СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Деятельность принятия решений основана на целеполагании работы лица принимающего решения. Для экономистов такие работы предопределены реализацией производственно-хозяйственных предприятий, которые гарантируют наилучшее функционирование на основе связей между ними. Научные изучения – дают возможность отметить основные академические трудности, отыскать методы их исследования, предопределяют формирование опытной основы и теоретического аппарата. При реализации новой техники – оформляют значимую стадию в конструировании приборов, устройств, строений, в исследовании технологических процессов и их работоспособности. В социально-общественной области – применяются с целью управления функционирования и формирования общественных действий, их координации вместе с хозяйственными и финансовыми действиями. Подходящие (результативные) действия дают возможность доводить целеполагание при наименьших расходах трудящийся, вещественных и сырьевых ресурсов [1].

В традиционной математике способы поиска подходящих решений оценивают в сегментах, сопряженных вместе с исследованием экстремумов функций, в математическом программировании. Таким образом, математическое программирование считается одной из областей исследования операций практического тенденции кибернетики, применяемого с целью постановления фактических организационных вопросов. Способы математического точного программирования применяются с целью решения так называемых распределительных задач, какие появляются в случае, если существующих ресурсов никак недостаточно для исполнения любой из запланированных работ результативным способом и следует лучшим способом разделить средства согласно работам, в согласовании вместе с подобранным критерием оптимальности. Решение проблем математического программирования обретают использование в разных сферах человеческой деятельности, в каком месте нужен подбор одного из предложенных процессов деятельности.

1 МЕТОД ПАРЕТО

1.1 Введение

Целью данной практической работы является ознакомление с методом многокритериальной оптимизации Парето и изучение методов сужения оптимального множества альтернатив, включая метод верхних/нижних границ, метод субоптимизации и лексикографический метод оптимизации. Важные понятия в этой области включают: лицо, принимающее решение (ЛПР), альтернативы, критерии и исходы управленческих решений. Оптимальность по Парето определяется как состояние, при котором невозможно улучшить один критерий без ухудшения другого, формируя множество Парето из неулучшаемых или эффективных точек. Подходы к многокритериальной оптимизации включают выделение множества Парето и выбор наиболее предпочтительного варианта на основе дополнительных критериев или логического рассуждения.

1.2 Выбор Парето-оптимального множества

Поставить задачу и расписать ее: альтернативы, критерии и стремления, рассчитать множество Парето

Определение альтернатив: Каждый вариант завтрака является альтернативой для выбора.

Критерии оценки: Питательная ценность, калорийность на порцию, время приготовления, доступность и цена

Стремления: Цель состоит в том, чтобы найти завтрак, который имеет высокую питательную ценность и калорийность, низкое время приготовления, высокую доступность и низкую цену.

Выбор представлен в таблице 1.

Таблица 1.1 – Альтернативы и критерии

№	Варианты завтрака	Критерии				
		Калорийность на порцию (+)	Питательная ценность (1-5) (+)	Время приготовления (мин) (-)	Доступность (1-3) (+)	Цена (1-5) (-)
1	Овсянка с фруктами и орехами	3	5	25	3	2
2	Гречневая каша с овощами	3	3	20	3	2
3	Яичница с овощами	3	4	15	3	2
4	Смузи из зелени и фруктов	2	3	5	3	1
5	Творожная запеканка с ягодами	3	3	30	2	3
6	Тосты с авокадо и яйцом	4	4	10	2	4
7	Кускус с овощами и фетой	3	4	15	2	3
8	Йогурт с мюсли и свежими фруктами	2	3	4	3	2
9	Бутерброды с лососем	4	4	7	2	5
10	Бутерброд с сыром на козьем молоке	2	2	7	1	4

Таблица 1.2 – Сравнения альтернатив

	1	2	3	4	5	6	7	8	9	10
1	X	X	X	X	X	X	X	X	X	X
2	X	X	X	X	X	X	X	X	X	X
3	X	A3	X	X	X	X	X	X	X	X
4	н	н	Н	X	X	X	X	X	X	X
5	A1	A2	A3	н	X	X	X	X	X	X
6	н	н	н	н	н	X	X	X	X	X
7	н	н	A3	н	A7	н	X	X	X	X
8	н	н	н	н	н	н	н	X	X	X
9	н	н	н	н	н	н	н	н	X	X
10	н	н	н	A4	н	н	н	A8	н	X

Таблица 1.3 – Парето-оптимальное множество

№	Варианты завтрака	Критерии				
		Калорийность на порцию (+)	Питательная ценность (1-5) (+)	Время приготовления (мин) (-)	Доступность (1-3) (+)	Цена (1-5) (-)
1	Овсянка с фруктами и орехами	3	5	25	3	2
2	Гречневая каша с овощами	3	3	20	3	2
3	Яичница с овощами	3	4	15	3	2
4	Смузи из зелени и фруктов	2	3	5	3	1
7	Кускус с овощами и фетой	3	4	15	2	3
8	Йогурт с мюсли и свежими ф.	2	3	4	3	2

1.3 Указание верхних/нижних границ критериев.

Установим для приведенного примера верхнюю и нижнею границу. Калорийность порции не менее 3, питательная ценность не менее 4, время приготовления не более 15 минут

Таблица 1.4 - Результат верхних и нижних границ критериев

№	Варианты завтрака	Критерии				
		Калорийность на порцию (+)	Питательная ценность (1-5) (+)	Время приготовления (мин) (-)	Доступность (1-3) (+)	Цена (1-5) (-)
3	Яичница с овощами	3	4	15	3	2
6	Тосты с авокадо и яйцом	4	4	10	2	4
7	Кускус с овощами и фетой	3	4	15	2	3
9	Бутерброды с лососем	4	4	7	2	5

1.4 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.

Выберем главный критерий: **Время приготовления.**

Установим верхние/нижние границы для остальных критериев:

Калорийность не менее 3, питательная ценность не менее 4, цена не более 2.

Таблица 1.5 - Результат субоптимизации

№	Варианты завтрака	Критерии				
		Калорийность на порцию (+)	Питательная ценность (1-5) (+)	Время приготовления (мин) (-)	Доступность (1-3) (+)	Цена (1-5) (-)
3	Яичница с овощами	3	4	15	3	2
1	Овсянка с фруктами и орехами	3	5	25	3	2

1.5 Лексикографическая оптимизация

Упорядочим критерии по их относительной важности:

1) Время приготовления

2) Калорийность на порцию

3) Доступность

4) Цена

5) Питательная ценность

Таблица 1.6 - Результат лексикографического

№	Варианты завтрака	Критерии				
		Калорийность на порцию (+)	Питательная ценность (1-5) (+)	Время приготовления (мин) (-)	Доступность (1-3) (+)	Цена (1-5) (-)
8	Йогурт с мюсли и свежими фруктами	2	3	4	3	2

1.6 Результаты работы программы

Таблица Попарное сравнение альтернатив:

	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10
A1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
A2	н	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
A3	н	A3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
A4	н	н	н	0.0	0.0	0.0	0.0	0.0	0.0	0.0
A5	A1	A2	A3	н	0.0	0.0	0.0	0.0	0.0	0.0
A6	н	н	н	н	н	0.0	0.0	0.0	0.0	0.0
A7	н	н	A3	н	A7	н	0.0	0.0	0.0	0.0
A8	н	н	н	н	н	н	н	0.0	0.0	0.0
A9	н	н	н	н	н	н	н	н	0.0	0.0
A10	н	н	н	A4	н	н	н	A8	н	0.0

Рисунок 1.1 Таблица попарного сравнения альтернатив

Результат указания верхней/нижней границы: (Калорийность => 3 , Питательная ценность >= 4), Время приготовления <= 15

	Калорийность	Питательная ценность	Время приготовления	Доступность	Цена
A3	3	4	15	3	2
A6	4	4	10	2	4
A7	3	4	15	2	3

Рисунок 1.2 Таблица указания верхней и нижней границы

Результат отбора вариантов, удовлетворяющих заданным критериям: (главный критерий: Время приготовления, Питательная ценность >= 4 , Цена <= 2)

	Калорийность	Питательная ценность	Время приготовления	Доступность	Цена
A3	3	4	15	3	2
A1	3	5	25	3	2

Рисунок 1.3 Таблица субоптимизации

Результат лексикографической оптимизации: (Самая важная: Цена)

	Калорийность	Питательная ценность	Время приготовления	Доступность	Цена
A8	2	3	4	3	2

Рисунок 1.4 Таблица лексикографической оптимизации

1.7 Выводы по разделу

В ходе практической работы по мы изучили и использовали метод Парето для решения задачи выбора лучшего завтрака, учитывая разные критерии. Мы показали, что метод Парето помогает уменьшить количество вариантов, выбирая только те, которые оптимальны, что облегчает процесс выбора.

Также мы рассмотрели дополнительные методы для уточнения выбора, включая установление верхних и нижних границ для критериев, субоптимизацию и лексикографическую оптимизацию. Эти методы позволили нам более точно отфильтровать и упорядочить варианты, делая процесс выбора еще проще.

В итоге, мы нашли идеальный вариант завтрака, который соответствует всем нашим критериям и предпочтениям. Это подтверждает, что метод Парето и связанные с ним подходы к выбору из множества вариантов действительно работают эффективно.

2 МЕТОД ЭЛЕКТРА II

2.1 Введение

Основу методологии решающих правил основанных на порогах чувствительности составляют методы класса ЭЛЕКТРА, которые были разработана коллективом французских ученых, возглавляемым профессором Б. Руа. В настоящее время разработан ряд методов семейства ЭЛЕКТРА.

ЭЛЕКТРА I позволяет из множества вариантов исключить неэффективные варианты. В основе данного метода лежит попарное сравнение отдельных вариантов.

ЭЛЕКТРА II служит для упорядочения индифферентных классов вариантов.

ЭЛЕКТРА III отличается от метода ЭЛЕКТРА 2 способом задания порогов чувствительности.

В данных подходах принято различать 2 этапа: 1) этап разработки, на котором строятся один или несколько индексов попарного сравнения альтернатив; 2) этап исследования, на котором построенные индексы используются для ранжирования (или классификации) заданного множества альтернатив.

На первом этапе определяется множество решений и для каждого из N критериев определяется вес – число, характеризующее важность соответствующего критерия.

На втором этапе исследуется матрица и граф предпочтений для ранжирования альтернатив.

Увеличивая порог C, можно добиться уменьшения количества и устранения малозначащих связей, а также петель.

2.2 Выбор лучшего варианта

Составлена таблица критериев, по которым оцениваются проекты (Таблица 2.1).

Таблица 2.1 – Таблица критериев для оценки альтернатив

Критерии	Вес критерия	Шкала	Код	Стремление
Калорийность	4	До 100 100-200 от 200	5 10 15	max
Питательная ценность	3	Большая Средняя Маленькая	15 10 5	max
Время приготовления	5	До 5 мин 5-10 мин от 10 мин	15 10 5	min
Доступность	5	Большая Средняя Маленькая	15 10 5	max
Цена	2	До 200р 200-300р от 300р	15 10 5	min

Составлена таблица оценок выбора лучшего завтрака. Для 10-ти альтернатив заполнена Таблица 2.2

Таблица 2.2 – Таблица оценок по критериям

№	Варианты решений	Критерии				
		Калорийность	Питательная ценность	Время приготовления	Доступность	Цена
1	Овсянка с фруктами и орехами	10	10	15	15	10
2	Гречневая каша с овощами	10	10	15	10	5
3	Яичница с овощами	5	5	5	5	5
4	Смузи из зелени и фруктов	5	15	5	10	10
5	Творожная запеканка с ягодами	15	10	10	15	5
6	Тосты с авокадо и яйцом	15	10	10	10	10
7	Кускус с овощами и фетой	5	5	10	10	5

Продолжение Таблицы 2.2

8	Йогурт с мюсли и свежими фруктами	15	15	15	15	5
9	Бутерброды с лососем	15	10	5	5	5
10	Бутерброд с сыром на козьем молоке	10	10	15	10	10
Вес		4	3	5	5	2
Стремление		max	max	min	max	min

