**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 6](#_Toc199807439)

[1 МЕТОД ПАРЕТО 7](#_Toc199807440)

[1.1 Выбор Парето-оптимального множества 7](#_Toc199807441)

[1.2 Указание верхних/нижних границ критериев. 9](#_Toc199807442)

[1.3 Субоптимизация 9](#_Toc199807443)

[1.4 Лексикографическая оптимизация 10](#_Toc199807444)

[1.5 Результаты работы программы 10](#_Toc199807445)

[1.6 Вывод по методу Парето 10](#_Toc199807446)

[2 МЕТОД ЭЛЕКТРА II 12](#_Toc199807447)

[2.1 Выбор лучшего варианта 12](#_Toc199807448)

[2.2 Веса предпочтений 14](#_Toc199807449)

[2.3 Результат работы программы 27](#_Toc199807450)

[2.4 Вывод по методу Электра II 27](#_Toc199807451)

[3 МЕТОД АНАЛИЗА ИЕРАРХИЙ 28](#_Toc199807452)

[3.1 Постановка задачи 28](#_Toc199807453)

[3.2 Представление проблемы в виде иерархии 28](#_Toc199807454)

[3.3 Установка приоритетов критериев 29](#_Toc199807455)

[3.4 Синтез приоритетов 31](#_Toc199807456)

[3.5 Согласованность локальных приоритетов 39](#_Toc199807457)

[3.6 Синтез альтернатив 46](#_Toc199807458)

[3.7 Результаты работы программы 47](#_Toc199807459)

[3.8 Вывод по методу МАИ 47](#_Toc199807460)

[4 ГРАФИЧЕСКИЙ МЕТОД 48](#_Toc199807461)

[4.1 Постановка задачи 48](#_Toc199807462)

[4.2 Данные индивидуального варианта 48](#_Toc199807463)

[4.3 Подготовка данных 49](#_Toc199807464)

[4.4 Построение графика 51](#_Toc199807465)

[4.5 Выделение области допустимых решений 51](#_Toc199807466)

[4.6 Максимум функции 52](#_Toc199807467)

[4.7 Минимум функции 54](#_Toc199807468)

[4.8 Вывод по графическому методу 55](#_Toc199807469)

[5 СИМПЛЕКСНЫЙ МЕТОД 56](#_Toc199807470)

[5.1 Постановка задачи 56](#_Toc199807471)

[5.2 Математическая модель задачи 57](#_Toc199807472)

[5.3 Консольный результат работы 64](#_Toc199807473)

[5.4 Вывод по симплексному методу 64](#_Toc199807474)

[6 ДВОЙСТВЕННАЯ ЗАДАЧА 66](#_Toc199807475)

[6.1 Постановка задачи 66](#_Toc199807476)

[6.2 Математическая модель исходной задачи 67](#_Toc199807477)

[6.3 Соответствующая исходной двойственная задача 67](#_Toc199807478)

[6.4 Первая теорема двойственности 68](#_Toc199807479)

[6.5 Вторая теорема двойственности 71](#_Toc199807480)

[6.6 Третья теорема двойственности 74](#_Toc199807481)

[6.7 Консольный результат программы 78](#_Toc199807482)

[6.8 Вывод по двойственному методу 78](#_Toc199807483)

[ЗАКЛЮЧЕНИЕ 80](#_Toc199807484)

[СПИСОК ИНФОРМАЦИОННЫХ ИСТОЧНИКОВ 81](#_Toc199807485)

[ПРИЛОЖЕНИЯ 82](#_Toc199807486)

ВВЕДЕНИЕ

Теория принятия решений занимается разработкой и применением методов выбора наилучшего варианта из множества альтернатив в условиях неопределенности и ограничений. В рамках этой дисциплины особое внимание уделяется мультикритериальному анализу, где необходимо учитывать сразу несколько, порой противоречивых, критериев. Метод Паретто позволяет отобрать решения, не уступающие другим по всем параметрам. Метод Электро II применяется для ранжирования и сравнения альтернатив с использованием отношения превосходства. Метод анализа иерархий (AHP) помогает структурировать сложные задачи принятия решений, разделяя их на уровни и определяя приоритеты. Также в теории активно используются графические методы, симплексный метод линейного программирования и двойственная задача для оптимизации решений при наличии численных ограничений.

1. МЕТОД ПАРЕТО

Метод Парето, или принцип Парето, широко используется в задачах многокритериального принятия решений и оптимизации. Основная идея метода заключается в следующем:

Сравнение альтернатив по нескольким критериям:

При наличии множества вариантов (альтернатив) решения сравниваются по ряду критериев. Один вариант считается доминирующим по отношению к другому, если он по каждому критерию не хуже, а хотя бы по одному – лучше.

