Оглавление

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.

1.1 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

1.2 Указание верхних/нижних границ критериев.

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.

№	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

Таблица 3. Список альтернатив с учетом границ

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

1.3 Субоптимизация

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: рейтинг не меньше 4.6, время поездки не больше 30 минут. Отбросим варианты, которые не удовлетворяют данным ограничениям и составим таблицу 4.

№	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

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

1.4 Лексикографическая оптимизация

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.

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

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 = 0.555555555555556 - 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	x	-	-	-	1.8	1.8
2	inf	x	inf	inf	inf	3.0
3	inf	-	x	inf	3.67	3.0
4	1.8	-	-	x	-	1.8
5	-	-	-	1.33	x	1.8
6	-	-	-	-	-	x

По матрице построен граф предпочтений (Рисунок 1).

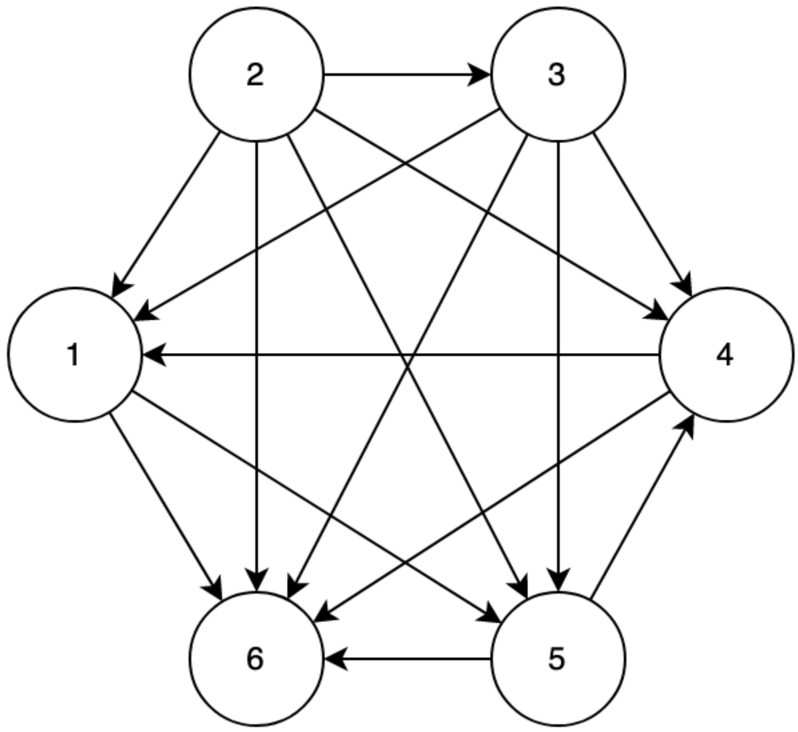

Рисунок 1 – Вид графа предпочтений

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 C=2.0

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

По этой матрице построен граф предпочтений (Рисунок 2).

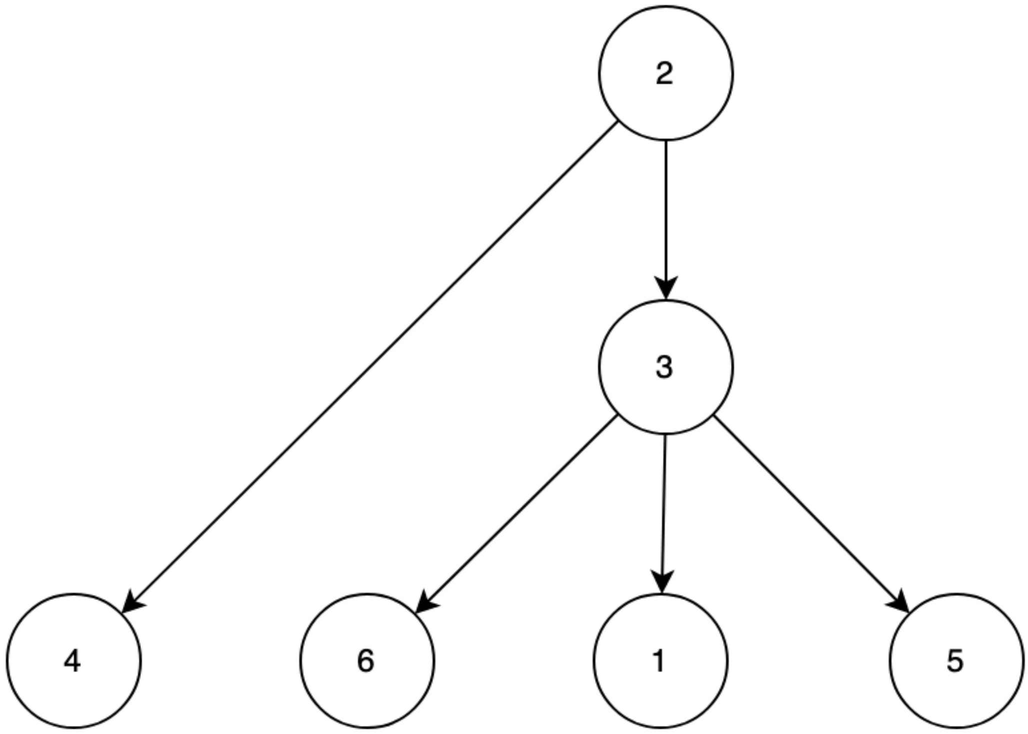

Рисунок 2 – Вид графа предпочтений для случая порога принятия решений C = 2.0

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).

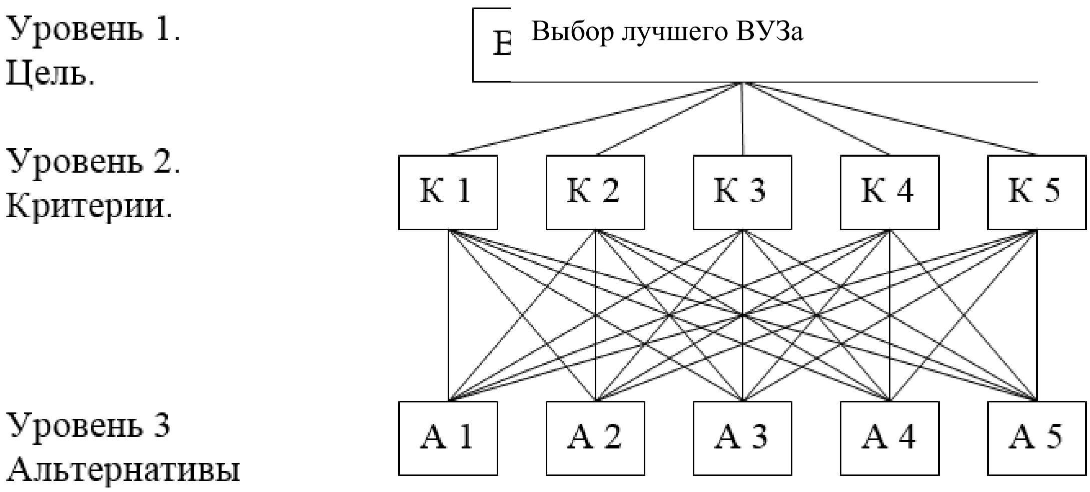

Рисунок 1.4 – Полная доминантная иерархия

Критерии:

К1 – Проходной балл

К2 – Кол-во бюджетных мест

К3 – Цена

К4 – Рейтинг

К5 – Время поездки

Альтернативы:

А1 – Синергия

А2 - РАНХиГС

А3 - МГТУ им. Баума

А4 - РТУ МИРЭА

А5 - 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

$V1 = (1x3x3x5x\ 5)^{1/5} = 2.954$

Строка № 2

$V2 = (1/3x1x3x3x3)^{1/5} = 1.551$

Строка № 3

$V3 = (1/3x1/3x1x3x5)^{1/5} = 1,107$

Строка № 4

$V4 = (1/5x1/3x1/3x1x7)^{1/5} = 0.581$

Строка № 5

$V5 = (1/5x1/3x1/5x1/7x1)^{1/5} = 0.338$

Находим сумму Vi и можем все вычисления переписывать в

Таблицу 1.2.

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

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

Строка № 1

$W_{21}= 2,954/6.531= 0.452$

Строка № 2

$W_{22}= 1,551/6.531= 0.237$

Строка № 3

$W_{23}= 1,107/6.531= 0.169$

Строка № 4

$W_{24}= 0.581/6.531= 0.088$

Строка № 5

$W_{25}= 0.338/6.531= 0.051$

В результате получаем вектор приоритетов:

$W_{2i} = (0.452; 0.237; 0.169; 0.088; 0.051)$

Таблица 1.3 – Таблица критерия “Проходной балл”

K1	A1	A2	A3	A4	A5	V_{K1Y}	W_{3K1Y}
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

ΣV_{K1Y}	8.5

Строка № 1

$V_{K11} = (1x1/3x1/3x1/9x1/9)^{1/5} = 0.267$

Строка № 2

$V_{K12} = (3x1x1/3x1/9x1/9)^{1/5} = 0.415$

Строка № 3

$V_{K13} = (3x3x1x1/7x1/7)^{1/5} = 0.712$

Строка № 4

V_{K14}=(9x9x7x1x1)1/5= 3.553

Строка № 5

V_{K15}=(9x9x7x1x1)1/5= 3.55

Находим сумму V_{K1Y} и можем переписывать вычисления в Таблицу 1.3.

$\sum V_{K1Y} = V_{K11}+V_{K12}+V_{K13}+V_{K14}+V_{K15}$ = 0.267+0.415+0.712+3,553+3,553 = 8,5

И каждое из чисел VK1Y делим на $\sum V_{K1Y}$, в результате найдем важность приоритетов.

Строка № 1

W_{3K11}= 0.267/8,5= 0.031

Строка № 2

W_{3K12}= 0.415/8,5= 0.048

Строка № 3

W_{3K13}= 0.712/8,5= 0.083

Строка № 4

W_{3K14}= 3,553/8,5= 0.418

Строка № 5

W_{3K15}= 3,553 /8,5= 0.418

В результате получаем вектор приоритетов:

W_{3K1Y} = (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	V_{K2Y}	W_{3K2Y}
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
ΣV_{K2Y}						7.995	

Строка № 1

$V_{K21} = (1x1/5x1/7x1/9x1/9)^{1/5}$= 0.203

Строка № 2

$V_{K22} = (5x1x1/7x1/9x1/9)^{1/5}$= 0.388

Строка № 3

$V_{K23} = (7x7x1x1/3x1/3)^{1/5}$= 1.403

Строка № 4

$V_{K24}=(9x9x3x1x1)^{1/5}$= 3

Строка № 5

$V_{K25}=(9x9x3x1x1)^{1/5}$= 3

Находим сумму V_{K2Y} и можем переписывать вычисления в Таблицу 1.4.

$\sum V_{K2Y}$= 0.203+0.388+1.403+3+3 = 7,994.

И каждое из чисел V_{K2Y} делим на $\sum V_{K2Y}$, в результате найдем важность приоритетов.

Строка № 1

W_{3K21}= 0.203/7,995= 0,025

Строка № 2

W_{3K22}= 0.388/7,995 = 0,048

Строка № 3

W_{3K23}= 1.403/7,995= 0,1755

Строка № 4

W_{3K24}= 3/7,995= 0,375

Строка № 5

W_{3K25}= 3/7,995= 0,375

В результате получаем вектор приоритетов:

W_{3K2Y} = (0,025; 0,048; 0,1755; 0,375; 0,375)

Таблица 1.5 - Таблица критерия "Цена"

K2	A1	A2	A3	A4	A5	V_{K2Y}	W_{3K2Y}
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
ΣV_{K2Y}						7.995	

Строка № 1

$V_{K31} = (1x1/3x1/5x1/9x1/7)^{1/5} = 0.254$

Строка № 2

$V_{K32} = (3x1x1/3x1/9x1/7)^{1/5} = 0.4366$

Строка № 3

$V_{K33} = (5x3x1x1/7x1/3)^{1/5} = 0.9349$

Строка № 4

$V_{K34} = (9x9x7x1x2)^{1/5} = 4.082$

Строка № 5

$V_{K35} = (7x7x3x1/2x1)^{1/5} = 2.3618$

Находим сумму V_{K3Y} и можем переписывать вычисления в Таблицу 1.5.

$\sum V_{K3Y} = 0.254+0.4366+0.9349+4.082+2.3618 = 8,0693$

И каждое из чисел V_{K3Y} делим на $\sum V_{K3Y}$, в результате найдем важность приоритетов.