2.3 Веса предпочтений

Рассмотрим альтернативы 1 и 2

P = 5

N = 2

D = 2.5 - Принимаем

Рассмотрим альтернативы 1 и 3

P = 12

N = 7

D = 1.714 - Принимаем

Рассмотрим альтернативы 1 и 4

P = 9

N = 8

D = 1.125 - Принимаем

Рассмотрим альтернативы 1 и 5

P = 0

N = 11

D = 0.0 - Отбрасываем

Рассмотрим альтернативы 1 и 6

P = 5

N = 9

D = 0.556 - Отбрасываем

Рассмотрим альтернативы 1 и 7

P = 12

N = 7

D = 1.714 - Принимаем

Рассмотрим альтернативы 1 и 8

P = 0

N = 9

D = 0.0 - Отбрасываем

Рассмотрим альтернативы 1 и 9

P = 5

N = 7

D = 0.714 - Отбрасываем

Рассмотрим альтернативы 1 и 10

P = 5

N = 0

D = inf - Принимаем

Рассмотрим альтернативы 2 и 1

P = 2

N = 5

D = 0.4 - Отбрасываем

Рассмотрим альтернативы 2 и 3

P = 12

N = 5

D = 2.4 - Принимаем

Рассмотрим альтернативы 2 и 4

P = 6

N = 8

D = 0.75 - Отбрасываем

Рассмотрим альтернативы 2 и 5

P = 0

N = 14

D = 0.0 - Отбрасываем

Рассмотрим альтернативы 2 и 6

P = 2

N = 9

D = 0.222 - Отбрасываем

Рассмотрим альтернативы 2 и 7

P = 7

N = 5

D = 1.4 - Принимаем

Рассмотрим альтернативы 2 и 8

P = 0

N = 12

D = 0.0 - Отбрасываем

Рассмотрим альтернативы 2 и 9

P = 5

N = 5

D = 1.0 - Отбрасываем

Рассмотрим альтернативы 2 и 10

P = 2

N = 0

D = inf - Принимаем

Рассмотрим альтернативы 3 и 1

P = 7

N = 12

D = 0.583 - Отбрасываем

Рассмотрим альтернативы 3 и 2

P = 5

N = 12

D = 0.417 - Отбрасываем

Рассмотрим альтернативы 3 и 4

P = 2

N = 8

D = 0.25 - Отбрасываем

Рассмотрим альтернативы 3 и 5

P = 5

N = 12

D = 0.417 - Отбрасываем

Рассмотрим альтернативы 3 и 6

P = 7

N = 12

D = 0.583 - Отбрасываем

Рассмотрим альтернативы 3 и 7

P = 5

N = 5

D = 1.0 - Отбрасываем

Рассмотрим альтернативы 3 и 8

P = 5

N = 12

D = 0.417 - Отбрасываем

Рассмотрим альтернативы 3 и 9

P = 0

N = 7

D = 0.0 - Отбрасываем

Рассмотрим альтернативы 3 и 10

P = 7

N = 12

D = 0.583 - Отбрасываем

Рассмотрим альтернативы 4 и 1

P = 8

N = 9

D = 0.889 - Отбрасываем

Рассмотрим альтернативы 4 и 2

P = 8

N = 6

D = 1.333 - Принимаем

Рассмотрим альтернативы 4 и 3

P = 8

N = 2

D = 4.0 - Принимаем

Рассмотрим альтернативы 4 и 5

P = 8

N = 11

D = 0.727 - Отбрасываем

Рассмотрим альтернативы 4 и 6

P = 8

N = 4

D = 2.0 - Принимаем

Рассмотрим альтернативы 4 и 7

P = 8

N = 2

D = 4.0 - Принимаем

Рассмотрим альтернативы 4 и 8

P = 5

N = 11

D = 0.455 - Отбрасываем

Рассмотрим альтернативы 4 и 9

P = 8

N = 6

D = 1.333 - Принимаем

Рассмотрим альтернативы 4 и 10

P = 8

N = 4

D = 2.0 - Принимаем

Рассмотрим альтернативы 5 и 1

P = 11

N = 0

D = inf - Принимаем

Рассмотрим альтернативы 5 и 2

P = 14

N = 0

D = inf - Принимаем

Рассмотрим альтернативы 5 и 3

P = 12

N = 5

D = 2.4 - Принимаем

Рассмотрим альтернативы 5 и 4

P = 11

N = 8

D = 1.375 - Принимаем

Рассмотрим альтернативы 5 и 6

P = 7

N = 0

D = inf - Принимаем

Рассмотрим альтернативы 5 и 7

P = 12

N = 0

D = inf - Принимаем

Рассмотрим альтернативы 5 и 8

P = 5

N = 3

D = 1.667 - Принимаем

Рассмотрим альтернативы 5 и 9

P = 9

N = 5

D = 1.8 - Принимаем

Рассмотрим альтернативы 5 и 10

P = 16

N = 0

D = inf - Принимаем

Рассмотрим альтернативы 6 и 1

P = 9

N = 5

D = 1.8 - Принимаем

Рассмотрим альтернативы 6 и 2

P = 9

N = 2

D = 4.5 - Принимаем

Рассмотрим альтернативы 6 и 3

P = 12

N = 7

D = 1.714 - Принимаем

Рассмотрим альтернативы 6 и 4

P = 4

N = 8

D = 0.5 - Отбрасываем

Рассмотрим альтернативы 6 и 5

P = 0

N = 7

D = 0.0 - Отбрасываем

Рассмотрим альтернативы 6 и 7

P = 7

N = 2

D = 3.5 - Принимаем

Рассмотрим альтернативы 6 и 8

P = 5

N = 10

D = 0.5 - Отбрасываем

Рассмотрим альтернативы 6 и 9

P = 9

N = 7

D = 1.286 - Принимаем

Рассмотрим альтернативы 6 и 10

P = 9

N = 0

D = inf - Принимаем

Рассмотрим альтернативы 7 и 1

P = 7

N = 12

D = 0.583 - Отбрасываем

Рассмотрим альтернативы 7 и 2

P = 5

N = 7

D = 0.714 - Отбрасываем

Рассмотрим альтернативы 7 и 3

P = 5

N = 5

D = 1.0 - Отбрасываем

Рассмотрим альтернативы 7 и 4

P = 2

N = 8

D = 0.25 - Отбрасываем

Рассмотрим альтернативы 7 и 5

P = 0

N = 12

D = 0.0 - Отбрасываем

Рассмотрим альтернативы 7 и 6

P = 2

N = 7

D = 0.286 - Отбрасываем

Рассмотрим альтернативы 7 и 8

P = 5

N = 12

D = 0.417 - Отбрасываем

Рассмотрим альтернативы 7 и 9

P = 5

N = 12

D = 0.417 - Отбрасываем

Рассмотрим альтернативы 7 и 10

P = 7

N = 7

D = 1.0 - Отбрасываем

Рассмотрим альтернативы 8 и 1

P = 9

N = 0

D = inf - Принимаем

Рассмотрим альтернативы 8 и 2

P = 12

N = 0

D = inf - Принимаем

Рассмотрим альтернативы 8 и 3

P = 12

N = 5

D = 2.4 - Принимаем

Рассмотрим альтернативы 8 и 4

P = 11

N = 5

D = 2.2 - Принимаем

Рассмотрим альтернативы 8 и 5

P = 3

N = 5

D = 0.6 - Отбрасываем

Рассмотрим альтернативы 8 и 6

P = 10

N = 5

D = 2.0 - Принимаем

Рассмотрим альтернативы 8 и 7

P = 12

N = 5

D = 2.4 - Принимаем

Рассмотрим альтернативы 8 и 9

P = 12

N = 5

D = 2.4 - Принимаем

Рассмотрим альтернативы 8 и 10

P = 14

N = 0

D = inf - Принимаем

Рассмотрим альтернативы 9 и 1

P = 7

N = 5

D = 1.4 - Принимаем

Рассмотрим альтернативы 9 и 2

P = 5

N = 5

D = 1.0 - Отбрасываем

Рассмотрим альтернативы 9 и 3

P = 7

N = 0

D = inf - Принимаем

Рассмотрим альтернативы 9 и 4

P = 6

N = 8

D = 0.75 - Отбрасываем

Рассмотрим альтернативы 9 и 5

P = 5

N = 9

D = 0.556 - Отбрасываем

Рассмотрим альтернативы 9 и 6

P = 7

N = 9

D = 0.778 - Отбрасываем

Рассмотрим альтернативы 9 и 7

P = 12

N = 5

D = 2.4 - Принимаем

Рассмотрим альтернативы 9 и 8

P = 5

N = 12

D = 0.417 - Отбрасываем

Рассмотрим альтернативы 9 и 10

P = 7

N = 5

D = 1.4 - Принимаем

Рассмотрим альтернативы 10 и 1

P = 0

N = 5

D = 0.0 - Отбрасываем

Рассмотрим альтернативы 10 и 2

P = 0

N = 2

D = 0.0 - Отбрасываем

Рассмотрим альтернативы 10 и 3

P = 12

N = 7

D = 1.714 - Принимаем

Рассмотрим альтернативы 10 и 4

P = 4

N = 8

D = 0.5 - Отбрасываем

Рассмотрим альтернативы 10 и 5

P = 0

N = 16

D = 0.0 - Отбрасываем

Рассмотрим альтернативы 10 и 6

P = 0

N = 9

D = 0.0 - Отбрасываем

Рассмотрим альтернативы 10 и 7

P = 7

N = 7

D = 1.0 - Отбрасываем

Рассмотрим альтернативы 10 и 8

P = 0

N = 14

D = 0.0 - Отбрасываем

Рассмотрим альтернативы 10 и 9

P = 5

N = 7

D = 0.714 - Отбрасываем

Матрица предпочтений:

Составлена матрица предпочтений с внесенными и принятыми значениями D (Таблица 2.3).

Таблица 2.3 – Полная матрица предпочтений альтернатив.

	1	2	3	4	5	6	7	8	9	10
1	x	2.5	1.714	1.125	-	-	1.714	-	-	inf
2	-	x	2.4	-	-	-	1.4	-	-	inf
3	-	-	x	-	-	-	-	-	-	-
4	-	1.333	4	x	-	2	4	-	1.333	2
5	inf	inf	2.4	1.375	x	inf	inf	1.667	1.8	inf
6	1.8	4.5	1.714	-	-	x	3.5	-	1.286	inf
7	-	-	-	-	-	-	x	-	-	-
8	inf	inf	2.4	2.2	-	2	2.4	x	2.4	inf
9	1.4	-	inf	-	-	-	2.4	-	x	1.4
10	-	-	1.714	-	-	-	-	-	-	x

По матрице построен граф предпочтений (Рисунок 2.1).

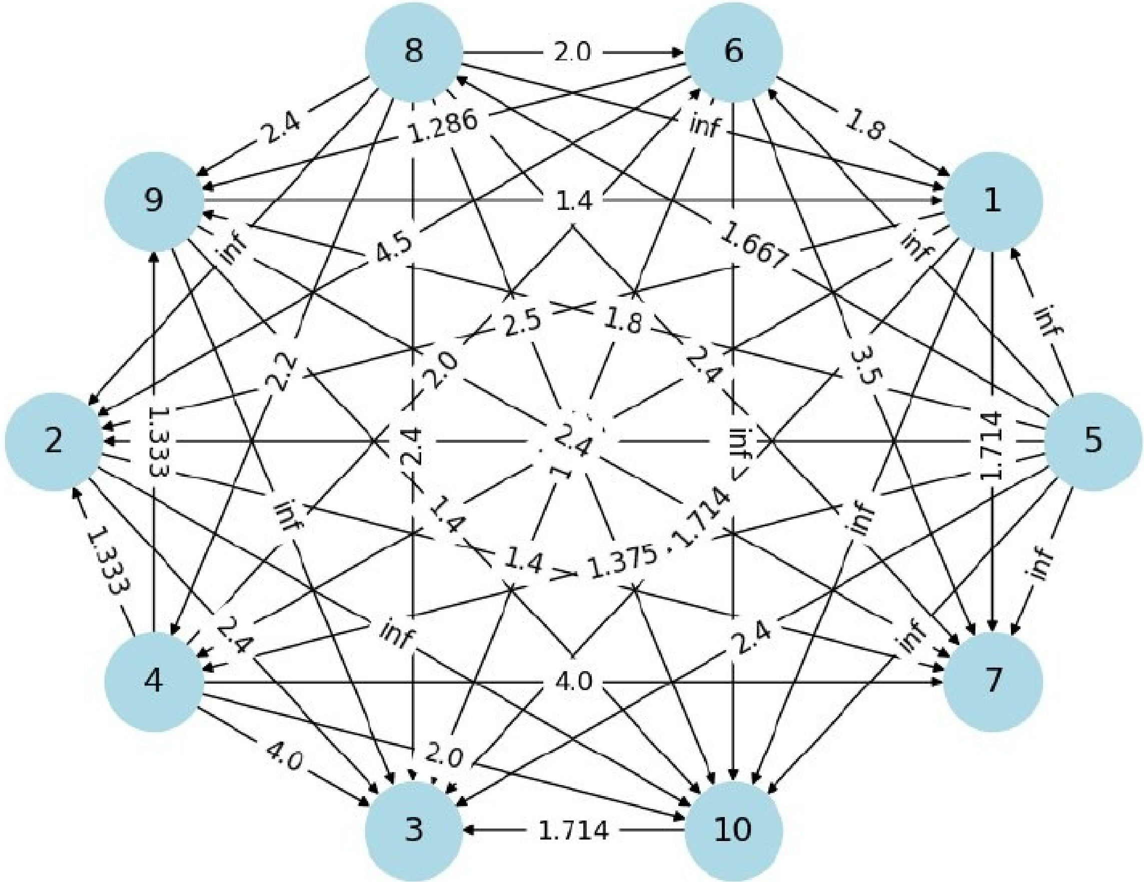


Рисунок 2.1 – Вид графа предпочтений

Назначен порог отбора предпочтений C = 2.3 (это соответствует тому, что учитываются только более сильные связи в графе).