Парето‑оптимальное множество:

Решения, которые не доминируются ни одним другим, называются Парето‑оптимальными. Это означает, что нельзя улучшить хотя бы один критерий, не ухудшив при этом другой. Такое множество представляет собой набор лучших компромиссных вариантов, из которого затем может быть выбран окончательный вариант с учётом дополнительных предпочтений или ограничений.

## **Выбор Парето-оптимального множества**

Задается множество альтернатив и их критериев со стремлениями.

Каждая альтернатива сравнивается с другими. Если найдется такая альтернатива sj , что sj≻si (то есть sj не хуже si по всем критериям и хотя бы по одному критерию лучше), то si считается доминируемой, а sj доминирующей.

Если для si ни одна другая альтернатива не доминирует над ней, si считается Парето‑оптимальной.

Собираем все альтернативы, которые не доминируются ни одной другой, в множество, которое является Парето-оптимальным множеством.

Предметная область: Выбор космического корабля.

*Таблица 1.1.1 – Альтернативы со стремлениями*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Космический корабль** | **Цена**  **(кредиты) (-)** | **Скорость**  **(км/ч) (+)** | **Время входа в гиперпространство**  **(сек) (-)** | **Количество орудий**  **(шт) (+)** | **Мощность щитов**  **(Вт) (+)** |
| 1 | TIE Fighter | 20000 | 5000 | 3.0 | 2 | 100 |
| 2 | TZ-24 | 22000 | 4900 | 3.2 | 4 | 120 |
| 3 | S-100 | 21000 | 4800 | 3.1 | 3 | 150 |
| 4 | F-T2 | 30000 | 5100 | 4.0 | 3 | 110 |
| 5 | CR90 | 25000 | 4600 | 3.5 | 2 | 130 |
| 6 | IL-5 | 26000 | 4700 | 3.7 | 2 | 100 |
| 7 | FT-6 | 35000 | 4400 | 4.5 | 2 | 100 |
| 8 | FT-8 | 34000 | 4500 | 4.3 | 3 | 115 |
| 9 | S-13 | 33000 | 4600 | 4.1 | 2 | 105 |
| 10 | S-SC4 | 32000 | 4700 | 3.9 | 3 | 125 |

Найдём оптимальное множество:

*Таблица 1.1.2 – Нахождение оптимального множества*

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 1 |  |  |  |  |  |  |  |  |  |  |
| 2 |  |  |  |  |  |  |  |  |  |  |
| 3 |  |  |  |  |  |  |  |  |  |  |
| 4 |  |  |  |  |  |  |  |  |  |  |
| 5 |  |  | 3 |  |  |  |  |  |  |  |
| 6 | 1 | 2 | 3 |  |  |  |  |  |  |  |
| 7 |  | 2 | 3 | 4 | 5 |  |  |  |  |  |
| 8 |  | 2 | 3 |  |  |  |  |  |  |  |
| 9 |  | 2 | 3 | 4 | 5 |  |  |  |  |  |
| 10 |  |  | 3 |  |  |  |  |  |  |  |

Парето-оптимальное множество альтернатив {TIE Fighter, TZ-24, S-100, F-T2, CR90}.

# Указание верхних/нижних границ критериев.

Парето-оптимальное множество альтернатив {TIE Fighter, TZ-24, S-100, F-T2, CR90}.

Для вариантов решений зададим границы. Цена корабля не должна превышать 22000 кредитов, скорость не менее 4800 км/ч, время входа в гиперпространство не менее 3.0 секунд, количество орудий 3 и более, мощность щитов больше 100 ватт.

*Таблица 1.2 – Парето-оптимальные варианты с указанием границ*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Космический корабль** | **Цена**  **(кредиты) (-)** | **Скорость**  **(км/ч) (+)** | **Время входа в гиперпространство**  **(сек) (-)** | **Количество орудий**  **(шт) (+)** | **Мощность щитов**  **(Вт) (+)** |
| 1 | TZ-24 | 22000 | 4900 | 3.2 | 4 | 120 |
| 2 | S-100 | 21000 | 4800 | 3.1 | 3 | 150 |

* 1. **Субоптимизация**

Парето-оптимальное множество альтернатив {TIE Fighter, TZ-24, S-100, F-T2, CR90}.

Главный критерий: Скорость.

Ограничения: Цена не более 25000 кредитов, время входа в гиперпространство больше 3.2 секунда, количество орудий больше 2, мощность щитов не менее 120 ватт.