Строка № 1

$W_{3K31} = 0.254/8,0693 = 0,0314$

Строка № 2

$W_{3K32} = 0.4366/8,0693 = 0,0541$

Строка № 3

$W_{3K33} = 0.9349/8,0693 = 0,1158$

Строка № 4

W_{3K34}= 4.082/8,0693= 0,5058

Строка № 5

W_{3K35}= 2.3618/8,0693= 0,2926

В результате получаем вектор приоритетов:

W_{3K3Y} = (0,0314; 0,0541; 0,1158; 0,5058; 0,2926)

Таблица 1.6 – Таблица критерия “Рейтинг”

K4	A1	A2	A3	A4	A5	V_{K4Y}	W_{3K4Y}
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
ΣV_{K4Y}						8.224	

Строка № 1

$V_{K41} = (1x1/3x1/5x1/9x1/9)^{1/5}$= 0.2415

Строка № 2

$V_{K42} = (3x1x1/7x1/9x1/7)^{1/5}$= 0.3685

Строка № 3

$V_{K43} = (5x7x1x1/5x1/3)^{1/5}$= 1.1846

Строка № 4

$V_{K44}=(9x9x5x1x3)^{1/5}$= 4.1391

Строка № 5

$V_{K45}=(7x7x3x1/5x1)^{1/5}$= 2.290

Находим сумму V_{K4Y} и можем переписывать вычисления в Таблицу 1.6.

$\sum V_{K4Y}$= 0.2415+0.3685+1.1846+4.1391+2.290 = 8.224

И каждое из чисел V_{K4Y} делим на $\sum V_{K4Y}$, в результате найдем важность приоритетов.

Строка № 1

W_{3K41}= 0.2415/8.224= 0.029

Строка № 2

W_{3K42}= 0.3685/8.224= 0.0448

Строка № 3

W_{3K43}= 1.1846/8.224= 0.144

Строка № 4

W_{3K44}= 4.139/8.224= 0.5032

Строка № 5

W_{3K45}= 2.290/8.224= 0.2784

В результате получаем вектор приоритетов:

W_{3K4Y} = (0.029; 0.0448; 0.144; 0.5032; 0.2784)

Таблица 1.7 – Таблица критерия “Время поездки”

K5	A1	A2	A3	A4	A5	V_{K5Y}	W_{3K5Y}
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
ΣV_{K5Y}						7.839	

Строка № 1

$V_{K51} = (1x3x5x7x9)^{1/5}= 3.9362$

Строка № 2

$V_{K52} = (1/3x1x3x5x9)^{1/5}= 2.1411$

Строка № 3

$V_{K53} = (1/5x1/3x1x3x5)^{1/5}= 1$

Строка № 4

$V_{K54}=(1/7x1/5x1/3x1x5)^{1/5}= 0.5439$

Строка № 5

$V_{K55}=(1/9x1/9x1/5x1/5x1)^{1/5}= 0.2181$

Находим сумму V_{K5Y} и можем переписывать вычисления в Таблицу 1.7.

$\sum V_{K5Y}= 3.9362+2.1411+1+0.5439+0.2181= 7.839.$

И каждое из чисел V_{K4Y} делим на $\sum V_{K5Y}$, в результате найдем важность приоритетов.

Строка № 1

$W_{3K51}= 3.9362/7.839= 0.5021$

Строка № 2

$W_{3K52}= 2.1411/7.839= 0.2731$

Строка № 3

$W_{3K53}= 1/7.839= 0.1275$

Строка № 4

$W_{3K54}= 0.5439/7.839= 0.0693$

Строка № 5

$W_{3K55}= 0.2181/7.839= 0.0278$

В результате получаем вектор приоритетов:

$W_{3K5Y} = (0.5021; 0.2731; 0.1275; 0.0693; 0.0278)$

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.

$S_1 = 1 + 1/3 + 1/3 + 1/5 + 1/5 = 31/15$

$S_2 = 3 + 1 + 1/3 + 1/3 + 1/3 = 5$

$S_3 = 3 + 3 + 1 + 1/3 + 1/5 = 113/15$

$S_4 = 5 + 3 + 3 + 1 + 1/3 = 37/3$

$S_5 = 5 + 3 + 5 + 3 + 1 = 17$

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.

$P_1 = S_1$ x $W_{21} = 31/15$ x $0,452 = 0,934$

$P_2 = S_2$ x $W_{22} = 5$ x $0,237 = 1,185$

$P_3 = S_3$ x $W_{23} = 113/15$ x $0,169 = 1,2769$

$P_4 = S_4$ x $W_{24} = 37/3$ x $0.089 = 1,098$

$P_5 = S_5$ x $W_{25} = 17$ x $0.0518 = 0,8807$

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.

$\lambda_{max} = P_1 + P_2 + P_3 + P_4 + P_5 = 0,934 + 1,185 + 1,2769 + 1,098 + 0,8807 = 5,377$

Отклонение от согласованности выражается индексом согласованности.

ИС = $(\lambda_{max} – n)/(n-1) = (5,377 – 5)/(5-1) = 0.09445$

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.

$S_{1K1} = 1 + 3 + 3 + 9 + 9 = 25$

$S_{2K1} = 1/3 + 1 + 3 + 9 + 9 = 22,33$

$S_{3K1} = 1/3 + 1/3 + 1 + 7 + 7 = 15.66$

S_{4K1} = 1/9 + 1/9 + 1/7 + 1 + 1 = 2,365

S_{5K1} = 1/9 + 1/9 + 1/7 + 1 + 1 = 2,365

Затем полученный результат умножается на компоненту нормализованного вектора приоритетов.

$P_{1K1} = S_1 \text{ x } W_{3K11}$ = 25 x 0,031 = 0,786

$P_{2K1} = S_2 \text{ x } W_{3K12}$ = 22,33x 0,048 = 1,09

$P_{3K1} = S_3 \text{ x } W_{3K13}$ = 15.66 x 0,083 = 1,312

$P_{4K1} = S_4 \text{ x } W_{3K14}$ = 2,365x 0,418 = 0,988

$P_{5K1} = S_5 \text{ x } W_{3K15}$ = 2,365x 0,418 = 0,988

Находим пропорциональность предпочтений.

$\lambda_{max\ K1}$ = 0,786 + 1,09 + 1,312 + 0,988 + 0,988 = 5.167

Отклонение от согласованности выражается индексом согласованности.

$$ИС_{K1} = (\lambda_{max\ K1} – n)/(n-1) = (5.167– 5)/(5-1) = 0,0417$$

Найдем отношением согласованности ОС.

$$ОС_{K1} = ИС/СИ = 0,0417/1,12 = 0,0372$$

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.

S_{1K2} = 1 + 5 + 7 + 9 + 9 = 31

S_{2K2} = 1/5 + 1 + 7 + 9 + 9 = 26,2

S_{3K2} = 1/7 + 1/7 + 1 + 3 + 3 = 7.285

S_{4K2} = 1/9 + 1/9 + 1/3 + 1 + 1 = 2.55

S_{5K2} = 1/9 + 1/9 + 1/3 + 1 + 1 = 2,55

Затем полученный результат умножается на компоненту нормализованного вектора приоритетов.

$P_{1K2} = S_1 \text{ x } W_{3K21}$ = 31 x 0,025 = 0,7906

$P_{2K2} = S_2$ x $W_{3K22} = 26{,}2$x $0{,}048 = 1{,}272$

$P_{3K2} = S_3$ x $W_{3K23} = 7.285$x $0{,}1755= 1{,}2788$

$P_{4K2} = S_4$ x $W_{3K24} = 2.55$x $0{,}375= 0{,}9588$

$P_{5K2} = S_5$ x $W_{3K25} = 2.55$x $0{,}375= 0{,}9588$

Находим пропорциональность предпочтений.

$\lambda_{max\ K2} = 0{,}7906+ 1{,}272+ 1{,}2788+ 0{,}9588 + 0{,}9588 = 5{,}259$.

Отклонение от согласованности выражается индексом согласованности.

$$ИС_{K2} = (\lambda_{max\ K2} – n)/(n-1) = (5{,}259 – 5)/(5-1) = 0{,}0648$$

Найдем отношением согласованности ОС.

$$ОС_{K2} = ИС/СИ = 0{,}0648/1{,}12 = 0{,}05789$$

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.

$S_{1K3} = 1 + 3 + 5 + 9 + 7 = 25$

$S_{2K3} = 1/3 + 1 + 3 + 9 + 7 = 20.33$

$S_{3K3} = 1/5 + 1/3 + 1 + 7 + 3 = 11.533$

$S_{4K3} = 1/9 + 1/9 + 1/7 + 1 + 1/2 = 1.865$

$S_{5K3} = 1/7 + 1/7 + 1/3 + 2 + 1 = 3.619$

Затем полученный результат умножается на компоненту нормализованного вектора приоритетов.

$P_{1K3} = S_1$ x $W_{3K31} =25$x $0{,}0314 = 0{,}787$

$P_{2K3} = S_2$ x $W_{3K32} = 20.33$ x $0{,}0541= 1.1$

$P_{3K3} = S_3$ x $W_{3K33} = 11.533$ x $0{,}1158 = 1{,}336$

$P_{4K3} = S_4$ x $W_{3K34} = 1.865$ x $0{,}5058= 0.943$

$P_{5K3} = S_5$ x $W_{3K35} = 3.619$ x $0{,}2926= 1{,}059$

Находим пропорциональность предпочтений.

$\lambda_{max\ K3}$ = 0,787 + 1.1 + 1,336 + 0.943 + 1,059 = 5,226

Отклонение от согласованности выражается индексом согласованности.

$ИС_{К3} = (\lambda_{max\ К3} – n)/(n-1) = (5,226 – 5)/(5-1) = 0,056$

Найдем отношением согласованности ОС.

$ОС_{К3}$ = 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.

$S_{1К4}$ = 1 + 3 + 5 + 9 + 7 = 27

$S_{2К4}$ = 1/3 + 1 + 7 + 9 + 7 = 24,33

$S_{3К4}$ = 1/5 + 1/7 + 1 + 5 + 3 = 9.342

$S_{4К4}$ = 1/9 + 1/9 + 1/5 + 1 + 1/5 = 1,755

$S_{5К4}$ = 1/9 + 1/7 + 1/3 + 3 + 1 = 4.587

Затем полученный результат умножается на компоненту нормализованного вектора приоритетов.

$P_{1К4} = S_1$ x $W_{3К41}$ = 27 x 0.0293= 0,7931

$P_{2К4} = S_2$ x $W_{3К42}$ = 24,33x 0.0448= 1.0905

$P_{3К4} = S_3$ x $W_{3К43}$ = 9.342x 0.144= 1,3458

$P_{4К4} = S_4$ x $W_{3К44}$ = 1,755x 0.5032= 0,8835

$P_{5К4} = S_5$ x $W_{3К45}$ = 4.587x 0.2784= 1,277

Находим пропорциональность предпочтений.

$\lambda_{max\ К4}$ = 0,7931+ 1.0905 + 1,3458 + 0,8835 + 1,277= 5,3904.

Отклонение от согласованности выражается индексом согласованности.

$ИС_{К4} = (\lambda_{max\ К4} – n)/(n-1) = (5,3904 – 5)/(5-1) = 0,0976$

Найдем отношением согласованности ОС.

$ОС_{К4} = ИС/СИ = 0,0976/1,12 = 0,0871$

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.

$S_{1К5} = 1 + 1/3 + 1/5 + 1/7 + 1/9 = 1,7873$

$S_{2К5} = 3 + 1 + 1/3 + 1/5 + 1/9 = 1.268$

$S_{3К5} = 5 + 3 + 1 + 1/3 + 1/5 = 9.533$

$S_{4К5} = 7 + 5 + 3 + 1 + 1/5 = 16.2$

$S_{5К5} = 9 + 9 + 5 + 5 + 1 = 29$

Затем полученный результат умножается на компоненту нормализованного вектора приоритетов.

$P_{1К5} = S_1 \text{ x } W_{3К51} = 1,7873\text{x } 0.5021 = 0,8974$

$P_{2К5} = S_2 \text{ x } W_{3К52} = 1.268 \text{ x } 0.2731 = 0.3462$

$P_{3К5} = S_3 \text{ x } W_{3К53} = 9.533\text{x } 0.1275 = 1,216$

$P_{4К5} = S_4 \text{ x } W_{3К54} = 16.2 \text{ x } 0.0693 = 1.126$

$P_{5К5} = S_5 \text{ x } W_{3К55} = 29 \text{ x } 0,0278 = 0,806$

Находим пропорциональность предпочтений.

$\lambda_{max\ К5} = P_{1К5} + P_{2К5} + P_{3К5} + P_{4К5} + P_{5К5} = 0,8974 + 0.3462 + 1,216 + 1.126 + 0,806 = 5,3129$

Отклонение от согласованности выражается индексом согласованности.

$ИС_{К5} = (\lambda_{max\ К5} – n)/(n-1) = (5,3129 – 5)/(5-1) = 0,078$

Найдем отношением согласованности ОС.

$ОС_{К5} = ИС/СИ = 0,078/1,12 = 0,069$

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.

W_{2i} = (0.452; 0.237; 0.169; 0.088; 0.051)

W_{3K1Y} = (0.031; 0.048; 0.083; 0.418; 0.418)

W_{3K2Y} = (0,025; 0,048; 0,1755; 0,375; 0,375)

W_{3K3Y} = (0,0314; 0,0541; 0,1158; 0,5058; 0,2926)

W_{3K4Y} = (0.029; 0.0448; 0.144; 0.5032; 0.2784)

W_{3K5Y} = (0.5021; 0.2731; 0.1275; 0.0693; 0.0278)

Приоритеты альтернатив получим следующим образом:

$W_1 = W_{21}$ x $W_{3K11} + W_{22}$ x $W_{3K21} + W_{23}$ x $W_{3K31} + W_{24}$ x $W_{3K41} + W_{25}$ x W_{3K51} = 0,452 x 0,031 + 0,237 x 0,025 + 0,169 x 0,0314 + 0,088 x 0,029 + 0,051 x 0,5021 = 0,05425

$W_2 = W_{21}$ x $W_{3K12} + W_{22}$ x $W_{3K22} + W_{23}$ x $W_{3K32} + W_{24}$ x $W_{3K42} + W_{25}$ x W_{3K52} = 0,452 x 0.048 + 0,237 x 0.048 + 0,169 x 0,0541 + 0,088 x 0.0448+ 0,051 x 0.2731 = 0.0609

$W_3 = W_{21}$ x $W_{3K13} + W_{22}$ x $W_{3K23} + W_{23}$ x $W_{3K33} + W_{24}$ x $W_{3K43} + W_{25}$ x W_{3K53} = 0,452 x 0.083 + 0,237 x 0,1755+ 0,169 x 0,1158 + 0,088 x 0.144 + 0,051 x 0.1275 = 0.1186

$W_4 = W_{21}$ x $W_{3K14} + W_{22}$ x $W_{3K24} + W_{23}$ x $W_{3K34} + W_{24}$ x $W_{3K44} + W_{25}$ x W_{3K54} = 0,452 x 0.418 + 0,237 x 0,375 + 0,169 x 0,5058 + 0,088 x 0.5032 + 0,051 x 0.0693 = 0.4122

$W_5 = W_{21}$ x $W_{3K15} + W_{22}$ x $W_{3K25} + W_{23}$ x $W_{3K35} + W_{24}$ x $W_{3K45} + W_{25}$ x W_{3K55} = 0,452 x 0.418 + 0,237 x 0,375 + 0,169 x 0,2926 + 0,088 x 0.2784 + 0,051 x 0.0278 = 0.3539

Таким образом, приоритеты альтернатив равны:

Альтернатива A1 (Проходной балл) - W1 приоритет равен 0,05425;

Альтернатива A2 (Кол-во бюджетных мест) – W2 приоритет равен 0.0609;

Альтернатива A3 (Цена) – W3 приоритет равен 0.1186;

Альтернатива A4 (Рейтинг) – W4 приоритет равен 0.4122;

Альтернатива A5 (Время поездки) – W5 приоритет равен 0.3539;

Наиболее перспективным с позиции метода анализа иерархий признается выбор ВУЗа A4 - РТУ МИРЭА. Однако видно, что выбор A5 - МГУ, оказывается тоже неплохим.

ГРАФИЧЕСКИЙ МЕТОД

1.1 Постановка задачи

Решить задачу линейного программирования с двумя переменными графическим методом.

1.2 Данные индивидуального варианта

$$f(x) = 2x_1 + 2x_2 \to min/max$$

$$\begin{cases} x_1 + 3x_2 \le 12 \\ 7x_1 + x_2 \le 34 \\ x_1, x_2 \ge 0 \end{cases}$$

1.3 Подготовка данных

В среде Microsoft Excel добавим 4 столбца:

1. x_1 – значения от 0 до 10 с шагом 0,5;
2. $x_2 = (\ldots)$ – значения ограничения $(x_1 + 3x_2 \le 12)$;
3. $x_2 = (\ldots)$ – значения ограничения $(7x_1 + x_2 \le 34)$;
4. $x_2 = (\ldots)$ – значения целевой функции $2x_1 + 2x_2 = 0$.

Таблица 1.1 – Данные для графика

x1	$x_1 + 3x_2 \le 12$	$7x_1 + x_2 \le 34$	f(x) = $2x_1 + 2x_2$
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).

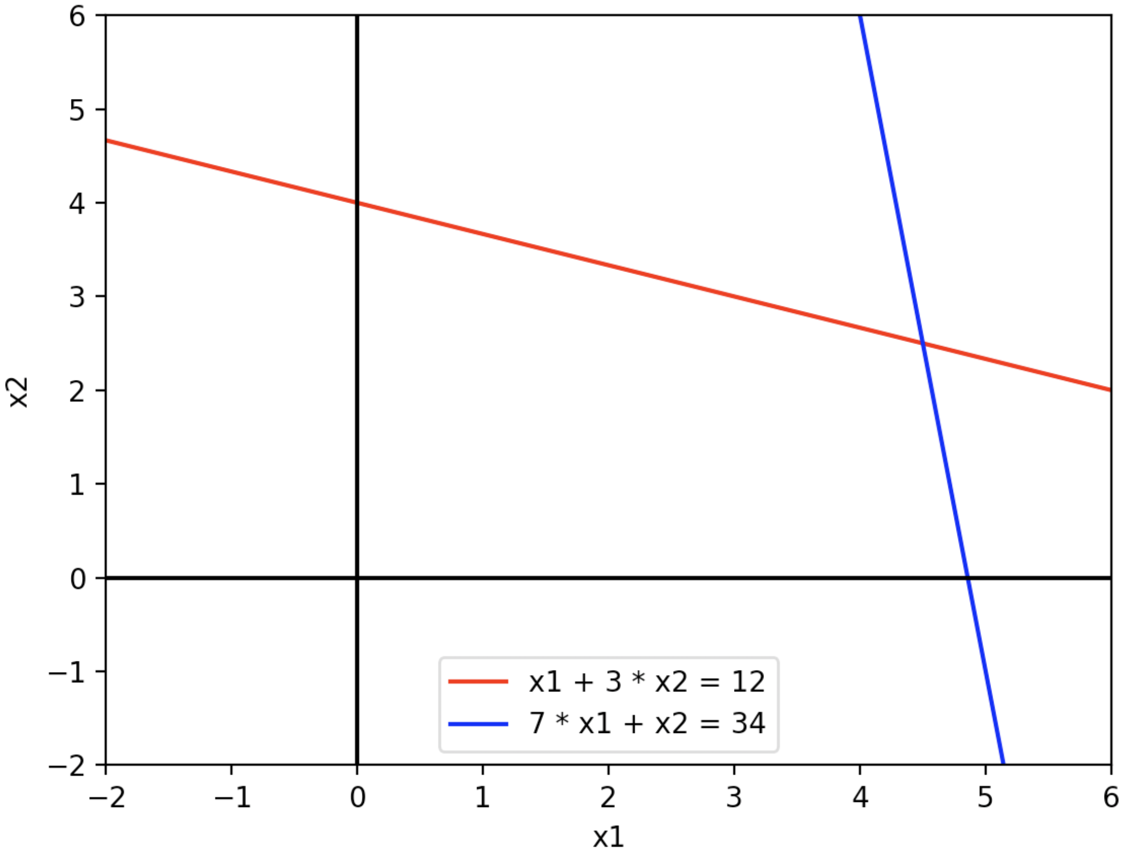

Рисунок 1.1 – Построение графиков по данным

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

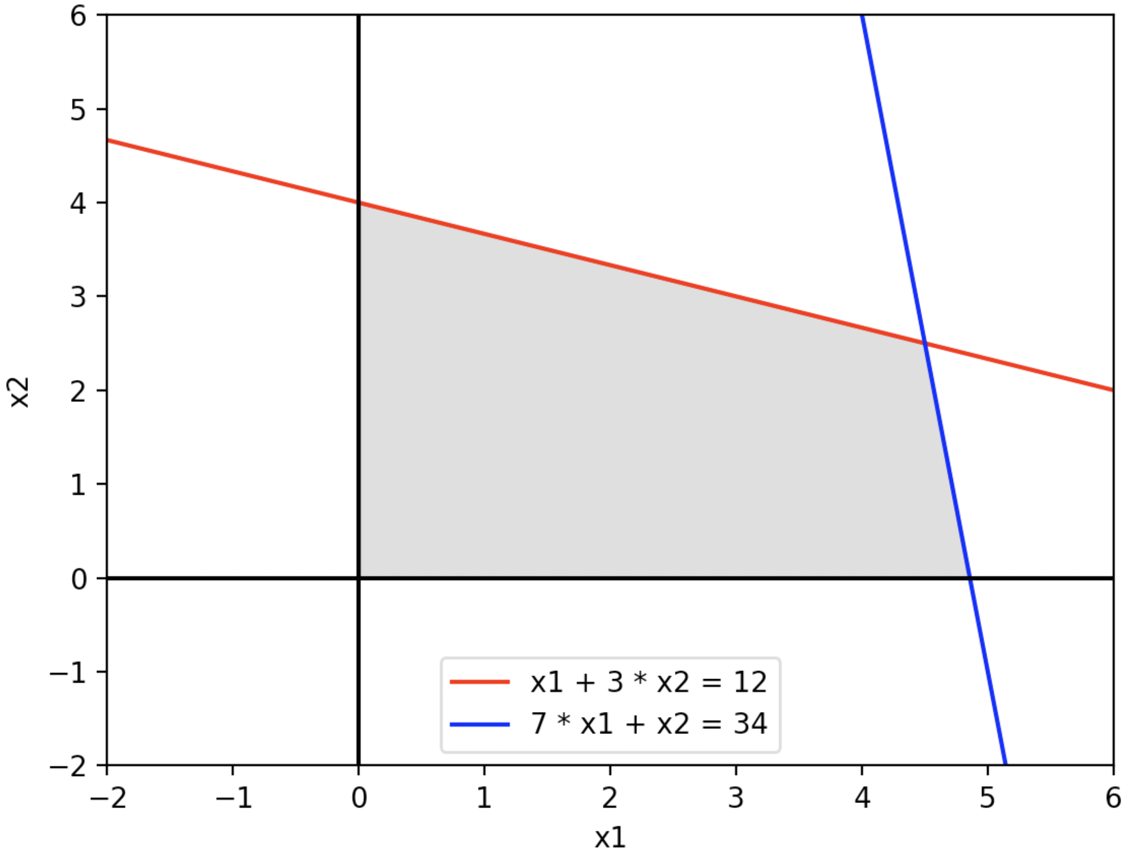

Рисунок 1.2 – Выделение области допустимых значений

1.6 Максимум функции

Для нахождения максимума функции найдем её градиент по формуле 1.1:

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

Для нахождения минимума функции найдем её градиент по формуле 1.1:

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

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}

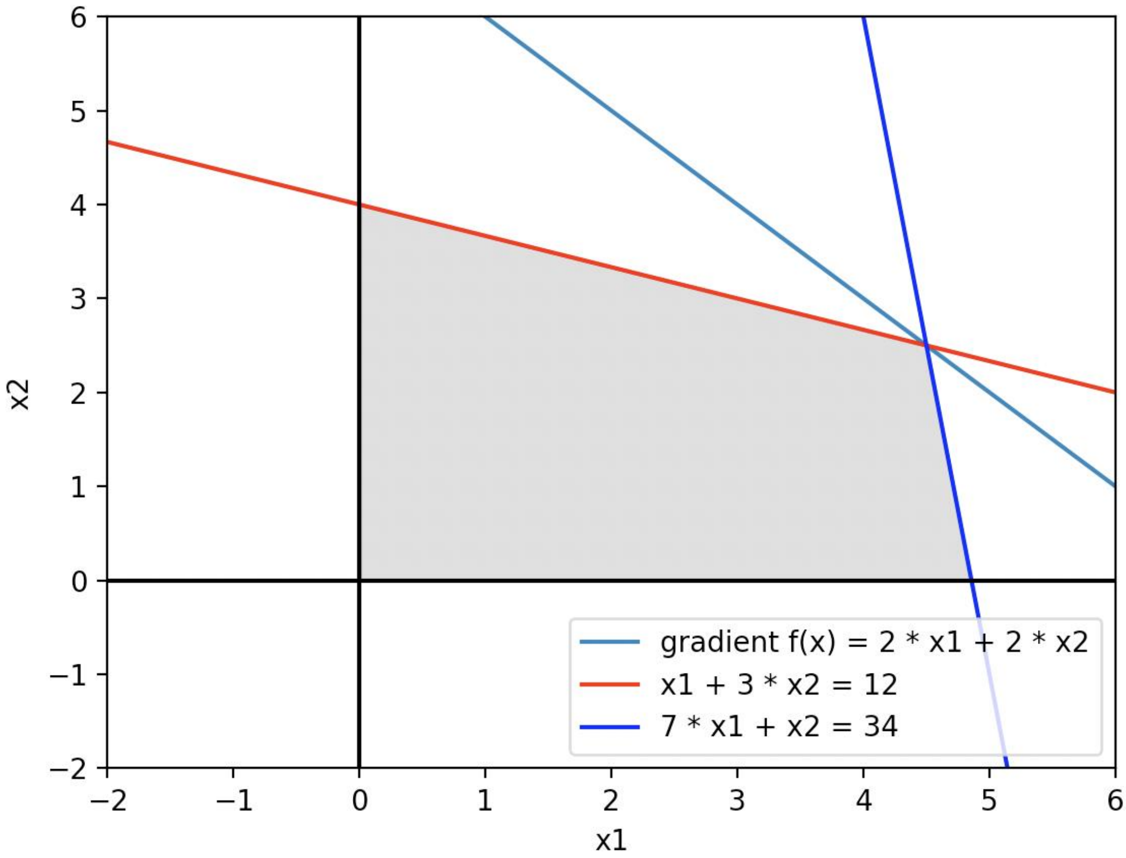

Рисунок 1.4 – Точка максимума функции

Найдем значение функции в точке максимума.

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

$$\begin{cases} 4{,}5 + 2 * 2{,}5 \leq 12 \\ 7 * 4{,}5 + 2{,}5 \leq 34 \\ x_1, x_2 \geq 0 \end{cases}$$

Получим значение равное F(x)max $= 2 * 4{,}5 + 2 * 2{,}5 = 14$

1.7 Минимум функции

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

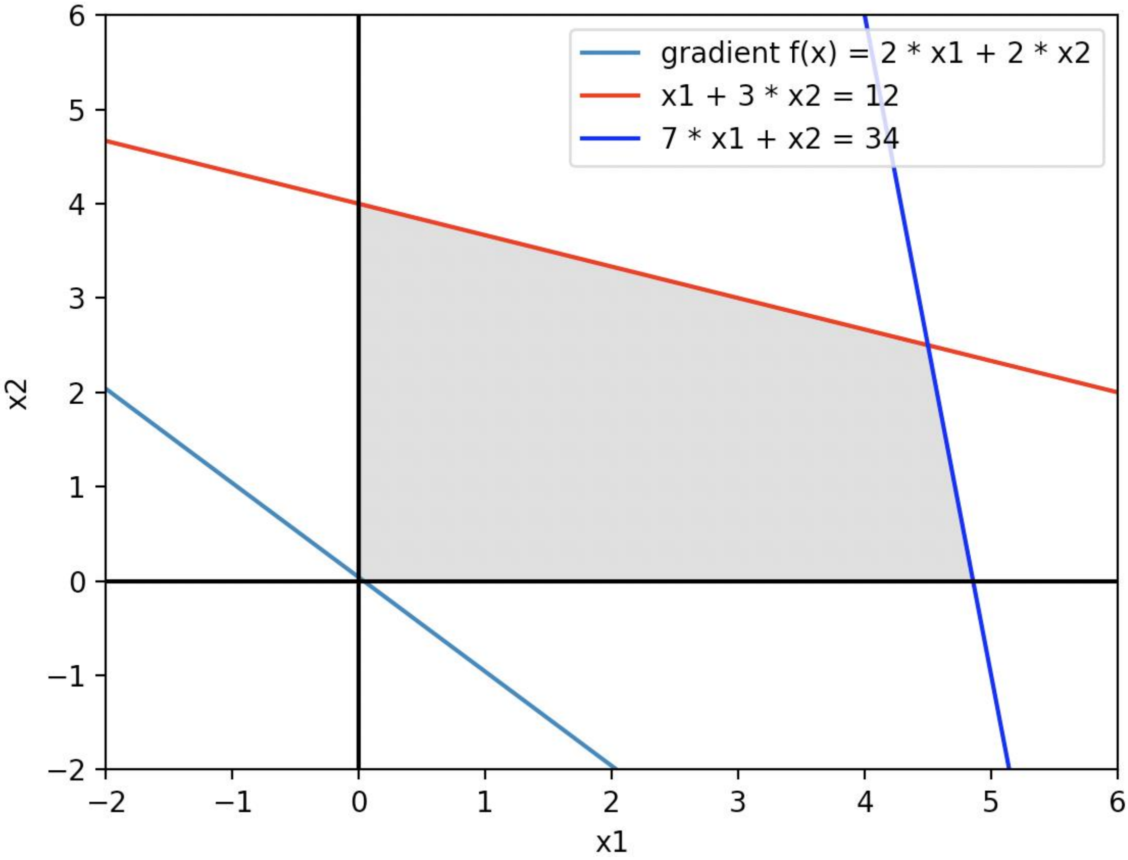

Рисунок 1.5 – Точка минимума функции

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:

$$\begin{cases} 2 + 0 + 2 * 0 \leq 12 \\ 7 * 0 + 0 \leq 34 \\ x_1, x_2 \geq 0 \end{cases}$$

Получим значение равное F(x)min $=2*0+2*0=0$

Ответ:

F(x)max = 14

F(x)min = 0

СИМПЛЕКСНЫЙ МЕТОД

1.1 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 x1 – количество выпускаемых бочек, x2 – количество выпускаемых ящиков. Прибыль от продажи сырья составит 1.5x1 + 1.2x2, прибыль требуется максимизировать.

Ограничения задачи:

$$\begin{cases} 0.15x1 + 0.2x2 \leq 60 \\ 0.2x1 + 0.1x2 \leq 40 \end{cases}$$

Таким образом, переходим к задаче линейного программирования:

$$f(x) = 1.5x1 + 1.2x2 \rightarrow max$$

$$\begin{cases} 0.15x1 \ + \ 0.2x2 \ \le \ 60 \\ 0.2x1 \ + \ 0.1x2 \ \le 40 \\ xi \ge 0, 1 \le i \le 2 \end{cases}$$

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

$$\begin{cases} 0.15x1 \ + \ 0.2x2 \ + \ x3 = \ 60 \\ 0.2x1 \ + \ 0.1x2 \ + \ x4 = 40 \\ xi \ge 0, 1 \le i \le 2 \end{cases}$$

$$f(x) = 1.5x1 + 1.2x2$$

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

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

$$\mathrm{A1} = \begin{pmatrix} 0.15 \\ 0.2 \end{pmatrix}, \mathrm{A2} = \begin{pmatrix} 0.2 \\ 0.1 \end{pmatrix}, \mathrm{A3} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \mathrm{A4} = \begin{pmatrix} 0 \\ 1 \end{pmatrix},$$

$$\mathrm{A0} = \begin{pmatrix} 60 \\ 40 \end{pmatrix}$$

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:

$$A3x3 + A4x4 = A0,$$

Которому соответствует первоначальный опорный план

$$x^{(0)} = (x1, x2, x3, x4) = (0{,}0{,}60{,}40),$$
$$f\left(x^{(0)}\right) = 0.$$

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

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

В левый столбец Таблицы 1.2 запишем переменные $x3$, $x4$ образующие базис, в верхней строке – небазисные переменные $x1$, $x2$. В строке c_j запишем коэффициенты целевой функции, соответствующие небазисным переменным c1 = 1.5, c2 = 1.2. 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.

$$\Delta_1 = (\overline{C_B} * \overline{A_1}) - c_1 = 0 * 0.15 + 0 * 0.2 - 1.5 = -1.5;$$

$$\Delta_2 = (\overline{C_B} * \overline{A_2}) - c_2 = 0 * 0.2 + 0 * 0.1 - 1.2 = -1.2;$$
$$\mathrm{Q} = (\overline{C_B} * \overline{A_0}) = 0 * 60 + 0 * 40 = 0.$$

Таблица 1.2 – Начальная симплекс-таблица задачи о максимальном доходе

	c_j	1.5	1.2	
$\overline{C_B}$		X1	X2	$\overline{A_0}$
0	X3	0.15	0.12	60
0	X4	0.2	0.1	40
	F	-1.5	-1.2	0
		Δ_1	Δ_2	Q

Таблица 1.3 – Заполнение f-строки

	c_j	1.5	1.2		
$\overline{C_B}$		X1	X2	$\overline{A_0}$	
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	
		Δ_1	Δ_2	Q	

Для оптимальности опорного решения в задаче на максимум требуется выполнение не отрицательности всех относительных оценок $\Delta i \geq 0$. Так как оценки $\Delta 1 = -1.5$, $\Delta 2 = -1.2$ в f-строке отрицательны, то это свидетельствуют о возможности улучшения полученного решения. Наибольшая по модулю отрицательная оценка $\Delta 1 = -1.5$. 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

	c_j	0	1.2	
$\overline{C_B}$		X4	X2	$\overline{A_0}$
0	X3			
1.5	X1	5		
	f			
		Δ_1	Δ_2	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

	c_j	7.5	3	
$\overline{C_B}$		X4	X2	$\overline{A_0}$
0	X3	-0.75		
1.5	X1	5	0.5	200
	F	7.5		
		Δ_1	Δ_2	Q

Таблица 1.6 – Итерация 0

	c_j	0	1.2		
$\overline{C_B}$		X4	X2	$\overline{A_0}$	
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».

$$\mathrm{a}_{12} = \frac{(0.2 * 0.2) - (0.15\ * 0.1)}{0.2} = 0.125;\ \mathrm{a}_{13} = \frac{(0.2 * 60) - (0.15 * 40)}{0.2} = 30;\ \mathrm{a}_{32} = \frac{(0.2 * -1.2) - (0.1 * -1.5)}{0.2} = -0{,}45;$$

$$\mathrm{a}_{33} = \frac{(0.15 * 0) - (60 * -1.5)}{0.2} = 300;$$

Базисное решение, которое дает последняя таблица

$$\mathrm{x}^{(1)} = (\mathrm{x}1, \mathrm{x}2, \mathrm{x}3, \mathrm{x}4) = (200, 0, 30{,}0),$$

$$f\left(x^{(1)}\right) = (\overline{C_B} * \overline{A_0}) = 0 * 30 + 1.5 * 200 = 300.$$

Это решение не является оптимальным, так как в f-строке имеются отрицательная оценка Δ2.

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

Так как оценка Δ2= -0.45 в f-строке отрицательна, то это свидетельствуют о возможности улучшения полученного решения. Наибольшая по модулю отрицательная оценка Δ1= −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

	c_j	0	0	
$\overline{C_B}$		X4	X3	$\overline{A_0}$
1.2	X2			
1.5	X1	5		
	F			
		Δ_1	Δ_2	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

	c_j	0	0	
$\overline{C_B}$		X4	X3	$\overline{A_0}$
1.2	X1	-6	8	240
1.5	X2		-4	
	F		3.6	
		Δ_1	Δ_2	Q

Таблица 1.9 – Итерация 1

	c_j	0	0	
$\overline{C_B}$		X4	X3	$\overline{A_0}$
1.2	X1	-6	8	240
1.5	X2	8	-4	80
	f	4.8	3.6	408
		Δ_1	Δ_2	Q

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

$$\mathrm{a}_{21} = \frac{(0.125 * 5) - (-0.75 * 0.5)}{0.125} = 8;\ \mathrm{a}_{23} = \frac{(0.125 * 200) - (0.5 * 30)}{0.125} = 80;\ \mathrm{a}_{31} = \frac{(0.125 * 7.5) - (-0.75 * -0.45)}{0.125} = 4.8;$$

$$\mathrm{a}_{33} = \frac{(0.125 * 300) - (-0.45 * 30)}{0.125} = 408;$$

Базисное решение, которое дает последняя таблица

$$\mathrm{x}^{(2)} = (\mathrm{x1}, \mathrm{x2}, \mathrm{x3}, \mathrm{x4}) = (240, 80, 0, 0),$$

$$f\left(x^{(2)}\right) = (\overline{C_B} * \overline{A_0}) = 1.2 * 240 + 80 * 1.5 = 408.$$

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

Подставляем базисное решение, которое дает последняя таблица

$$\mathrm{x}^* = \mathrm{x}^{(n)} = (\mathrm{x1}, \mathrm{x2}, \mathrm{x3}, \mathrm{x4}) = (240, 80, 0, 0)$$

Где n – количество итераций

$$f_{max} = f(x^*) = (\overline{C_B} * \overline{A_0}) = 408$$

Проверим решение по «правилу прямоугольника».

$f_{max} = Q = (\overline{C_B} * \overline{A_0})$ = ((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.

2. 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.

3. Зобнина О.В., Дю А.И., Бабаева Ю.А. Многокритериальная оптимизация// Научно-образовательный журнал для студентов и преподавателей «StudNet». – 2021. - №1.

4. Теория принятия решений: учебное пособие / С. М. Бородачёв. - Екатеринбург : Изд-во Урал, ун-та, 2014. - 124 с.

5. 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")
```