Таким образом, матрица разрежается. В ней остаются только самые сильные связи (Таблица 2.4).

Таблица 2.4 – Матрица предпочтений проектов, при пороге C=2.3

	1	2	3	4	5	6	7	8	9	10
1	x	2.5	-	-	-	-	-	-	-	inf
2	-	x	2.4	-	-	-	-	-	-	inf
3	-	-	x	-	-	-	-	-	-	-
4	-	-	4	x	-	-	4	-	-	-
5	inf	inf	2.4	-	x	inf	inf	-	-	inf
6	-	4.5	-	-	-	x	3.5	-	-	inf
7	-	-	-	-	-	-	x	-	-	-

Продолжение Таблицы 2.4

8	inf	inf	2.4	-	-	-	2.4	x	2.4	inf
9	-	-	inf	-	-	-	2.4	-	x	-
10	-	-	-	-	-	-	-	-	-	x

По этой матрице построен граф предпочтений (Рисунок 2.2).

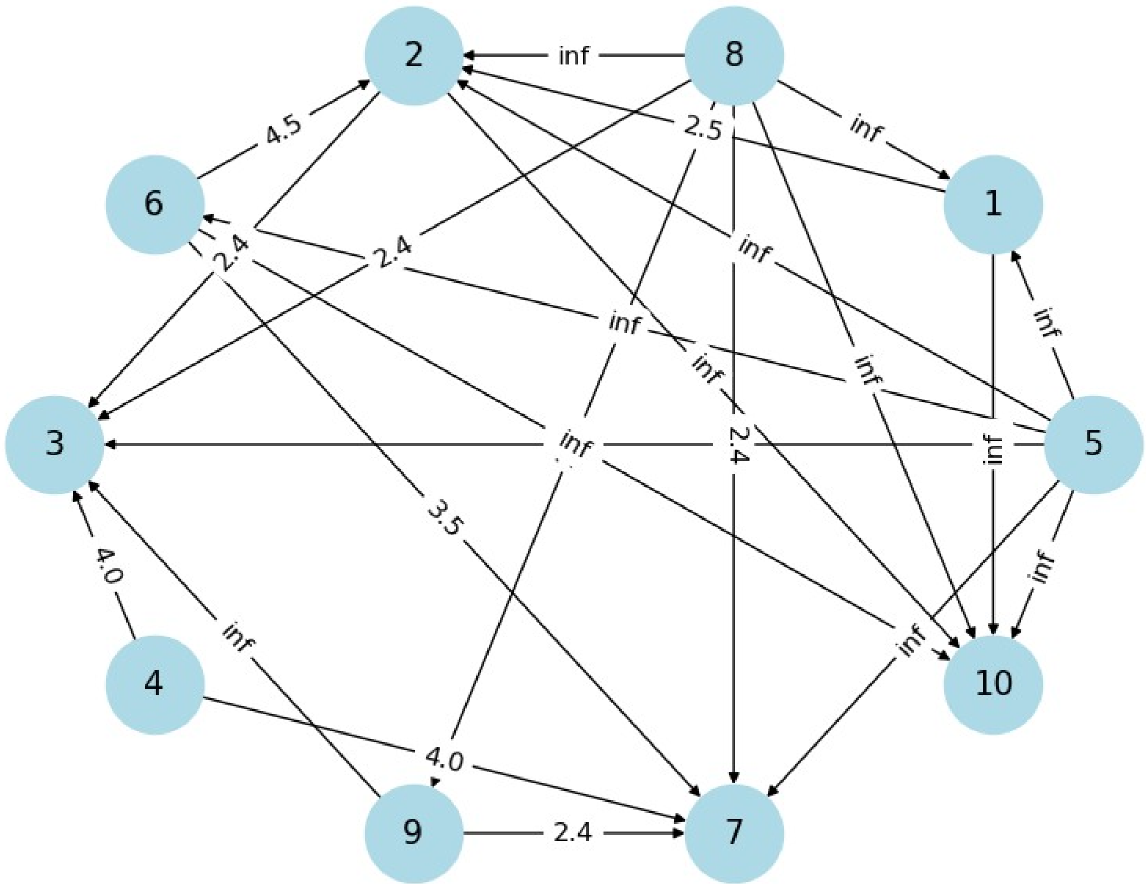


Рисунок 2.2 – Вид графа предпочтений для случая порога принятия решений С = 2.3

2.4 Вывод

Решение говорит нам о том, что лучшие завтраки - 5 и 8, а худший - 10.

2.5 Результат работы программы

	Овсянка с фруктами и орехами	Гречневая каша с овощами	Яичн
Овсянка с фруктами и орехами	0.0	2.500	
Гречневая каша с овощами	-1.0	0.000	
Яичница с овощами	-1.0	-1.000	
Смузи из зелени и фруктов	-1.0	1.333	
Творожная запеканка с ягодами	inf	inf	
Тосты с авокадо и яйцом	1.8	4.500	
Кускус с овощами и фетой	-1.0	-1.000	
Йогурт с мюсли и свежими фруктами	inf	inf	
Бутерброды с лососем	1.4	1.000	
Бутерброд с сыром на козьем молоке	-1.0	-1.000	

Рисунок 2.3 – Результат работы программы. Вывод матрицы предпочтений.

	Овсянка с фруктами и орехами	Гречневая каша с овощами	Яичница с овощами	См
сянка с фруктами и орехами	NaN	2.5	NaN	
ечневая каша с овощами	NaN	NaN	NaN	
чница с овощами	NaN	NaN	NaN	
узи из зелени и фруктов	NaN	NaN	4.0	
орожная запеканка с ягодами	inf	inf	NaN	
сты с авокадо и яйцом	NaN	4.5	NaN	
скус с овощами и фетой	NaN	NaN	NaN	
гурт с мюсли и свежими фруктами	inf	inf	NaN	
терброды с лососем	NaN	NaN	inf	
терброд с сыром на козьем молоке	NaN	NaN	NaN	

Рисунок 2.4 – Результат работы программы. Вывод матрицы после ограничения

2.6 Выводы по разделу

Был выбран оптимальный завтрак с учетом критериев и их весов, при помощи более лучшего метода Электра.

3 МЕТОД АНАЛИЗА ИЕРАРХИЙ

3.1 Введение

Метод Анализа Иерархий (Analitic Hierarchy Process) разработан американским математиком Томасом Саати в 70-х годах прошлого века. Метод анализа иерархий (МАИ) является замкнутой логической конструкцией, которая обеспечивает с помощью простых и хорошо обоснованных правил, решение задач МКО, включающих как качественные, так и количественные факторы, причем количественные факторы могут иметь разную размерность. Метод основан на декомпозиции задачи и представлении ее в виде иерархической структуры, что позволяет включить в иерархию все имеющиеся у лица, принимающего решение знания по решаемой проблеме и последующей обработке суждений. В результате может быть выявлена относительная степень взаимодействия элементов в иерархии, которые затем выражаются численно. МАИ включает процедуры синтеза множественных суждений, получения приоритетности критериев и нахождения альтернативных решений. Весь процесс решения подвергается проверке и переосмыслению на каждом этапе, что позволяет проводить оценку качества полученного решения.

Решение многокритериального выбора основано на трех основных этапах:

Первый этап – представление системы критериев (целей) в виде иерархической структуры.

Второй этап – оценки приоритетов (весов) критериев с учётом их места в иерархии относительной важности.

Третий этап – определение лучшей альтернативы по значениям её характеристик и важности критериев

3.2 Постановка задачи

Задача практической работы: выбрать оптимальный завтрак.

3.3 Представление проблемы в виде иерархии

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

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

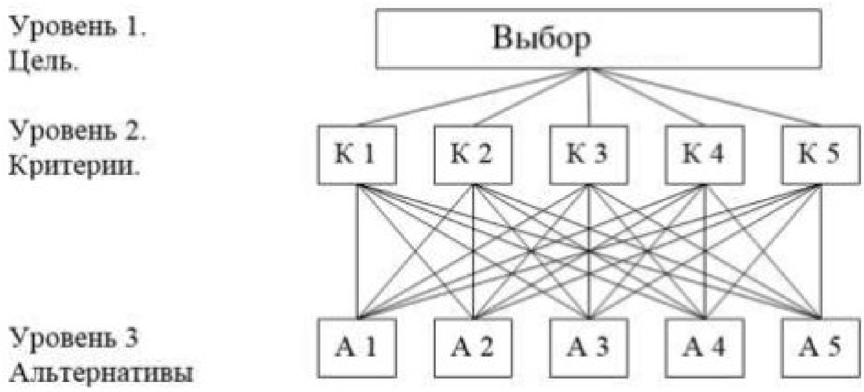


Рисунок 3.1 – Полная доминантная иерархия.

Критерии:

К 1 – Калорийность;

К 2 – Питательная ценность;

К 3 – Время приготовления;

К 4 – Доступность;

К 5 – Цена.

Альтернативы:

А 1 - Творожная запеканка с ягодами;

А 2 - Тосты с авокадо и яйцом;

А 3 - Кускус с овощами и фетой;

А 4 - Овсянка с фруктами и орехами;

А 5 - Яичница с овощами.

3.4 Установка приоритетов критериев

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

Таблица 3.1 – Шкала относительной важности.

Интенсивность относительной важности	Определение	Объяснение
1	Равная важность	Равный вклад двух критериев в цель.
3	Слабое превосходство	Дают легкое превосходство одной альтернативы над другой
5	Умеренное превосходство	Опыт и суждения дают умеренное превосходство
7	Сильное превосходство	Одному из критериев дается настолько сильное предпочтение.
9	Абсолютное превосходство	Очевидность превосходства одного критерия над другим

Продолжение Таблицы 3.1

2,4,6,8	Промежуточные решения между двумя соседними суждениями	Применяется в компромиссных случаях

Шкала содержит соответствующие обратные значения.

3.5 Синтез приоритетов

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

Таблица 3.2 – Матрица парного сравнения критериев.

Цель	К 1	К 2	К 3	К 4	К 5	Vi	W_{2i}
К 1	1	2	1/2	1/3	3	1	0.17
К 2	1/2	1	1/3	1/4	1	0.53	0.09
К 3	2	3	1	1/2	2	1.431	0.244
К 4	3	4	2	1	3	2.352	0.4
К 5	1/3	1	1/2	1/3	1	0.561	0.096
∑Vi						5.874	

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

Строка № 1

$V_1=(1x2x1/2x1/3x3)^{1/5} = 1;$

Строка № 2

$V_2=(1/2x1x1/3x1/4x1)^{1/5}=0.53$;

Строка № 3

$V_3=(2x3x1x1/2x2)^{1/5}=1.431$;

Строка № 4

$V_4=(3x4x2x1x3)^{1/5}=2.352$;

Строка № 5

$V_5=(1/3x1x1/2x1/3x1)^{1/5}=0.561$.

Проведена нормализация полученных чисел. Для этого определен **нормирующий коэффициент** $\sum Vi$.

$\sum Vi = V_1 + V_2 + V_3 + V_4 + V_5 = 1+0.53+1.431+2.352+ 0.561 =5.874$.

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

Строка № 1

$W_{21}= 1 / \sum Vi =Y_{21}$;

Строка № 2

$W_{22}= 0.53 / \sum Vi = Y_{22}$;

Строка № 3

$W_{23}= 1.431 / \sum Vi = Y_{23}$;

Строка № 4

$W_{24}= 2.352 / \sum Vi = Y_{24}$;

Строка № 5

$W_{25}= 0.561/ \sum Vi = Y_{25}$.

В результате получен вектор приоритетов:

W_{2i} = (0.17;0.09;0.244;0.4;0.096), где индекс 2 означает, что вектор приоритетов относится ко второму уровню иерархии.

К 1 – Калорийность (Таблица 3.3);

Таблица 3.3 – Матрица сравнения по критерию 1.

K1	A1	A2	A3	A4	A5	V_{K1Y}	W_{3K1Y}
A1	1	7	1/3	1/2	5	1.423	0.207
A2	1/7	1	1/7	1/6	1/2	0.279	0.041
A3	3	7	1	1	5	2.537	0.369
A4	2	6	1	1	4	2.169	0.316
A5	1/5	2	1/5	1/4	1	0.457	0.067
$\sum V_{K1Y}$						6.865	

Определена относительная ценность каждого элемента.

Строка № 1

V_{K11}= 1.423;

Строка № 2

V_{K12}= 0.279;

Строка № 3

V_{K13}=2.537;

Строка № 4

V_{K14}=2.169

Строка № 5

V_{K15}=0.457.

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

$\sum V_{K1Y} = V_{K11} + V_{K12} + V_{K13} + V_{K14} + V_{K15} = 6.865$

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

Строка № 1

W_{3K11}= 0.207

Строка № 2

W_{3K12}= 0.041;

Строка № 3

W_{3K13}= 0.369;

Строка № 4

W_{3K14}= 0.316;

Строка № 5

W_{3K15}= 0.067.

В результате получаем вектор приоритетов:

W_{3K1Y} = (0.207; 0.041; 0.369; 0.316; 0.067),

где индекс 3 означает, что вектор приоритетов относится к третьему уровню иерархии критерия К1.

К 2 – Питательная ценность (Таблица 3.4);

Таблица 3.4– Матрица сравнения по критерию 2.

K2	A1	A2	A3	A4	A5	V_{K2Y}	W_{3K2Y}
A1	1	3	1/2	1/4	1/3	0.66	0.106
A2	1/3	1	1/3	1/6	1/4	0.341	0.055
A3	2	3	1	1/2	1/3	1	0.16
A4	4	6	2	1	1/2	1.888	0.303
A5	3	4	3	2	1	2.352	0.377
$\sum V_{K2Y}$						6.241	

Определена относительная ценность каждого элемента.

Строка № 1

V_{K21}= 0.66;

Строка № 2

V_{K22}= 0.341

Строка № 3

V_{K23}=1;

Строка № 4

V_{K24}=1.888;

Строка № 5

V_{K25}=2.352.

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

$\sum V_{K2Y} = V_{K21} + V_{K22} + V_{K23} + V_{K24} + V_{K25} = 6.241$.

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

Строка № 1

W_{3K21}=0.106 ;

Строка № 2

W_{3K22}=0.055;

Строка № 3

W_{3K23}=0.16;

Строка № 4

W_{3K24}=0.303 ;

Строка № 5

W_{3K25}=0.377.

В результате получаем вектор приоритетов:

W_{3K2Y} = (0.106; 0.055;0.16;0.303;0.377),

где индекс 3 означает, что вектор приоритетов относится к третьему уровню иерархии критерия К2.

К 3 – Время приготовления (Таблица 3.5);

Таблица 3.5 – Матрица сравнения по критерию 3.

К3	А1	А2	А3	А4	А5	V_{K3Y}	W_{3K3Y}
А1	1	3	1/2	5	1/3	1.201	0.172
А2	1/3	1	1/4	2	1/5	0.506	0.072
А3	2	4	1	7	1/2	1.947	0.278
А4	1/5	1/2	1/7	1	1/9	0.276	0.039
А5	3	5	2	9	1	3.064	0.438
V_{K35}						6.994	

Определена относительная ценность каждого элемента.

Строка № 1

V_{K31}=1.201;

Строка № 2

V_{K32}=0.506;

Строка № 3

V_{K33}=1.947;

Строка № 4

V_{K34}=0.276;

Строка № 5

V_{K35}=3.064.

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

$\sum V_{K3Y} = V_{K31} + V_{K32} + V_{K33} + V_{K34} + V_{K35} = 6.994$.

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

Строка № 1

W_{3K31}=0.172;

Строка № 2

W_{3K32}=0.072;

Строка № 3

W_{3K33}=0.278;

Строка № 4

W_{3K34}=0.039;

Строка № 5

W_{3K35}= 0.438.

В результате получаем вектор приоритетов:

W_{3K3Y} = (0.172;0.072;0.278;0.039;0.438),

где индекс 3 означает, что вектор приоритетов относится к третьему уровню иерархии критерия К3.

К 4 – Доступность (Таблица 3.6);

Таблица 3.6– Матрица сравнения по критерию 4.

К4	А1	А2	А3	А4	А5	V_{K4Y}	W_{3K4Y}
А1	1	5	1/6	1/7	5	0.901	0.108
А2	1/5	1	1/8	1/8	3	0.393	0.047
А3	6	8	1	1	9	3.366	0.402
А4	7	8	1	1	9	3.471	0.415
А5	1/5	1/3	1/9	1/9	1	0.242	0.029
$\sum Vi$						8.373	

Определена относительная ценность каждого элемента.

Строка № 1

V_{K41}=0.901;

Строка № 2

V_{K42}=0.393;

Строка № 3

V_{K43}=3.366;

Строка № 4

V_{K44}= 3.471;

Строка № 5

V_{K45}=0.242.

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

$\sum V_{K4Y} = V_{K41} + V_{K42} + V_{K43} + V_{K44} + V_{K45} = 8.373$.

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

Строка № 1

W_{3K41}=0.108;

Строка № 2

W_{3K42}= 0.047;

Строка № 3

W_{3K43}= 0.402;

Строка № 4

W_{3K44}= 0.415;

Строка № 5

W_{3K45}= 0.029

В результате получаем вектор приоритетов:

W_{3K4Y} = (0.108; 0.047; 0.402; 0.415; 0.029)

где индекс 3 означает, что вектор приоритетов относится к третьему уровню иерархии критерия К4.

К 5 – Цена (Таблица 3.7).

Таблица 3.7– Матрица сравнения по критерию 5.

К5	А1	А2	А3	А4	А5	V_{K5Y}	W_{3K5Y}
А1	1	2	5	1/3	4	1.679	0.243
А2	1/2	1	3	1/4	1/2	0.715	0.103
А3	1/5	1/3	1	1/7	1/6	0.276	0.04
А4	3	4	7	1	5	3.347	0.484
А5	1/4	2	6	1/5	1	0.903	0.13
$\sum V_{K5Y}$						6.92	

Определена относительная ценность каждого элемента.

Строка № 1

V_{K51}= 1.679;

Строка № 2

V_{K52}=0.715;

Строка № 3

V_{K53}=0.276;

Строка № 4

V_{K54}=3.347;

Строка № 5

V_{K55}=0.903.

Проведена нормализация полученных чисел. Для этого определен нормирующий коэффициент $\sum V_{K5Y}$.

$\sum V_{K5Y} = V_{K51} + V_{K52} + V_{K53} + V_{K54} + V_{K55} = 6.92$.

Найдена важность приоритетов W_{3K5Y}, для этого каждое из чисел V_{K5Y} разделено на $\sum V_{K5Y}$.

Строка № 1

$W_{3K51}=0.243$;

Строка № 2

$W_{3K52}=0.103$;

Строка № 3

$W_{3K53}= 0.04$;

Строка № 4

$W_{3K54}= 0.484$;

Строка № 5

$W_{3K55}= 0.13$.

В результате получаем вектор приоритетов:

$W_{3K5Y} = (0.243;0.103;0.04;0.484;0.13)$,

где индекс 3 означает, что вектор приоритетов относится к третьему уровню иерархии критерия К5.

3.6 Согласованность локальных приоритетов

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

Метод анализа иерархий дает возможность провести такую оценку.

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

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

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

Определены индекс согласованности и отношение согласованности для матрицы «Выбор оптимального завтрака» (Таблица 3.8).

Таблица 3.8– Матрица «Выбор лучшего завтрака».

Цель	К 1	К 2	К 3	К 4	К 5	W_{2i}
К 1	1	2	1/2	1/3	3	0.17
К 2	1/2	1	1/3	1/4	1	0.09
К 3	2	3	1	1/2	2	0.244
К 4	3	4	2	1	3	0.4
К 5	1/3	1	1/2	1/3	1	0.096

Определена сумма каждого столбца матрицы суждений.

S_1 =6.833;

S_2=11;

S_3 =4.333;

S_4 =2.417;

S_5 =10.

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

$P_1 = S_1 \times W_{21} = 1.163$;

$P_2 = S_2 \times W_{22} = 0.992$;

$P_3 = S_3 \times W_{23} = 1.056$;

$P_4 = S_4 \times W_{24} = 0.968$;

$P_5 = S_5 \times W_{25} = 0.955$.

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.

$$\lambda_{max} = P_1 + P_2 + P_3 + P_4 + P_5 = 5.134.$$

Отклонение от согласованности выражается индексом согласованности.

$$ИС = (\lambda_{max} - n)/(n - 1) = 0.033.$$

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

$$ОС = ИС/СИ = 0.03$$

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

Определнены индекс согласованности и отношение согласованности для матрицы К 1 – Калорийность (Таблица 3.9).

Таблица 3.9 – Матрица сравнения по критерию 1.

К1	А1	А2	А3	А4	А5	W_{3K1Y}
А1	1	7	1/3	1/2	5	0.207
А2	1/7	1	1/7	1/6	1/2	0.041
А3	3	7	1	1	5	0.369
А4	2	6	1	1	4	0.316
А5	1/5	2	1/5	1/4	1	0.067

Определяется сумма каждого столбца матрицы суждений.

$S_{1K1} = 6.343$;

$S_{2\,K1} = 23$;

$S_{3\,K1} = 2.676$;

$S_{4\,K1} = 2.917$;

$S_{5\,K1} = 15.5$.

Затем полученный результат умножен на компоненту нормализованного вектора приоритетов.

$P_{1\,K1} = S_1 \times W_{3K11} = 1.315$;

$P_{2\,K1} = S_2 \times W_{3K12} = 0.936$;

$P_{3\,K1} = S_3 \times W_{3K13} = 0.989$;

$P_{4\,K1} = S_1 \times W_{3K14} = 0.921$;

$P_{5\,K1} = S_1 \times W_{3K15} = 1.033$.

Найдена пропорциональность предпочтений.

$$\lambda_{max\ K1} = P_{1K1} + P_{2K1} + P_{3K1} + P_{4K1} + P_{5K1} = 5.193.$$

Отклонение от согласованности выражается индексом согласованности.

$$ИС_{K1} = (\lambda_{max\ K1} - n)/(n - 1) = 0.048.$$

Найдено отношение согласованности ОС.

$$ОС_{K1} = ИС/СИ = 0.043.$$

Значение ОС меньше или равное 0.10 считается приемлемым, значит матрица К 1 (цена дня проживания) согласована.

Определены индекс согласованности и отношение согласованности для матрицы К 2 – Питательная ценность (Таблица 3.10).

Таблица 3.10 – Матрица сравнения по критерию 2.

К2	А1	А2	А3	А4	А5	W_{3K2Y}
А1	1	3	1/2	1/4	1/3	0.106
А2	1/3	1	1/3	1/6	1/4	0.055
А3	2	3	1	1/2	1/3	0.16
А4	4	6	2	1	1/2	0.303
А5	3	4	3	2	1	0.377

Определена сумма каждого столбца матрицы суждений.

$S_{1K2} = 10.333$;

$S_{2\,K2} = 17$;

$S_{3\,K2} = 6.833$;

$S_{4\,K2} = 3.917$;

$S_{5\,K2} = 2.417$.

Затем полученный результат умножен на компоненту нормализованного вектора приоритетов.

$P_{1\,K2} = S_1 \times W_{3\,K21}$ =1.092;

$P_{2\,K2} = S_2 \times W_{3\,K22}$ =0.93;

$P_{3\,K2} = S_3 \times W_{3\,K23}$ =1.095;

$P_{4\,K2} = S_4 \times W_{3\,K24}$ =1.185;

$P_{5\,K2} = S_5 \times W_{3\,K25}$ =0.911.

Найдена пропорциональность предпочтений.

$$\lambda_{max\,K2} = P_{1K2} + P_{2K2} + P_{3K2} + P_{4K2} + P_{5K2} = 5.212.$$

Отклонение от согласованности выражается индексом согласованности.

$$ИС_{K2} = (\lambda_{max\,K2} - n)/(n - 1) = 0.053.$$

Найдено отношение согласованности ОС.

$$ОС_{K2} = ИС/СИ = 0.047.$$

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

Определены индекс согласованности и отношение согласованности для матрицы К 3 – Время приготовления (Таблица 3.11).

Таблица 3.11 – Матрица сравнения по критерию 3.

К3	A1	A2	A3	A4	A5	W_{3K3Y}
A1	1	3	1/2	5	1/3	0.172
A2	1/3	1	1/4	2	1/5	0.072
A3	2	4	1	7	1/2	0.278
A4	1/5	1/2	1/7	1	1/9	0.039
A5	3	5	2	9	1	0.438

Определена сумма каждого столбца матрицы суждений.

S_{1K3} =6.533;

$S_{2\,K3}$ =13.5;

$S_{3\,K3}$ =3.893;

$S_{4\,K3}$ =24;

$S_{5\,К3}$ =2.144.

Затем полученный результат умножен на компоненту нормализованного вектора приоритетов.

$P_{1\,К3} = S_1 \times W_{3\,К31}$ =1.122;

$P_{2\,К3} = S_2 \times W_{3\,К32}$ =0.978;

$P_{3\,К3} = S_3 \times W_{3\,К33}$ =1.084;

$P_{4\,К3} = S_4 \times W_{3\,К34}$ =0.945;

$P_{5\,К3} = S_5 \times W_{3\,К35}$ =0.939.

Найдем пропорциональность предпочтений.

$$\lambda_{max\,К3} = P_{1К3} + P_{2К3} + P_{3К3} + P_{4К3} + P_{5К3} = 5.068.$$

Отклонение от согласованности выражается индексом согласованности.

$$ИС_{К3} = (\lambda_{max\,К3} - n)/(n - 1) = 0.017.$$

Найдено отношение согласованности ОС.

$$ОС_{К3} = ИС/СИ = 0.015.$$

Значение ОС меньше или равное 0.10 считается приемлемым, значит матрица К 3 (Время приготовления) согласована.

Определены индекс согласованности и отношение согласованности для матрицы К 4 – Доступность (Таблица 3.12).

Таблица 3.12 – Матрица сравнения по критерию 4.

К4	А1	А2	А3	А4	А5	$W_{3К4Y}$
А1	1	5	1/6	1/7	5	0.108
А2	1/5	1	1/8	1/8	3	0.047
А3	6	8	1	1	9	0.402
А4	7	8	1	1	9	0.415
А5	1/5	1/3	1/9	1/9	1	0.029

Определена сумма каждого столбца матрицы суждений.

$S_{1К4}$ =14.4;

$S_{2К4}$ =22.333;

$S_{3К4}$ =2.403;

S_{4K4} =2.379;

S_{5K4} =27.

Затем полученный результат умножен на компоненту нормализованного вектора приоритетов.

$P_{1K4} = S_1 \times W_{3\,K41}$ =1.55;

$P_{2K4} = S_2 \times W_{3\,K42}$ =1.048 ;

$P_{3K4} = S_3 \times W_{3\,K43}$ =0.966;

$P_{4K4} = S_4 \times W_{3\,K44}$ =0.986;

$P_{5K4} = S_5 \times W_{3\,K45}$ =0.779.

Найдена пропорциональность предпочтений.

$$\lambda_{max\,K4} = P_{1K4} + P_{2K4} + P_{3K4} + P_{4K4} + P_{5K4} = 5.33.$$

Отклонение от согласованности выражается индексом согласованности.

$$ИС_{K4} = (\lambda_{max\,K4} - n)/(n - 1) = 0.082.$$

Найдено отношение согласованности ОС.

$$ОС_{K4} = ИС/СИ = 0.074.$$

Значение ОС меньше или равное 0,10 считается приемлемым, значит матрица К 4 (Доступность) согласована.

Определены индекс согласованности и отношение согласованности для матрицы К 5 – Цена (Таблица 3.13).

Таблица 3.13 – Матрица сравнения по критерию 5.

К5	А1	А2	А3	А4	А5	W_{3K5Y}
А1	1	2	5	1/3	4	0.243
А2	1/2	1	3	1/4	1/2	0.103
А3	1/5	1/3	1	1/7	1/6	0.04
А4	3	4	7	1	5	0.484
А5	1/4	2	6	1/5	1	0.13

Определена сумма каждого столбца матрицы суждений.

S_{1K5} =4.95;

S_{2K5} =9.333;

S_{3K5} =22;

S_{4K5} =1.926;

S_{5K5} =10.667.

Затем полученный результат умножен на компоненту нормализованного вектора приоритетов.

$P_{1K5} = S_1 \text{ x } W_{3\ K41}$ =1.201;

$P_{2K5} = S_2 \text{ x } W_{3\ K42}$ =0.965;

$P_{3K5} = S_3 \text{ x } W_{3\ K43}$ =0.876;

$P_{4K5} = S_1 \text{ x } W_{3\ K44}$ =0.932;

$P_{5K5} = S_1 \text{ x } W_{3\ K45}$ =1.392.

Найдена пропорциональность предпочтений.

$$\lambda_{max\ K5} = P_{1K5} + P_{2K5} + P_{3K5} + P_{4K5} + P_{5K5} = 5.365.$$

Отклонение от согласованности выражается индексом согласованности.

$$ИС_{K5} = (\lambda_{max\ K5} - n)/(n - 1) = 0.091.$$

Найдено отношение согласованности ОС.

$$ОС_{K5} = ИС/СИ = 0.082.$$

Значение ОС меньше или равное 0,10 считается приемлемым, значит матрица К 5 (Цена) согласована.

3.7 Синтез альтернатив

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

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

W_{2i} = (0.17, 0.09, 0.244, 0.4, 0.096);

W_{3K1Y} = (0.207, 0.041, 0.369, 0.316, 0.067);

W_{3K2Y} = (0.106, 0.055, 0.16 , 0.303, 0.377);

W_{3K3Y} = (0.172, 0.072, 0.278, 0.039, 0.438);

W_{3K4Y} = (0.108 0.047 0.402 0.415 0.029);

W_{3K5Y} = (0.243, 0.103, 0.04 , 0.484, 0.13).

Приоритеты альтернатив получены следующим образом:

$W_1 = W_{21} \times W_{3K11} + W_{22} \times W_{3K21} + W_{23} \times W_{3K31} + W_{24} \times W_{3K41} + W_{25} \times W_{3K51}$ =.

$W_2 = W_{21} \times W_{3K12} + W_{22} \times W_{3K22} + W_{23} \times W_{3K32} + W_{24} \times W_{3K42} + W_{25} \times W_{3K52}$ =.

$W_3 = W_{21} \times W_{3K13} + W_{22} \times W_{3K23} + W_{23} \times W_{3K33} + W_{24} \times W_{3K43} + W_{25} \times W_{3K53} = W_4 = W_{21} \times W_{3K14} + W_{22} \times W_{3K24} + W_{23} \times W_{3K34} + W_{24} \times W_{3K44} + W_{25} \times W_{3K54}$ =.

$W_5 = W_{21} \times W_{3K15} + W_{22} \times W_{3K25} + W_{23} \times W_{3K35} + W_{24} \times W_{3K45} + W_{25} \times W_{3K55}$ =.

Таким образом, приоритеты альтернатив равны:

1. альтернатива А1 - W_1 приоритет равен =0.262;
2. альтернатива А2 - W_2 приоритет равен =0.219;
3. альтернатива А3 - W_3 приоритет равен =0.161;
4. альтернатива А4 - W_4 приоритет равен =0.289;
5. альтернатива А5 - W_5 приоритет равен =0.266

3.8 Вывод

Наиболее оптимальным выбором по методу МАИ получается А4 с приоритетом 0.289.

3.9 Результаты работы программы

Результаты работы программы, реализующей метод анализа иерархий, приведены на Рисунке 3.2.

```
Результат:
A1: 0.262
A2: 0.219
A3: 0.161
A4: 0.289
A5: 0.266
```

Рисунок 3.2 – Вывод программы

3.10 Выводы по разделу

Сначала были выбраны альтернативы и критерии, установлены приоритеты критериев. Далее был произведен синтез приоритетов для цели и для критериев, затем проверил согласованность локальных приоритетов и в конце был произведен синтез альтернатив и выбор лучшей/лучших.

МАИ хорош тем, что позволяет качественно и подробно сравнить объекты по выбранным критериям с установленными приоритетами.

Основной минус в трудоемкости процесса.

4 ГРАФИЧЕСКИЙ МЕТОД

4.1 Введение

Линейное программирование - это способ поиска оптимального решения задачи, где целью является линейная функция, а условия задачи ограничены системой линейных равенств и неравенств.

В рамках линейного программирования существуют различные классы задач, для которых разработаны специальные методы решения, отличающиеся от общих методов. Один из таких классов задач - транспортные задачи, которые возникли как отдельное направление в линейном программировании.

Если целью задачи является поиск экстремума линейных функций, то это задача линейного программирования. Если хотя бы одна из функций не является линейной, то это уже задача нелинейного программирования.

Нелинейное программирование - это способ решения задач, где как целевая функция, так и условия ограничений задачи являются нелинейными. Задача линейного программирования -состоит в нахождении минимума (или максимума) линейной функции при линейных ограничениях.

Какие задачи решают при помощи методов линейного программирования:

- задача об оптимальном использовании ресурсов при производственном планировании;
- задача о смесях (планирование состава продукции);
- задача о нахождении оптимальной комбинации различных видов продукции для хранения на складах (управление товарно-материальными запасами или "задача о рюкзаке");
- транспортные задачи (анализ размещения предприятия, перемещение грузов.

4.2 Постановка задачи

Решить задачу линейного программирования с двумя переменными графическим методом.

4.3 Данные индивидуального варианта

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

$$\begin{cases} 4x_1 - x_2 \ge -4 \\ 2x_1 + 3x_2 \le 12 \\ 5x_1 - 3x_2 \le 15 \\ x_1, x_2 \ge 0 \end{cases}$$

4.4 Подготовка данных

В среде Microsoft Excel добавим 4 столбца:

1. x_1 – значения от 0 до 5 с шагом 0,5;
2. $x_2 = 4 + 4x_1$ – значения ограничения $(4x_1 - x_2 \ge -4)$;
3. $x_2 = \frac{12 - 2x_1}{3}$ – значения ограничения $(2x_1 + 3x_2 \le 12)$;
4. $x_2 = \frac{5x_1 - 15}{3}$ – значения ограничения $(5x_1 - 3x_2 \le 15)$;
5. $x_2 = 2x_1$ – значения *целевой функции при условии* $f(x) = 0$.

Таблица 4.1 – Данные для графика

x1	$x_2 = 4 + 4x_1$	$x_2 = \frac{12 - 2x_1}{3}$	$x_2 = \frac{5x_1 - 15}{3}$	$x_2 = 2x_1$
0	4	4	-5	0
0,5	6	3,67	-4,17	1
1	8	3,33	-3,33	2
1,5	10	3	-2,5	3
2	12	2,67	-1,67	4

Продолжение Таблицы 4.1

2,5	14	2,33	-0,83	5
3	16	2	0	6
3,5	18	1,67	0,83	7
4	20	1,33	1,67	8
4,5	22	1	2,5	9
5	24	0,67	3,33	10

4.5 Построение графика

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

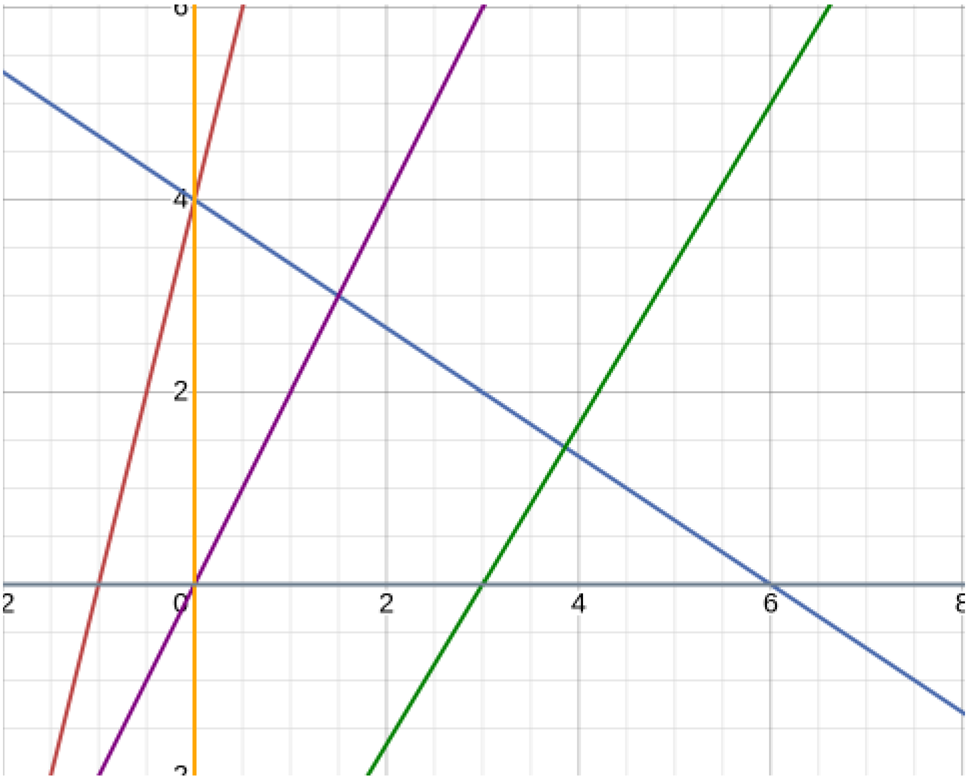


Рисунок 4.1 – Построение графиков по данным

4.6 Выделение области допустимых решений

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

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

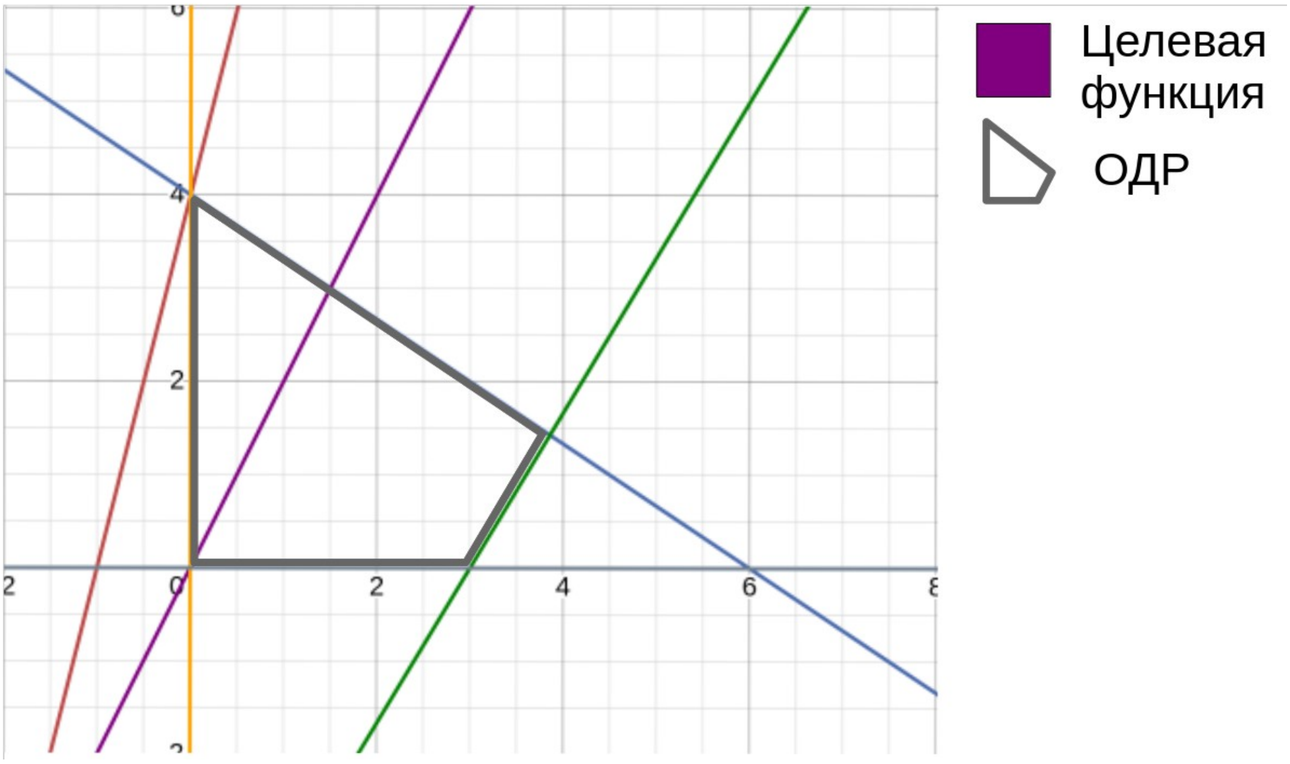


Рисунок 4.2 – Выделение области допустимых решений

4.7 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, 1}, а антиградиент функции будет равен {2, -1}. Изобразим эти вектора на графике (Рисунок 4.3).

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

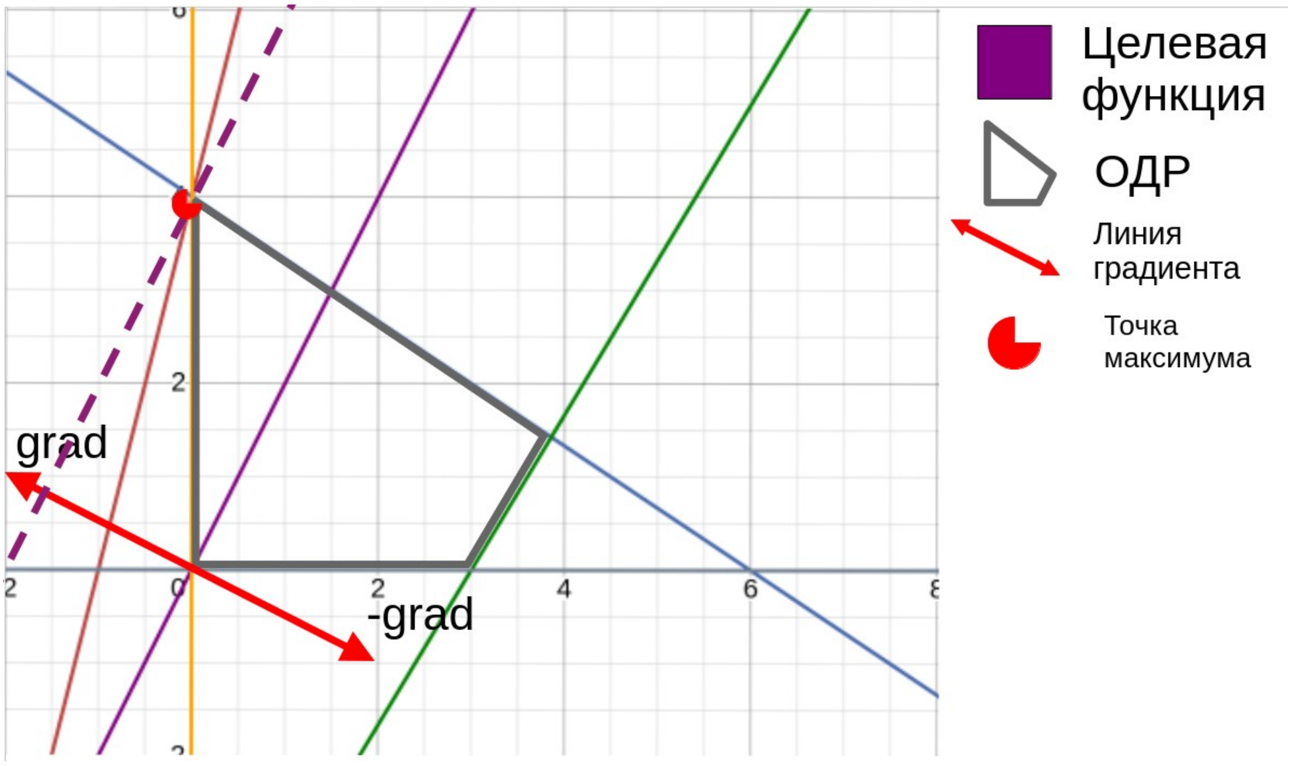


Рисунок 4.3 – Точка максимума функции

В результате получим точку с координатами (0, 4). Найдем значение функции в этой точке.

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

$$\begin{cases} -4 \geq -4 \\ 12 \leq 12 \\ -12 \leq 15 \\ 0,4 \geq 0 \end{cases}$$

Получим значение равное F(x)max = 4.

4.8 Минимум функции

Для нахождения минимума функции будем перемещать прямую в сторону антиградиента. Отметим на графике найденную точку (Рисунок 4.4).

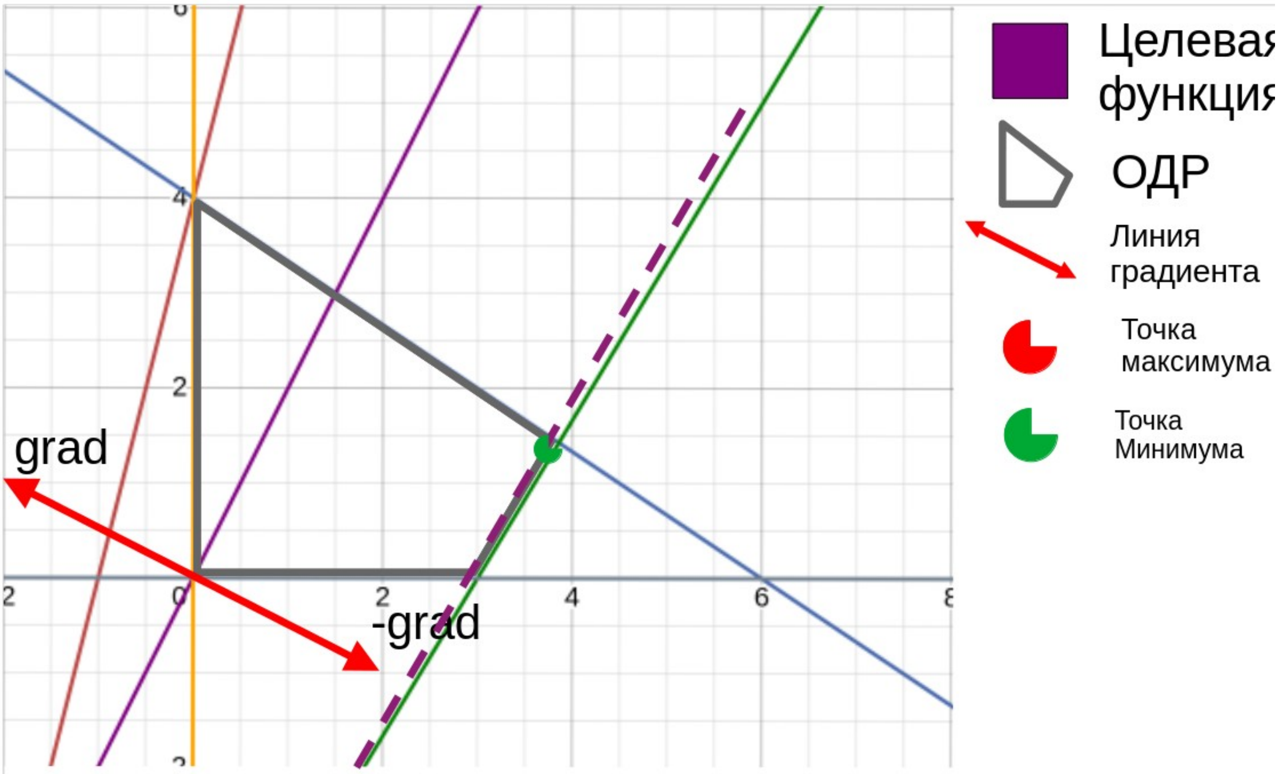


Рисунок 4.4 – Точка минимума функции

Получаем точку на пересечении двух прямых, найдем ее при помощи системы.

В результате получим точку с координатами (27/7, 10/7). Найдем значение функции в этой точке.

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

$$\begin{cases} 14 \ge -4 \\ 12 \le 12 \\ 15 \le 15 \\ \frac{27}{10}, \frac{10}{7} \ge 0 \end{cases}$$

Получим результат F(x)min = -6,28.

Ответ:

F(x)max = 4.

F(x)min = -6,28

4.9 Выводы по разделу

В excel была оформлена матрица. При помощи сайта Mathway и paint проводилась визуализации, выделение ОДР, градиента и антиградиента, точек минимума и максимума, также проверил что точки принадлежат системе уравнений.

5 СИМПЛЕКСНЫЙ МЕТОД

5.1 Введение

Симплексный метод — это алгоритм для решения задач линейного программирования, которые включают в себя максимизацию или минимизацию линейной функции при ограничениях в виде линейных неравенств. Этот метод был разработан Джорджем Данцигом в 1947 году и является одним из наиболее широко используемых методов оптимизации.

Принцип работы симплексного метода начинается с математической модели. Задача линейного программирования формулируется в виде целевой функции и системы ограничений. Например: максимизировать целевую функцию при ограничениях, заданных системой линейных неравенств.

Преимущества симплексного метода включают гарантированное нахождение оптимального решения, если оно существует, и возможность применения к большим задачам благодаря эффективным алгоритмам. Ограничения метода включают возможные проблемы с вырожденными решениями, что может привести к увеличению числа итераций, и сложности с задачами, содержащими большое количество переменных и ограничений без предварительной обработки данных.

5.2 Постановка задачи

Таблица 5.1 Исходные данные задачи.

Ресурсы	Нормы расхода сырья на ед. продукции, ед.	Запасы ресурсов, ед.	Ресурсы	Нормы расхода сырья на ед. продукции, ед.	Запасы ресурсов, ед.
	A	B	C	D	

Продолжение Таблицы 5.1

I	2	1	0,5	4	3400
II	1	5	3	0	1200
III	3	0	6	1	3000
Прибыль от единицы продукции, ден. ед.			7,5	3	6

5.3 Математическая модель задачи

Пусть x1 – тип ткани I, x2 –тип ткани II, x3 –тип ткани III. Прибыль от продажи тканей составит 8x1 + 7x2 + 6x3, прибыль требуется максимизировать.

Ограничения задачи:

$$\begin{cases} 2x1+3x2+4x3 \leq 700 \\ x1+4x2+5x3 \leq 800 \\ 3x1+4x2+2x3 \leq 600 \end{cases}$$

Таким образом, переходим к задаче линейного программирования:

$$f(x)=8x1+7x2+6x3 \rightarrow max$$

$$\begin{cases} 2x1+3x2+4x3 \leq 700 \\ x1+4x2+5x3 \leq 800 \\ 3x1+4x2+2x3 \leq 600 \\ xi \geq 0, 1 \leq i \leq 3 \end{cases}$$

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

$$\begin{cases} 2x1+3x2+4x3+x4 \le 700 \\ x1+4x2+5x3+x5 \le 800 \\ 3x1+4x2+2x3+x6 \le 600 \\ xi \ge 0,\ 1 \le i \le 6 \end{cases}$$

$$f(x)=8x1+7x2+6x3+0x4+0x5+0x6$$

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

$$A1x1+A2x2+A3x3+A4x4+A5x5+A6x6=A0,$$

$$A1=\begin{pmatrix}2\\1\\3\end{pmatrix}, A2=\begin{pmatrix}3\\4\\4\end{pmatrix}, A3=\begin{pmatrix}4\\5\\2\end{pmatrix}, A4=\begin{pmatrix}1\\0\\0\end{pmatrix}, A5=\begin{pmatrix}0\\1\\0\end{pmatrix}, A6=\begin{pmatrix}0\\0\\1\end{pmatrix},$$

$$A0=\begin{pmatrix}700\\800\\600\end{pmatrix}$$

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

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

Для этого свободные переменные $x1$, $x2$, $x3$ приравниваем нулю. В результате получим разложение

$$A4x4+A5x5+A6x6=A0,$$

Которому соответствует первоначальный опорный план

$$x^{(0)}=(x1,x2,x3,x4,x5,x6)=(0,0,0,700,800,600),$$

$$f(x^{(0)})=0.$$

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

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

В левый столбец Таблицы 5.2 запишем переменные $x4$, $x5$, $x6$ образующие базис, в верхней строке – небазисные переменные $x1$, $x2$, $x3$. В строке c_j запишем коэффициенты целевой функции, соответствующие небазисным переменным c1 = 8, c2 = 7, c3 = 6. 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 $\overline{A_2}$. И столбец, определяемый переменной $x3$ состоит из коэффициентов вектора $\overline{A_3}$ Крайний правый столбец заполняется элементами столбца $\overline{A_0}$, в нем же в результате вычислений получаем оптимальный план.

Заполнение f-строки (Таблица 5.3). Найдем относительные оценки Δ1, Δ2, Δ3 и значение целевой функции Q.

$$\Delta_1=(\overline{C_B}*\overline{A_1})-c_1=0*2+0*1+0*3-8=-8;$$

$$\Delta_2=(\overline{C_B}*\overline{A_2})-c_2=0*3+0*4+0*4-7=-7$$

$$\Delta_3=(\overline{C_B}*\overline{A_3})-c_3=0*4+0*5+0*2-6=-6;$$

$$Q=(\overline{C_B}*\overline{A_0})=0*700+0*800+0*600=0.$$

Таблица 5.2 – Начальная симплекс-таблица задачи о максимальном доходе

	c_j	8	7	6	
$\overline{C_B}$		X1	X2	X3	$\overline{A_0}$
0	X4	2	3	4	700
0	X5	1	4	5	800
0	X6	3	4	2	600
	f				
		Δ_1	Δ_2	Δ_3	Q

Таблица 5.3 – Заполнение f-строки

	c_j	8	7	6		
$\overline{C_B}$		X1	X2	X3	$\overline{A_0}$	
0	X4	2	3	4	700	700/2=350
0	X5	1	4	5	800	800/1=800
0	X6	**3**	4	2	600	600/3=200(min)
	f	-8	-7	-6	0	
		Δ_1	Δ_2	Δ_3	Q	

Для оптимальности опорного решения в задаче на максимум требуется выполнение неотрицательности всех относительных оценок $\Delta i \geq 0$. Так как оценки $\Delta 1 = -8$, $\Delta 2 = -7$ и $\Delta 3 = -6$ в f-строке отрицательны, то это свидетельствуют о возможности улучшения полученного решения. Наибольшая по модулю отрицательная оценка $\Delta 1 = -8$. 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 $x6$. Эта переменная исключается из базиса. В Таблице 5.3 разрешающий столбец и разрешающая строка выделены. Разрешающим элементом является число $a31 = 3$.

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

Таблица 5.4 – Новая симплекс-таблица

	c_j	0	7	6	
$\overline{C_B}$		X6	X2	X3	$\overline{A_0}$
0	X4				
0	X5				
8	X1	1/3			
	f				
		Δ_1	Δ_2	Δ_3	Q

В Таблице 1.4 переменные $x1$ и $x6$ меняются местами вместе с коэффициентами cj. Разрешающий элемент заменяется на обратный. В Таблице 5.5 элементы разрешающей строки делятся на разрешающий элемент. Элементы разрешающего столбца делятся на разрешающий элемент и меняют знак.

Таблица 5.5 – Симплекс преобразования

	c_j	0	7	6	
$\overline{C_B}$		X6	X2	X3	$\overline{A_0}$
0	X4	-2/3			
0	X5	-1/3			
8	X1	1/3	4/3	2/3	200
	f	8/3			
		Δ_1	Δ_2	Δ_3	Q

Таблица 5.6 – Итерация 0

	c_j	0	7	6		
$\overline{C_B}$		X6	X2	X3	$\overline{A_0}$	
0	X4	-2/3	1/3	**8/3**	300	900/8=112(min)
0	X5	-1/3	8/3	13/3	600	*1800/13 = 138*
8	X1	1/3	4/3	2/3	200	*600/2 =300*
	f	8/3	1	-2/3	1600	
		Δ_1	Δ_2	Δ_3	Q	

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

$$a_{12}=\frac{(3*3)-(2*4)}{3}=1/3;a_{13}=\frac{(4*3)-(2*2)}{3}=8/3;$$

$$a_{14}=\frac{(700*3)-(2*600)}{3}=300;a_{22}=\frac{(4*3)-(4*1)}{3}=8/3;$$

$$a_{23}=\frac{(5*3)-(1*2)}{3}=13/3;a_{24}=\frac{(800*3)-(1*600)}{3}=600;$$

$$\Delta_2=\frac{(-7*3)-(-8*4)}{3}=1;$$

$$\Delta_3=\frac{(-6*3)-(-8*2)}{3}=-2/3;$$

$$Q=\frac{(0*3)-(-8*600)}{3}=1600;$$

Базисное решение, которое дает последняя таблица

$$x^{(1)}=(x1,x2,x3,x4,x5,x6)=(200,0,0,300,600,0),$$

$$f(x^{(1)})=(\overline{C_B}*\overline{A_0})=8*200+0*600+0*300=1600.$$

Это решение не является оптимальным, так как в f-строке имеются отрицательные оценки Δ1, Δ2.

Выполняем следующую итерацию до тех пор пока в таблице f-строка не будет отрицательных оценок Δ1, Δ2, Δ3.

Таблица 5.7 – Итерация 1

	c_j	0	7	0	
$\overline{C_B}$		X6	X2	X4	$\overline{A_0}$
6	X3	-1/4	1/8	3/8	900/8
0	X5	3/4	17/8	-13/8	900/8
8	X1	1/2	10/8	-1/4	1000/8
	f	5/2	13/12	1/4	1675
		Δ_1	Δ_2	Δ_3	Q

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

Подставляем базисное решение, которое дает последняя таблица

$$x^{(4)}=(x1,x2,x3,x4,x5,x6)=(1000/8,0,900/8,0,900/8,0),$$

Где n – количество итераций

$$f(x^{(4)})=(\overline{C_B}*\overline{A_0})=6*900/8+8*1000/8=1675.$$

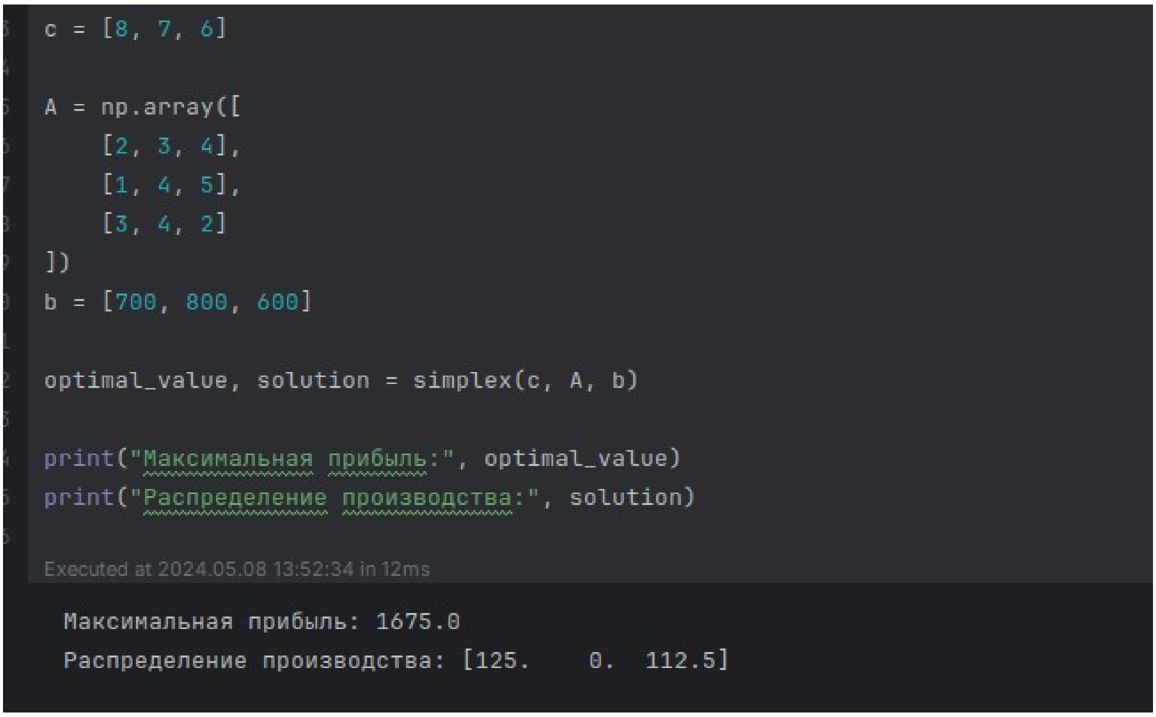
Проверим решение по «правилу прямоугольника».

$$f_{max}=Q=(\overline{C_B}*\overline{A_0})=\frac{(1600*8/3)-((-2/3)*300)}{8/3}=1675.$$

Таким образом, фабрика должна выпускать в течении недели $x1$ = 1000/8 шт. тканей типа I и $x3$= 900/8 шт. тканей типа III. Тогда фабрика получит максимальный доход от продажи 1675 [ден.ед].

5.5 Пример работы программы

Результаты выполнения программы, реализующей симплексный метод, представлены на Рисунке 5.1.



```
c = [8, 7, 6]

A = np.array([
    [2, 3, 4],
    [1, 4, 5],
    [3, 4, 2]
])
b = [700, 800, 600]

optimal_value, solution = simplex(c, A, b)

print("Максимальная прибыль:", optimal_value)
print("Распределение производства:", solution)

Executed at 2024.05.08 13:52:34 in 12ms

  Максимальная прибыль: 1675.0
  Распределение производства: [125.    0.  112.5]
```

Рисунок 5.1 – Результат работы программы

5.6 Выводы по разделу

Алгоритм симплекс метода заключается в составлении новых таблиц с помощью процедуры повторяющихся расчетов итераций до тех пор, пока не будет получено оптимальное решение задачи, либо сделан вывод о неразрешимости поставленной задачи.

Задача была переведена в каноническую форму линейного программирования с добавлением дополнительных переменных для преобразования неравенств в равенства.

С использованием начальной симплекс-таблицы было найдено первое базисное допустимое решение, которое затем подверглось проверке на оптимальность. На основе относительных оценок и процедуры повторяющихся расчетов в рамках симплекс-метода было продемонстрировано, как происходит переход к новому базисному решению, улучшая значение целевой функции.

После нескольких итераций, было достигнуто оптимальное решение задачи, гарантирующее максимальную прибыль при данных ограничениях.

ЗАКЛЮЧЕНИЕ

В ходе выполнения данной курсовой работы были рассмотрены три метода многокритериальной оптимизации: метод Парето и его оптимизационные варианты, метод Электра II и метод анализа иерархий.

Метод Парето является наиболее простым, но имеет значительный недостаток: он редко приводит к единственному оптимальному решению. Для устранения этого недостатка разработаны различные методы оптимизации, такие как метод указания верхних/нижних границ критериев, метод субоптимизации и лексикографическая оптимизация. Однако, и они часто приводят к множеству оптимальных решений или к слишком субъективным результатам.

Метод Электра II менее субъективен по сравнению с методом Парето, но его реализация сложнее, особенно в программном обеспечении. Кроме того, этот метод также может приводить к нескольким решениям, и для устранения этого недостатка необходимо экспериментально подбирать значение порога.

Метод анализа иерархий обеспечивает получение единственного оптимального решения, что является его преимуществом. Однако его основной недостаток заключается в высокой субъективности, так как приоритеты критериев устанавливаются ЛПР вручную. Кроме того, при несогласованности матриц сравнения критериев, требуется заново расставлять все приоритеты, что может занять много времени.

В ходе курсовой работы также было изучено линейное программирование и методы его решения – графический и симплексный методы.

Графический метод является очень наглядным и позволяет легко решать небольшие задачи линейного программирования. Однако его недостаток заключается в том, что он практически неприменим для задач с более чем двумя переменными.

Симплексный метод лишен этого недостатка и позволяет решать задачи линейного программирования с любым количеством переменных и ограничений. Однако время решения задачи может значительно увеличиваться при неудачных входных данных.

СПИСОК ИНФОРМАЦИОННЫХ ИСТОЧНИКОВ

1. Болотова Л. С. Многокритериальная оптимизация. Болотова Л. С., Сорокин А. Б. [Электронный ресурс] / Метод. указания по вып. курсовой работы — М.: МИРЭА, 2015.

2. Сорокин А. Б. Методы оптимизации: гибридные генетические алгоритмы. Сорокин А. Б. [Электронный ресурс] / Учебно-метод. пособие — М.: МИРЭА, 2016.

3. Сорокин А. Б. Линейное программирование: практикум. Сорокин А. Б., Бражникова Е. В., Платонова О. В. [Электронный ресурс] / Учебно-метод. пособие — М.: МИРЭА, 2017.

ПРИЛОЖЕНИЯ

Приложение А – Код реализации метода Парето на языке Python.

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

Приложение В – Код реализации МАИ на языке Python.

Приложение Г – Код реализации симплексного метода на языке Python.

Приложение А

Код реализации метода Парето на языке Python

Листинг А.1 — Реализация метода Парето

```
import pandas as pd
import numpy as np

data = {
    'Варианты завтрака': [
        'Овсянка с фруктами и орехами', 'Гречневая каша с овощами',
        'Яичница с овощами', 'Смузи из зелени и фруктов',
        'Творожная запеканка с логодами', 'Тосты с авокадо и лимон',
        'Кускус с овощами и фетой', 'Йогурт с мюсли и свежими фруктами',
        'Бутерброд с лососем', 'Бутерброд с сыром на козьем молоке'
    ],
    'Калорийность на порцию': [3, 3, 3, 2, 3, 4, 3, 2, 4, 2],
    'Питательная ценность': [5, 3, 4, 3, 3, 4, 4, 3, 4, 2],
    'Время приготовления': [25, 20, 15, 5, 30, 10, 15, 4, 7, 7],
    'Доступность': [3, 3, 3, 3, 2, 2, 2, 3, 2, 1],
    'Цена': [2, 2, 2, 1, 3, 4, 3, 2, 5, 4]
}

pd.set_option('display.max_columns', None)
pd.set_option('display.width', 200)

df = pd.DataFrame(data)
save_df = df
print(df)

for i in range(20): print()


def alternatives(alt1, alt2, directions):
    for i in range(len(directions)):
        if (directions[i] and alt1.iloc[i] < alt2.iloc[i]) or (not
directions[i] and alt1.iloc[i] > alt2.iloc[i]):
            return False
    return True


def generate_comparison_df(altered_df, criteria_directions):
    comparison_matrix = np.full((10, 10), None)
    for i in range(10):
        for j in range(i):
            if alternatives(altered_df.iloc[i, 1:], altered_df.iloc[j, 1:],
criteria_directions):
                comparison_matrix[i, j] = 'A' + str(i + 1)
            elif alternatives(altered_df.iloc[j, 1:], altered_df.iloc[i, 1:],
criteria_directions):
                comparison_matrix[i, j] = 'A' + str(j + 1)
            else:
                comparison_matrix[i, j] = 'н'
    comparison_df = pd.DataFrame(comparison_matrix,
                                 columns=['A1', 'A2', 'A3', 'A4', 'A5',
'A6', 'A7', 'A8', 'A9', 'A10'])
    comparison_df.index = ['A1', 'A2', 'A3', 'A4', 'A5', 'A6', 'A7', 'A8',
'A9', 'A10']

    return comparison_df
```

Продолжение Листинга А.1

```
criteria_directions = [True, True, False, True, False]
comparison_df = generate_comparison_df(df, criteria_directions)
print(comparison_df)
for i in range(20): print()

print(df[(df['Калорийность на порцию'] >= 3) & (df['Питательная ценность'] >=
4) & (df['Время приготовления'] <= 15)])
for i in range(20): print()

df = df[(df['Калорийность на порцию'] >= 3) & (df['Питательная ценность'] >=
4) & (df['Цена'] <= 2)]
df = df.sort_values(['Время приготовления'])
print(df)
for i in range(20): print()

df = save_df.sort_values(['Время приготовления', 'Калорийность на порцию',
'Доступность', 'Цена', 'Питательная ценность'],
                    ascending=[False, True, True, False, True])
print(df.iloc[::-1])
for i in range(20): print()
```

Приложение Б

Код реализации метода Электра II на языке Python.

Листинг Б.1. Реализация метода Электра II.

```
import numpy as np
import pandas as pd

crits = {
    "Калорийность": {"weight": 4, "direction": "max"},
    "Питательная ценность": {"weight": 3, "direction": "max"},
    "Время приготовления": {"weight": 5, "direction": "min"},
    "Доступность": {"weight": 5, "direction": "max"},
    "Цена": {"weight": 2, "direction": "min"}
}

alternatives = {
    "Овсянка с фруктами и орехами": [10, 10, 15, 15, 10],
    "Гречневая каша с овощами": [10, 10, 15, 10, 5],
    "Яичница с овощами": [5, 5, 5, 5, 5],
    "Смузи из зелени и фруктов": [5, 15, 5, 10, 10],
    "Творожная запеканка с ягодами": [15, 10, 10, 15, 5],
    "Тосты с авокадо и яйцом": [15, 10, 10, 10, 10],
    "Кускус с овощами и фетой": [5, 5, 10, 10, 5],
    "Йогурт с мюсли и свежими фруктами": [15, 15, 15, 15, 5],
    "Бутерброды с лососем": [10, 10, 5, 5, 5],
    "Бутерброд с сыром на козьем молоке": [10, 10, 15, 10, 10]
}
pd.set_option('display.max_columns', 100)
preference_matrix          =          pd.DataFrame(np.zeros((len(alternatives),
len(alternatives))),
                                                    index=alternatives.keys(),
columns=alternatives.keys())

def calculate_P_N_D(alt1, alt2, crits):
    P, N = 0, 0
    for i, crit in enumerate(crits):
        weight = crits[crit]["weight"]
        direction = crits[crit]["direction"]
        a1, a2 = alt1[i], alt2[i]
        if a1 != a2:
```

Продолжение Листинга Б.1.

```
                if (direction == "max" and a1 > a2) or (direction == "min"
and a1 < a2):
                P += weight
            else:
                N += weight
    D = P / N if N != 0 else np.inf
    return P, N, D

for i, alt1 in enumerate(alternatives.keys(), start=1):
    for j, alt2 in enumerate(alternatives.keys(), start=1):
        if alt1 != alt2:
                    P, N, D = calculate_P_N_D(np.array(alternatives[alt1]),
np.array(alternatives[alt2]), crits)
            value_to_input = None
            if D>=1 and D != np.inf:
                value_to_input = np.round(D, 3)
            elif D == np.inf:
                value_to_input = np.inf
            else:
                value_to_input = -1
            preference_matrix.at[alt1, alt2] = value_to_input

print(preference_matrix)

preference_matrix = preference_matrix.applymap(lambda x: np.nan if
isinstance(x, (int, float)) and x <= 2.4 else x)

for i in range(10): print()

print(preference_matrix)
```

Приложение В

Код реализации МАИ на языке Python.

Листинг В.1. Реализация метода анализа иерархий.

```
from print import pprint
import numpy as np

matrix = np.array([
 [1,      2,    1/2,   1/3,   3],
 [1/2,    1,    1/3,   1/4,   1],
 [2,      3,    1,     1/2,   2],
 [3,      4,    2,     1,     3],
 [1/3,    1,    1/2,   1/3,   1]
])

matrix1 = np.array([
 [1, 7, 1/3, 1/2, 5],
 [1/7, 1, 1/7, 1/6, 1/2],
 [3, 7, 1, 1, 5],
 [2, 6, 1, 1, 4],
 [1/5, 2, 1/5, 1/4, 1]
])

matrix2 = np.array([
[1, 3, 1/2, 1/4, 1/3],
 [1/3, 1, 1/3, 1/6, 1/4],
 [2, 3, 1, 1/2, 1/3],
 [4, 6, 2, 1, 1/2],
 [3, 4, 3, 2, 1]
])

matrix3 = np.array([
    [1,    3,   1/2,  5,  1/3],
    [1/3,  1,   1/4,  2,  1/5],
    [2,    4,   1,    7,  1/2],
    [1/5,  1/2, 1/7,  1,  1/9],
    [3,    5,   2,    9,  1]
])

matrix4 = np.array([
    [1, 5, 1/6, 1/7, 5],
    [1/5, 1, 1/8, 1/8, 3],
    [6, 8, 1, 1, 9],
    [7, 8, 1, 1, 9],
    [1/5, 1/3, 1/9, 1/9, 1]
])

matrix5 = np.array([
    [1,    2,    5,  1/3,  4],
    [1/2,  1,    3,  1/4,  1/2],
    [1/5,  1/3,  1,  1/7,  1/6],
    [3,    4,    7,  1,    5],
    [1/4,  2,    6,  1/5,  1]
])


def calculate_matrices(matrices):
    W_matrices = []
    for matrix in matrices:
```

Продолжение Листинга В.1.

```
          geometric_means = np.prod(matrix, axis=1) ** (1 / matrix.shape[1])
        W = geometric_means / np.sum(geometric_means)
        W_matrices.append(W)

        geometric_means_rounded = np.round(geometric_means, 3)
        sum_rounded = np.round(np.sum(geometric_means), 3)

        S = np.sum(matrix, axis=0)
        P = S * W
        lambda_max = np.sum(P) / np.sum(W)

        n = matrix.shape[0]
        IC = (lambda_max - n) / (n - 1)
        RI = {1: 0, 2: 0, 3: 0.58, 4: 0.9, 5: 1.12, 6: 1.24, 7: 1.32, 8:
1.41, 9: 1.45, 10: 1.49}
        OC = IC / RI[n] if n in RI and RI[n] != 0 else None
        W = np.round(W, 3)
        S = np.round(S, 3)
        P = np.round(P, 3)
        lambda_max = round(lambda_max, 3)
        IC = round(IC, 3)
        OC = round(OC, 3) if OC is not None else None

        pprint("Результат для матрицы:")
        print({
            'V': geometric_means_rounded,
            'sum': sum_rounded,
            'W': W,
            'S': S,
            'P': P,
            'lambda_max': lambda_max,
            'IC': IC,
            'OC': OC
        })
        print()

    W_c = W_matrices[0]
    W_x = np.array(W_matrices[1:])

    W_result = np.dot(W_x, W_c)
    print("Результат:")
    for i, weight in enumerate(W_result, start=1):
        print(f"A{i}: {weight:.3f}")

matrix = [matrix, matrix1, matrix2, matrix3, matrix4, matrix5]

calculate_matrices(matrix)
```

Приложение Г

Код реализации симплексного метода на языке Python.

Листинг Г.1. Реализация симплексного метода.

```
import numpy as np


def simplex(objective, constraints, limits):
    num_constraints, num_variables = constraints.shape

    simplex_table = np.zeros((num_constraints + 1, num_variables +
num_constraints + 1))
    simplex_table[:num_constraints, :num_variables] = constraints
    simplex_table[:num_constraints, num_variables:num_variables +
num_constraints] = np.eye(num_constraints)
    simplex_table[:num_constraints, -1] = limits
    simplex_table[-1, :num_variables] = -np.array(objective)

    basic_vars = list(range(num_variables, num_variables + num_constraints))

    while any(simplex_table[-1, :-1] < 0):
        entering = np.argmin(simplex_table[-1, :-1])
        ratios = [simplex_table[i, -1] / simplex_table[i, entering] if
simplex_table[i, entering] > 0 else float('inf') for i in
range(num_constraints)]
        leaving = np.argmin(ratios)
        pivot = simplex_table[leaving, entering]
        simplex_table[leaving, :] /= pivot
        for i in range(num_constraints + 1):
            if i != leaving:
                simplex_table[i, :] -= simplex_table[i, entering] *
simplex_table[leaving, :]
        basic_vars[leaving] = entering

    solution = np.zeros(num_variables)
    for i, var in enumerate(basic_vars):
        if var < num_variables:
            solution[var] = simplex_table[i, -1]

    return simplex_table[-1, -1], solution

objective = [7.5, 3, 6, 12]

constraints = np.array([
    [2, 1, 0.5, 4],
    [1, 5, 3, 0],
    [3, 0, 6, 1]
])
limits = [3400, 1200, 3000]

optimal_profit, production_distribution = simplex(objective, constraints,
limits)

print("Максимальная прибыль:", float(optimal_profit))
print("Распределение производства:", [float(x) for x in
production_distribution])
```