*Таблица 3 – Парето-оптимальные варианты*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Космический корабль** | **Цена**  **(кредиты) (-)** | **Скорость**  **(км/ч) (+)** | **Время входа в гиперпространство**  **(сек) (-)** | **Количество орудий**  **(шт) (+)** | **Мощность щитов**  **(Вт) (+)** |
| 1 | TIE Fighter | 20000 | 5000 | 3.0 | 2 | 100 |
| 2 | TZ-24 | 22000 | 4900 | 3.2 | 4 | 120 |
| 3 | CR90 | 25000 | 4600 | 3.5 | 2 | 130 |
| 4 | S-100 | 21000 | 4800 | 3.1 | 3 | 150 |
| 5 | F-T2 | 30000 | 5100 | 4.0 | 3 | 110 |

Самый лучший корабль: TZ-24

**1.4 Лексикографическая оптимизация**

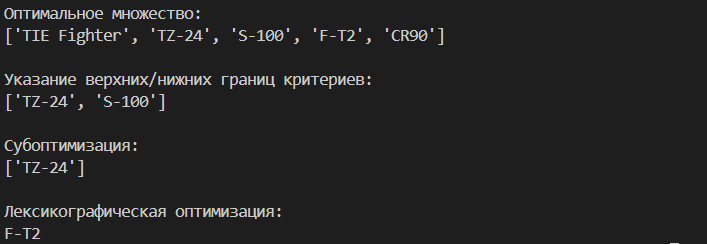
Упорядочим критерии по приоритету: важнейший критерий - Скорость, следующие: количество орудий, мощность щитов, цена, время входа в гиперпространство.

*Таблица 4 – Альтернативы со стремлениями*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Космический корабль** | **Цена**  **(кредиты) (-)** | **Скорость**  **(км/ч) (+)** | **Время входа в гиперпространство**  **(сек) (-)** | **Количество орудий**  **(шт) (+)** | **Мощность щитов**  **(Вт) (+)** |
| 1 | TIE Fighter | 20000 | 5000 | 3.0 | 2 | 100 |
| 2 | TZ-24 | 22000 | 4900 | 3.2 | 4 | 120 |
| 3 | S-100 | 21000 | 4800 | 3.1 | 3 | 150 |
| 4 | F-T2 | 30000 | 5100 | 4.0 | 3 | 110 |
| 5 | CR90 | 25000 | 4600 | 3.5 | 2 | 130 |
| 6 | IL-5 | 26000 | 4700 | 3.7 | 2 | 100 |
| 7 | FT-6 | 35000 | 4400 | 4.5 | 2 | 100 |
| 8 | FT-8 | 34000 | 4500 | 4.3 | 3 | 115 |
| 9 | S-13 | 33000 | 4600 | 4.1 | 2 | 105 |
| 10 | S-SC4 | 32000 | 4700 | 3.9 | 3 | 125 |

Самый лучший корабль: F-T2.

**1.5 Результаты работы программы**

**Рисунок 1.1 – Лексикографическая оптимизация.**

**1.6 Вывод по методу Парето**

В данной работе был изучен алгоритм Парето, выполнен ручной расчёт по этому алгоритму, написана программа на языке Python, которая реализует данный алгоритм, а также изучены улучшения этого алгоритма указания верхних и нижних границ критериев, субоптимизация, лексикографическая оптимизация.

# 2 МЕТОД ЭЛЕКТРА II

Данный метод сравнивает все альтернативы попарно и составляет таблицу предпочтений в которую записываются коэффициенты, полученные во время сравнения. Рассматриваем все пары проектов i и j. Если по какому-либо критерию i-ый проект лучше, чем j-ый, то соответствующий критерию вес прибавляется к Pij (эти баллы символизируют выбор «За»), в противном случае — к Nij (эти баллы символизируют выбор «Против»). Затем, когда по паре i и j рассмотрены все критерии, находятся отношения Dij = Pij/Nij и Dji = Pji/Nji. Эти отношения и записываются в таблицу предпочтений.

Данный метод используется в логистике, для выбора оптимального маршрута, финансовых операциях, оценка выгодных инвестиций и в управлении множеством других проектов.

## **2.1 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 |
| Цена | 5 | 31000 кредитов и более  26000 – 30000 кредитов  25000 кредитов и менее | 30  20  10 | min |
| Скорость | 5 | 5000 км/ч и более  4900 км/ч  4800 км/ч  4700 км/ч  4600 км/ч  4500 км/ч и менее | 30  25  20  15  10  5 | max |
| Время входа в гиперпространство | 4 | 4.1 – 4.5 сек  3.6 – 4.0 сек  3.0 – 3.5 сек | 30  20  10 | min |
| Количество орудий | 4 | 4 шт  3 шт  2 шт | 30  20  10 | max |
| Мощность щитов | 3 | 141 – 150 Вт  131 – 140 Вт  121 – 130 Вт  111 – 120 Вт  100 – 110 Вт | 25  20  15  10  5 | max |

Составлена таблица оценок выбора лучшего космического коробля. Для 10-ти альтернатив заполнена Таблицу 2.2.

*Таблица 2.2 – Таблица оценок по критериям*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| № | Варианты решений | Критерии | | | | |
| **Цена** | **Скорость** | **Время входа в гиперпространство** | **Количество орудий** | **Мощность щитов** |
| 1 | TIE Fighter | 10 | 30 | 10 | 10 | 5 |
| 2 | TZ-24 | 10 | 25 | 10 | 30 | 10 |
| 3 | S-100 | 10 | 20 | 10 | 20 | 25 |
| 4 | F-T2 | 20 | 30 | 20 | 20 | 5 |
| 5 | CR90 | 10 | 10 | 10 | 10 | 15 |
| 6 | IL-5 | 10 | 15 | 20 | 10 | 5 |
| 7 | FT-6 | 30 | 5 | 30 | 10 | 5 |
| 8 | FT-8 | 30 | 5 | 30 | 20 | 10 |
| 9 | S-13 | 30 | 10 | 30 | 10 | 5 |
| 10 | S-SC4 | 30 | 15 | 20 | 20 | 15 |
| Вес | | 5 | 5 | 4 | 4 | 3 |
| Стремление | | min | max | min | max | max |

**2.2 Веса предпочтений**

Рассмотрим альтернативы 1 и 2 (i=1,j=2):

P12 = 0 + 30 + 0 + 0 + 0 = 30;

N12 = 0 + 0 + 0 + 30+ 10 = 40;

D12 = P12/N12 = 30/40 < 1 – отбрасываем.

P21 = 0 + 0 + 0 + 30 + 10 = 40;

N21 = 0 + 30 + 0 + 0 + 0 = 25;

D21 = P21/N21 = 40/30 > 1 – принимаем.

Рассмотрим альтернативы 1 и 3 (i=1,j=3):

P13 = 0 + 30 + 0 + 0 + 0= 30;

N13 = 0 + 0 + 0 + 20 + 25 = 45;

D13 = P13/N13 = 30/45 < 1 – отбрасываем.

P31 = 0 + 0 + 0 + 20 + 25 = 45;

N31 = 0 + 30 + 0 + 0 + 0 = 30;

D31 = P31/N31 = 45/30 > 1 – принимаем.

Рассмотрим альтернативы 1 и 4 (i=1,j=4):

P14 = 10 + 0 + 10 + 0 + 0 = 20;

N14 = 0 + 0 + 0 + 20 + 0 = 20;

D14 = P14/N14 = 30/20 == 1 – отбрасываем.

P41 = 0 + 0 + 0 + 20 + 0 = 20;

N41 = 10 + 0 + 10 + 0 + 0 = 20;

D41 =P41/N41 = 20/30 == 1 – отбрасываем.

Рассмотрим альтернативы 1 и 5 (i=1,j=5):

P15 = 0 + 30 + 0 + 0 + 0 = 30;

N15 = 0 + 0 + 0 + 0 + 15 = 15;

D15 = P15/N15 = 30/15 > 1 – принимаем.

P51 = 0 + 0 + 0 + 0 + 15 = 15;

N51 = 0 + 30 + 0 + 0 + 0 = 30;

D51 = P51/N51 = 15/30 < 1 – отбрасываем.

Рассмотрим альтернативы 1 и 6 (i=1,j=6):

P16 = 0 + 30 + 10 + 0 + 0 = 40;

N16 = 0 + 0 + 0 + 0 + 0 = 0;

D16 = P16/N16 = 40/0 > 1 – принимаем.

P61 = 0 + 0 + 0 + 0 + 0 = 0;

N61 = 0 + 30 + 10 + 0 + 0 = 40;

D61 = P61/N61 = 0/40 < 1 – отбрасываем.

Рассмотрим альтернативы 1 и 7 (i=1,j=7):

P17 = 10 + 30 + 10 + 0 + 0 = 50;

N17 = 0 + 0 + 0 + 0 + 0 = 0;

D17 = P17/N17 = 50/0 > 1 – принимаем.

P71 = 0 + 0 + 0 + 0 + 0 = 0;

N71 = 10 + 30 + 10 + 0 + 0 =5 0;

D71 = P71/N71 = 0/50 < 1 – отбрасываем.

Рассмотрим альтернативы 1 и 8 (i=1,j=8):

P18 = 10 + 30 + 10 + 0 + 0 = 50;

N18 = 0 + 0 + 0 + 20 + 10 = 30;

D18 = P18/N18 = 50/30 > 1 – принимаем.

P81 = 0 + 0 + 0 + 20 + 10 = 30;

N81 = 10 + 30 + 10 + 0 + 0 = 50;

D81 = P81/N81 = 30/50 < 1 – отбрасываем.

Рассмотрим альтернативы 1 и 9 (i=1,j=9):

P19 = 10 + 30 + 10 + 0 + 0 = 50;

N19 = 0 + 0 + 0 + 0 + 0 = 0;

D19 = P19/N19 = 50/0 > 1 – принимаем.

P91 = 0 + 0 + 0 + 0 + 0 = 0;

N91 = 10 + 30 + 10 + 0 + 0 = 50;

D91 = P91/N91 = 0/50 < 1 – отбрасываем.

Рассмотрим альтернативы 1 и 10 (i=1,j=10):

P110 = 10 + 30 + 10 + 0 + 0 = 50;

N110 = 0 + 0 + 0 + 20 + 15 = 35;

D110 = P110/N110 = 50/35 > 1 – принимаем.

P101 = 0 + 0 + 0 + 20 + 15 = 35;

N101 = 10 + 30 + 10 + 0 + 0 = 50;

D101 = P101/N101 = 35/50 < 1 – отбрасываем.

Рассмотрим альтернативы 2 и 3 (i=2,j=3):

P23 = 0 + 25 + 0 + 30 + 0 = 55;

N23 = 0 + 0 + 0 + 0 + 25 = 25;

D23 = P23/N23 = 55/25 > 1 – принимаем.

P32 = 0 + 0 + 0 + 0 + 25 = 25;

N32 = 0 + 25 + 0 + 30 + 0 = 55;

D32 = P32/N32 = 25/55 < 1 – отбрасываем.

Рассмотрим альтернативы 2 и 4 (i=2,j=4):

P24 = 10 + 0 + 10 + 30 + 10 = 60;

N24 = 0 + 30 + 0 + 0 + 0 = 30;

D24 = P24/N24 = 60/30 > 1 – принимаем.

P42 = 0 + 30 + 0 + 0 + 0 = 30;

N42 = 10 + 0 + 10 + 30 + 10 = 60;

D42 = P42/N42 = 30/60 < 1 – отбрасываем.

Рассмотрим альтернативы 2 и 5 (i=2,j=5):

P25 = 0 + 25 + 0 + 30 + 0 = 55;

N25 = 0 + 0 + 0 + 0 + 15 = 15;

D25 = P25/N25 = 55/15 > 1 – принимаем.

P52 = 0 + 0 + 0 + 0 + 15 = 15;

N52 = 0 + 25 + 0 + 30 + 0 = 55;

D52 = P52/N52 = 15/55 < 1 – отбрасываем.

Рассмотрим альтернативы 2 и 6 (i=2,j=6):

P26 = 0 + 25 + 10 + 30 + 10 = 65;

N26 = 0 + 0 + 0 + 0 + 0 = 0;

D26 = P26/N26 = 65/0 > 1 – принимаем.

P62 = 0 + 0 + 0 + 0 + 0 = 0;

N62 = 0 + 25 + 10 + 30 + 10 = 65;

D62 = P62/N62 = 0/65 < 1 – отбрасываем.

Рассмотрим альтернативы 2 и 7 (i=2,j=7):

P27 = 10 + 25 + 10 + 30 + 10 = 85;

N27 = 0 + 0 + 0 + 0 + 0 = 0;

D27 = P27/N27 = 85/0 > 1 – принимаем.

P72 = 0 + 0 + 0 + 0 + 0 = 0;

N72 = 10 + 25 + 10 + 30 + 10 = 85;

D72 = P72/N72 = 0/85 < 1 – отбрасываем.

Рассмотрим альтернативы 2 и 8 (i=2,j=8):

P28 = 10 + 25 + 10 + 30 + 0 = 75;

N28 = 0 + 0 + 0 + 0 + 0 = 0;

D28 = P28/N28 = 75/0 > 1 – принимаем.

P82 = 0 + 0 + 0 + 0 + 0 = 0;

N82 = 10 + 25 + 10 + 30 + 0 = 75;

D82 = P82/N82 = 0/75 < 1 – отбрасываем.

Рассмотрим альтернативы 2 и 9 (i=2,j=9):

P29 = 10 + 25 + 10 + 30 + 10 = 85;

N29 = 0 + 0 + 0 + 0 + 0 = 0;

D29 = P29/N29 = 85/0 > 1 – принимаем.

P92 = 0 + 0 + 0 + 0 + 0 = 0;

N92 = 10 + 25 + 10 + 30 + 10 = 85;

D92 = P92/N92 = 0/85 < 1 – отбрасываем.

Рассмотрим альтернативы 2 и 10 (i=2,j=10):

P210 = 10 + 25 + 10 + 30 + 0 = 75;

N210 = 0 + 0 + 0 + 0 + 15 = 15;

D210 = P210/N210 = 75/15 > 1 – принимаем.

P102 = 0 + 0 + 0 + 0 + 15 = 15;

N102 = 10 + 25 + 10 + 30 + 0 = 75;

D102 = P102/N102 = 15/75 < 1 – отбрасываем.

Рассмотрим альтернативы 3 и 4 (i=3,j=4):

P34 = 10 + 0 + 10 + 0 + 25 = 45;

N34 = 0 + 30 + 0 + 0 + 0 = 30;

D34 = P34/N34 = 45/30 > 1 – принимаем.

P43 = 0 + 30 + 0 + 0 + 0 = 30;

N43 = 10 + 0 + 10 + 0 + 25 = 45;

D43 = P43/N43 = 30/45 < 1 – отбрасываем.

Рассмотрим альтернативы 3 и 5 (i=3,j=5):

P35 = 0 + 20 + 0 + 20 + 25 = 65;

N35 = 0 + 0 + 0 + 0 + 0 = 0;

D35 = P35/N35 = 65/0 > 1 – принимаем.

P53 = 0 + 0 + 0 + 0 + 0 = 0;

N53 = 0 + 20 + 0 + 20 + 25 = 65;

D53 = P53/N53 = 0/65 < 1 – отбрасываем.

Рассмотрим альтернативы 3 и 6 (i=3,j=6):

P36 = 0 + 20 + 10 + 20 + 25 = 75;

N36 = 0 + 0 + 0 + 0 + 0 = 0;

D36 = P36/N36 = 75/0 > 1 – принимаем.

P63 = 0 + 0 + 0 + 0 + 0 = 0;

N63 = 0 + 20 + 10 + 20 + 25 = 75;

D63 = P63/N63 = 0/75 < 1 – отбрасываем.

Рассмотрим альтернативы 3 и 7 (i=3,j=7):

P37 = 10 + 20+ 10 + 20 + 25 = 85;

N37 = 0 + 0 + 0 + 0 + 0 = 0;

D37 = P37/N37 = 85/0 > 1 – принимаем.

P73 = 0 + 0 + 0 + 0 + 0 = 0;

N73 = 10 + 20+ 10 + 20 + 25 = 85;

D73 = P73/N73 = 0/85 < 1 – отбрасываем.

Рассмотрим альтернативы 3 и 8 (i=3,j=8):

P38 = 10 + 20 + 10 + 0 + 25 = 65;

N38 = 0 + 0 + 0 + 0 + 0 = 0;

D38 = P38/N38 = 65/0 > 1 – принимаем.

P83 = 0 + 0 + 0 + 0 + 0 = 0;

N83 = 10 + 20 + 10 + 0 + 25 = 65;

D83 = P83/N83 = 0/65 < 1 – отбрасываем.

Рассмотрим альтернативы 3 и 9 (i=3,j=9):

P39 = 10 + 20 + 10 + 20 + 25 = 85;

N39 = 0 + 0 + 0 + 0 + 0 = 0;

D39 = P39/N39 = 85/0 > 1 – принимаем.

P93 = 0 + 0 + 0 + 0 + 0 = 0;

N93 = 10 + 20 + 10 + 20 + 25 = 85;

D93 = P93/N93 = 0/85 < 1 – отбрасываем.

Рассмотрим альтернативы 3 и 10 (i=3,j=10):

P310 = 10 + 20 + 10 + 0 + 25 = 65;

N310 = 0 + 0 + 0 + 0 + 0 = 0;

D310 = P310/N310 = 65/0 > 1 – принимаем.

P103 = 0 + 0 + 0 + 0 + 0 = 0;

N103 = 10 + 20 + 10 + 0 + 25 = 65;

D103 = P103/N103 = 0/65 < 1 – отбрасываем.

Рассмотрим альтернативы 4 и 5 (i=4,j=5):

P45 = 0 + 30 + 0 + 20 + 0 = 50;

N45 = 10 + 0 + 10 + 0 + 15 = 35;

D45 = P45/N45 = 50/35 > 1 – принимаем.

P54 = 10 + 0 + 10 + 0 + 15 = 35;

N54 = 0 + 30 + 0 + 20 + 0 = 50;

D54 = P54/N54 = 35/50 < 1 – отбрасываем.

Рассмотрим альтернативы 4 и 6 (i=4,j=6):

P46 = 0 + 30 + 0 + 20 + 0 = 50;

N46 = 10 + 0 + 0 + 0 + 0 = 10;

D46 = P46/N46 = 50/10 > 1 – принимаем.

P64 = 10 + 0 + 0 + 0 + 0 = 10;

N64 = 0 + 30 + 0 + 20 + 0 = 50;

D64 = P64/N64 = 10/50 < 1 – отбрасываем.

Рассмотрим альтернативы 4 и 7 (i=4,j=7):

P47 = 20 + 30 + 20 + 20 + 0 = 90;

N47 = 0 + 0 + 0 + 0 + 0 = 0;

D47 = P47/N47 = 90/0 > 1 – принимаем.

P74 = 0 + 0 + 0 + 0 + 0 = 0;

N74 = 20 + 30 + 20 + 20 + 0 = 90;

D74 = P74/N74 = 0/90 < 1 – отбрасываем.

Рассмотрим альтернативы 4 и 8 (i=4,j=8):

P48 = 20 + 30 + 20 + 0 + 0 = 70;

N48 = 0 + 0 + 0 + 0 + 10 = 10;

D48 = P48/N48 = 70/10 > 1 – принимаем.

P84 = 0 + 0 + 0 + 0 + 10 = 10;

N84 = 20 + 30 + 20 + 0 + 0 = 70;

D84 = P84/N84 = 10/70 < 1 – отбрасываем.

Рассмотрим альтернативы 4 и 9 (i=4,j=9):

P49 = 20 + 30 + 20 + 20 + 0 = 90;

N49 = 0 + 0 + 0 + 0 + 0= 0;

D49 = P49/N49 = 90/0 > 1 – принимаем.

P94 = 0 + 0 + 0 + 0 + 0= 0;

N94 = 20 + 30 + 20 + 20 + 0 = 90;

D94 = P94/N94 = 0/90 < 1 – отбрасываем.

Рассмотрим альтернативы 4 и 10 (i=4,j=10):

P410 = 20 + 30 + 0 + 0 + 0 = 50;

N410 = 0 + 0 + 0 + 0 + 15 = 15;

D410 = P410/N410 = 50/15 > 1 – принимаем.

P104 = 0 + 0 + 0 + 0 + 15 = 15;

N104 = 20 + 30 + 0 + 0 + 0 = 60;

D104 = P104/N104 = 15/50 < 1 – отбрасываем.

Рассмотрим альтернативы 5 и 6 (i=5,j=6):

P56 = 0 + 0 + 10 + 0 + 15 = 25;

N56 = 0 + 15 + 0 + 0 + 0 = 15;

D56 = P56/N56 = 25/15 > 1 – принимаем.

P65 = 0 + 15 + 0 + 0 + 0 = 15;

N65 = 0 + 0 + 10 + 0 + 15 = 25;

D65 = P65/N65 = 15/25 < 1 – отбрасываем.

Рассмотрим альтернативы 5 и 7 (i=5,j=7):

P57 = 10 + 10 + 10 + 0 + 15 = 35;

N57 = 0 + 0 + 0 + 0 + 0 = 0;

D57 = P57/N57 = 35/0 > 1 – принимаем.

P75 = 0 + 0 + 0 + 0 + 0 = 0;

N75 = 10 + 10 +1 0 + 0 + 15 = 35;

D75 = P75/N75 = 0/35 < 1 – отбрасываем.

Рассмотрим альтернативы 5 и 8 (i=5,j=8):

P58 = 10 + 10 + 10 + 0 + 15 = 45;

N58 = 0 + 0 + 0 + 20 + 0 = 20;

D58 = P58/N58 = 35/20 > 1 – принимаем.

P85 = 0 + 0 + 0 + 20 + 0 = 20;

N85 = 10 + 10 + 10 + 0 + 15 = 45;

D85 = P85/N85 = 20/35 < 1 – отбрасываем.

Рассмотрим альтернативы 5 и 9 (i=5,j=9):

P59 = 10 + 0 + 10 + 0 + 15 = 35;

N59 = 0 + 0 + 0 + 0 + 0 = 0;

D59 = P59/N59 = 35/0 > 1 – принимаем.

P95 = 0 + 0 + 0 + 0 + 0 = 0;

N95 = 10 + 0 + 10 + 0 + 15 = 35;

D95 = P95/N95 = 0/35 < 1 – отбрасываем.

Рассмотрим альтернативы 5 и 10 (i=5,j=10):

P510 = 10 + 0 + 10 + 0 + 0 = 20;

N510 = 0 + 15 + 0 + 20 + 0 = 35;

D510 = P510/N510 = 20/35 < 1 – отбрасываем.

P105 = 0 + 15 + 0 + 20 + 0 = 35;

N105 = 10 + 0 + 10 + 0 + 0 = 20;

D105 = P105/N105 = 35/20 > 1 – принимаем.

Рассмотрим альтернативы 6 и 7 (i=6,j=7):

P67 = 10 + 15 + 20 + 0 + 0 = 45;

N67 = 0 + 0 + 0 + 0 + 0 = 0;

D67 = P67/N67 = 45/0 > 1 – принимаем.

P76 = 0 + 0 + 0 + 0 + 0 = 0;

N76 = 10 + 15 + 20 + 0 + 0 = 45;

D76 = P76/N76 = 0/45 < 1 – отбрасываем.

Рассмотрим альтернативы 6 и 8 (i=6,j=8):

P68 = 10 + 15 + 20 + 0 + 0 = 45;

N68 = 0 + 0 + 0 + 20 + 10 = 30;

D68 = P68/N68 = 45/30 > 1 – принимаем.

P86 = 0 + 0 + 0 + 20 + 10 = 30;

N86 = 10 + 15 + 20 + 0 + 0 = 45;

D86 = P86/N86 = 30/45 < 1 – отбрасываем.

Рассмотрим альтернативы 6 и 9 (i=6,j=9):

P69 = 10 + 15 + 20 + 0 + 0 = 45;

N69 = 0 + 0 + 0 + 0 + 0 = 0;

D69 = P69/N69 = 45/0 > 1 – принимаем.

P96 = 0 + 0 + 0 + 0 + 0 = 0;

N96 = 10 + 15 + 20 + 0 + 0 = 45;

D96 = P96/N96 = 0/45 < 1 – отбрасываем.

Рассмотрим альтернативы 6 и 10 (i=6,j=10):

P610 = 10 + 0 + 0 + 0 + 0 = 10;

N610 = 0 + 0 + 0 + 20 + 15 = 35;

D610 = P610/N610 = 10/35 < 1 – отбрасываем.

P106 = 0 + 0 + 0 + 20 + 15 = 35;

N106 = 10 + 0 + 0 + 0 + 0 = 10;

D106 = P106/N106 = 35/10 > 1 – принимаем.

Рассмотрим альтернативы 7 и 8 (i=7,j=8):

P78 = 0 + 0 + 0 + 0 + 0 = 0;

N78 = 0 + 0 + 0 + 20 + 10 = 30;

D78 = P78/N78 = 0/30 < 1 – отбрасываем.

P87 = 0 + 0 + 0 + 20 + 10 = 30;

N87 = 0 + 0 + 0 + 0 + 0 = 0;

D87 = P87/N87 = 30/0 > 1 – принимаем.

Рассмотрим альтернативы 7 и 9 (i=7,j=9):

P79 = 0 + 0 + 0 + 0 + 0 = 0;

N79 = 0 + 10 + 0 + 0 + 0 = 0;

D79 = P79/N79 = 0/10 < 1 – отбрасываем.

P97 = 0 + 10 + 0 + 0 + 0 = 0;

N97 = 0 + 0 + 0 + 0 + 0 = 0;

D97 = P97/N97 = 10/0 > 1 – принимаем.

Рассмотрим альтернативы 7 и 10 (i=7,j=10):

P710 = 0 + 0 + 0 + 0 + 0 = 0;

N710 = 0 + 15 + 20 + 20 + 15 = 70;

D710 = P710/N710 = 0/70 < 1 – отбрасываем.

P107 = 0 + 15 + 20 + 20 + 15 = 70;

N107 = 0 + 0 + 0 + 0 + 0 = 0;

D107 = P107/N107 = 70/0 > 1 – принимаем.

Рассмотрим альтернативы 8 и 9 (i=8,j=9):

P89 = 0 + 0 + 0 + 20 + 10 = 30;

N89 = 0 + 10 + 0 + 0 + 0 = 10;

D89 = P89/N89 = 30/10 > 1 – принимаем.

P98 = 0 + 10 + 0 + 0 + 0 = 10;

N98 = 0 + 0 + 0 + 20 + 10 = 30;

D98 = P98/N98 = 10/30 < 1 – отбрасываем.

Рассмотрим альтернативы 8 и 10 (i=8,j=10):

P810 = 0 + 0 + 0 + 0 + 0 = 0;

N810 = 0 + 15 + 20 + 0 + 15 = 50;

D810 = P810/N810 = 0/50 < 1 – отбрасываем.

P108 = 0 + 15 + 20 + 0 + 15 = 50;

N108 = 0 + 0 + 0 + 0 + 0 = 0;

D108 = P108/N108 = 50/0 > 1 – принимаем.

Рассмотрим альтернативы 9 и 10 (i=9,j=10):

P910 = 0 + 0 + 0 + 0 + 0 = 0;

N910 = 0 + 15 + 20 + 20 + 15 = 70;

D910 = P910/N910 = 0/70 < 1 – отбрасываем.

P109 = 0 + 15 + 20 + 20 + 15 = 70;

N109 = 0 + 0 + 0 + 0 + 0 = 0;

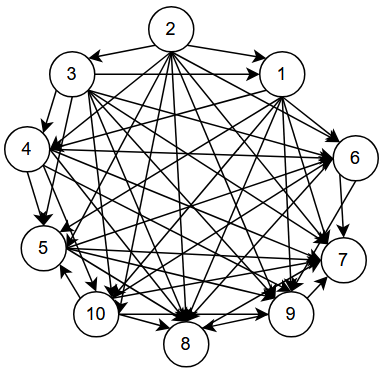
D109 = P109/N109 = 70/0 > 1 – принимаем.

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 |  |  |  | 30/15 | ∞ | ∞ | 50/30 | ∞ | 50/35 |
| 2 | 40/30 | x | 55/25 | 60/30 | 55/15 | ∞ | ∞ | ∞ | ∞ | 75/15 |
| 3 | 45/30 |  | x | 45/30 | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ |
| 4 |  |  |  | x | 50/35 | 50/10 | ∞ | 70/10 | ∞ | 50/15 |
| 5 |  |  |  |  | x | 25/15 | ∞ | 45/20 | ