| 1.5 | X2 | 8 | -4 | 80 |
|  | f | 4.8 | 3.6 | 408 |
|  |  | Δ1 | Δ2 | Q |

Остальные элементы (Таблица 1.9) рассчитываются по «правилу прямоугольника».

Базисное решение, которое дает последняя таблица

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

Подставляем базисное решение, которое дает последняя таблица

Где n – количество итераций

Проверим решение по «правилу прямоугольника».

((0.125\*300) - (-0.45 \* 30)) / (0.125) = 408

Таким образом, фабрика должна выпускать в течении недели 𝑥1 = 240 шт. ящиков и x2 = 80 шт. бочек. Тогда фабрика получит максимальный доход от продажи 456 [ден.ед].

**ЗАКЛЮЧЕНИЕ**

В ходе выполнения курсовой работы были изучены основы линейного программирования и многокритериальной оптимизации в качестве методов теории принятия решений. Для решения задач были реализованы программы на языке программирования Python, произведены вычисления в электронных таблицах Excel, мануальные расчёты, изучена теория и практическое применение задач ТПР.

Линейное программирование состоит в нахождении экстремального значения линейной функции многих переменных при наличии линейных ограничений, связывающих эти переменные, и предоставляет собой метод для оптимизации операций в рамках определенных ограничений. Он используется, чтобы сделать управленческие процессы более эффективными, а работу предприятий более рентабельной.

**СПИСОК ИНФОРМАЦИОННЫХ ИСТОЧНИКОВ**

1. Информатика: Учебник / Под ред. проф. Н. В. Макаровой.

— М.: Финансы и статистика, 2015. — 768 с.

1. Богданова Е.Л. Оптимизация в проектном менеджменте:

линейное программирование: учебное пособие / Е.Л. Богданова, К.А.

Соловейчик, К.Г. Аркина. – СПб.: Университет ИТМО, 2017. – 165 с.

1. Зобнина О.В., Дю А.И., Бабаева Ю.А. Многокритериальная оптимизация// Научно-образовательный журнал для студентов и преподавателей «StudNet». – 2021. - №1.
2. Теория принятия решений: учебное пособие / С. М.

Бородачёв. - Екатеринбург : Изд-во Урал, ун-та, 2014. - 124 с.

1. Теория принятия решений: учебное пособие. Изд.2-е, перераб. / Е.Е. Воробьева, В.Ю. Емельянов; Балт. гос. техн. ун-т. – СПб., 2018. – 136 с.

Приложение А

Код реализации метода Парето на языке C++.

Листинг 1. Реализация Метода Парето.

#include <iostream>

const char\* university\_names[] =

{

"RANHiGS",

"RTU MIREA",

"MGTU im. Baumana",

"Moscow State University",

"MTUSI",

"MAI",

"MEI",

"Politeh",

"Sinergia",

};

const float variants[][5] =

{

{250000, 4.4, 144, 35, 50},

{200000, 4.9, 100, 20, 100},

{270000, 4.6, 115, 120, 70},

{226000, 4.9, 100, 30, 120},

{230000, 4.5, 120, 100, 40},

{210000, 3.6, 90, 90, 120},

{220000, 4.1, 130, 95, 90},

{180000, 4.0, 100, 150, 100},

{200000, 3.5, 90, 80, 130},

};

const int num\_vars = sizeof(variants) / sizeof(\*variants);

const int num\_conds = sizeof(\*variants) / sizeof(float);

const float conditions[num\_conds] = { -1, 1, -1, -1, 1, };

const int priority[num\_conds] = { 5, 3, 1, 4, 2 };

const float rating\_min = 3.9;

const float time\_min = 60;

int main()

{

for(int i = 0; i < num\_vars; i++)

{

if(variants[i][1] < rating\_min)

continue;

if(variants[i][3] > time\_min)

continue;

int best\_prior = 0;

for(int j = i + 1; j < num\_vars; j++)

{

int num\_comps[3] = {0}; // 0 - лучше, 1 - равно, 2 - хуже

int num\_priors[5] = {0};

for(int k = 0; k < num\_conds; k++)

{

if(variants[j][k] \* conditions[k] > variants[i][k] \* conditions[k])

{

num\_comps[0]++;

num\_priors[priority[j] - 1]++;

}

else if(variants[j][k] \* conditions[k] == variants[i][k] \* conditions[k])

{

num\_comps[1]++;

num\_priors[priority[j] - 1]++;

}

else

num\_comps[2]++;

}

if(num\_comps[0] && num\_comps[0] + num\_comps[1] == num\_conds)

std::cout << university\_names[j] << " > " << university\_names[i] << std::endl;

}

}

return 0;

}

Приложение Б

Код реализации метода Электра II на языке Python.

Листинг 2. Реализация метода Электра II.

import numpy as np

alt = [

[250000, 4.4, 144, 35],

[200000, 4.9, 100, 20],

[226000, 4.9, 100, 30],

[230000, 4.5, 120, 100],

[220000, 4.1, 130, 95],

[180000, 4.0, 100, 150],

]

m = [3, 4, 2, 5]

appr = [-1, 1, -1, -1]

for i in range(len(alt)):

for j in range(i + 1, len(alt)):

P = np.zeros((1, 4))[0]

for k in range(len(m)):

if(alt[i][k] \* appr[k] > alt[j][k] \* appr[k]):

P[k] = m[k]

print(f'P{i+1}{j+1} = ', end='')

print(\*P, sep=' + ', end=' = ')

print(sum(P))

N = np.zeros((1, 4))[0]

for k in range(len(m)):

if(alt[i][k] \* appr[k] < alt[j][k] \* appr[k]):

N[k] = m[k]

print(f'N{i+1}{j+1} = ', end='')

print(\*N, sep=' + ', end=' = ')

print(sum(N))

print(f'D{i+1}{j+1} = P{i+1}{j+1} / N{i+1}{j+1} = ', end='')

if sum(N) == 0:

print('inf - не подходит')

else:

D = sum(P) / sum(N)

print(D, end='')

if(D > 1):

print(' - принимаем')

else:

print(' - отбрасываем')

print(f'D{j+1}{i+1} = N{j+1}{i+1} / P{j+1}{i+1} = ', end='')

if(sum(P) == 0):

print('inf - не подходит', end='\n\n')

else:

D = sum(N) / sum(P)

print(D, end='')

if(D > 1):

print(' - принимаем', end='\n\n')

else:

print(' - отбрасываем', end='\n\n')

Приложение В

*Листинг 3 - Код практической работы № 3*

|  |
| --- |
| from data import matrix   def synthesis\_of\_priorities(matrix):  sum\_V = 0   for line in matrix:  V = round(pow(line[0] \* line[1] \* line[2] \* line[3] \* line[4], 0.2), 3)  line.append(V)  sum\_V = round(sum\_V + V, 3)   for line in matrix:  W = round(line[5] / sum\_V, 3)  line.append(W)   return matrix   def synthesis\_of\_local\_priorities(matrix, id):  S1 = 0  S2 = 0  S3 = 0  S4 = 0  S5 = 0   for line in matrix:  S1 = S1 + line[0]  S2 = S2 + line[1]  S3 = S3 + line[2]  S4 = S4 + line[3]  S5 = S5 + line[4]   P1 = S1 \* matrix[0][6]  P2 = S2 \* matrix[1][6]  P3 = S3 \* matrix[2][6]  P4 = S4 \* matrix[3][6]  P5 = S5 \* matrix[4][6]   lamda = P1 + P2 + P3 + P4 + P5  index\_sogl = (lamda - 5) / 4  res\_sogl = index\_sogl / 1.12  print("Лямда =", lamda)  if res\_sogl <= 0.10:  print(f'\n> Матрица {id} согласована!')  print("\n------------------------------------------------------------")  else:  print('Матрица не согласована', res\_sogl)  print("\n------------------------------------------------------------")   def print\_matrix(matrix, id):   print()  print(id, end="\t\t")  for i in range(1, 6):  print(str(i), end="\t\t")  print('V\t\tW')   for j in range(0, 5):  for i in range(0, 7, 1):  if i == 0:  print(str(j + 1), end="\t\t")   line = round(float(matrix[j][i]), 1)  print(line, end='\t ')  print()   if \_\_name\_\_ == '\_\_main\_\_':  new\_purpose = matrix.purpose  main\_matrix = synthesis\_of\_priorities(new\_purpose)  print\_matrix(main\_matrix, 'Pur')  synthesis\_of\_local\_priorities(main\_matrix, 'Цель')   new\_durable = matrix.durable  K1 = synthesis\_of\_priorities(new\_durable)  print\_matrix(K1, 'K1')  synthesis\_of\_local\_priorities(K1, 'Проходной балл')   new\_controllability = matrix.controllability  K2 = synthesis\_of\_priorities(new\_controllability)  print\_matrix(K2, 'K2')  synthesis\_of\_local\_priorities(K2, 'Кол-во бюджетных мест')   new\_stability = matrix.stability  K3 = synthesis\_of\_priorities(new\_stability)  print\_matrix(K3, 'K3')  synthesis\_of\_local\_priorities(K3, 'Цена')   new\_count\_layer = matrix.count\_layers  K4 = synthesis\_of\_priorities(new\_count\_layer)  print\_matrix(K4, 'K4')  synthesis\_of\_local\_priorities(K4, 'Рейтинг')   new\_count\_bearings = matrix.count\_bearings  K5 = synthesis\_of\_priorities(new\_count\_bearings)  print\_matrix(K5, 'K5')  synthesis\_of\_local\_priorities(K5, 'Время поездки') |

*Листинг 3.1 - Код практической работы № 3*

|  |
| --- |
| purpose = [[1, 3, 3, 5, 5],  [0.33, 1, 3, 3, 3],  [0.33, 0.33, 1, 3, 5],  [0.2, 0.33, 0.33, 1, 3],  [0.2, 0.33, 0.2, 0.33, 1]]  durable = [[1, 0.33, 0.33, 0.11, 0.11],  [3, 1, 0.33, 0.11, 0.11],  [3, 3, 1, 0.14, 0.14],  [9, 9, 7, 1, 1],  [9, 9, 7, 1, 1]]  controllability = [[1, 0.2, 0.14, 0.11, 0.11],  [5, 1, 0.14, 0.11, 0.11],  [7, 7, 1, 0.33, 0.33],  [9, 9, 3, 1, 1],  [9, 9, 3, 1, 1]]  stability = [[1, 0.2, 0.14, 0.11, 0.11],  [5, 1, 0.14, 0.11, 0.11],  [7, 7, 1, 0.33, 0.33],  [9, 9, 3, 1, 1],  [9, 9, 3, 1, 1]]  count\_layers = [[1, 0.33, 0.2, 0.11, 0.11],  [3, 1, 0.14, 0.11, 0.14],  [5, 7, 1, 0.2, 0.33],  [9, 9, 5, 1, 3],  [7, 7, 3, 0.2, 1]]  count\_bearings = [[1, 3, 5, 7, 9],  [0.33, 1, 3, 5, 9],  [0.2, 0.33, 1, 3, 5],  [0.14, 0.2, 0.33, 1, 5],  [0.11, 0.11, 0.2, 0.2, 1]] |

Приложение Г

Код реализации симплексного метода на языке python

*Листинг 4. Реализация симплексного метода.*

st=2

n = st + 1

sv=2

k = sv + n

mas = [[0.15, 0.2, 1, 0, 60], [0.2, 0.1, 0, 1, 40], [-1.5, -1.2, 0, 0, 0]]

print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

print(" Таблица: ")

for i in range(n):

for j in range(k):

print(" "+str(mas[i][j]),end = ' ')

print(" ")

print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

kpol=0

for j in range (sv):

if (mas[n-1][j] > 0):

kpol+=1

print("целевая функция равняется "+str(mas[n-1][k-1])+"\n")

count=0

while (kpol != sv):

print("\nИТЕРАЦИЯ "+str(count)+"\n")

kpol = 0

max = 0

for j in range (sv):

if (abs(mas[n-1][j]) > abs(mas[n-1][max])):

max = j

print("этот столбец ведущий "+str(mas[n-1][max]))

t=mas[n-1][max]

print("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n")

myn=0

r=0

z=(mas[myn][k-1]) / (mas[myn][max])

for i in range(st):

if (mas[i][max])>0:

r= (mas[i][k-1]) / (mas[i][max])

if (z > r):

myn=i

z=r

print("эта строка ведущая "+str(mas[myn][k-1]))

print("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n")

print("коэффицент "+str(mas[myn][max]))

l=mas[myn][max]

for i in range(n):

p=mas[i][max]

if (i != myn):

for j in range(k):

if j!=max:

mas[i][j]=(mas[i][j]\*mas[myn][max]-mas[myn][j] \* p)/mas[myn][max]

for j in range(k):

if j!=max:

mas[myn][j]=mas[myn][j] / l

for i in range(n):

if i!=myn:

mas[i][max]=-mas[i][max]/l

print("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n")

print(" Таблица: ")

for i in range(n):

for j in range(k):

print(" " + str(round(mas[i][j],1)), end=" ")

print("")

print("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n")

print("целевая функция равняется " + str(round(mas[n - 1][k - 1],1)) + "\n")

for j in range(sv):

(mas[n-1][j] > 0):

kpol+=1

count+=1

print("решение оптимально, целевая функция равняется "+str(round(mas[n - 1][k - 1],1))+"\n")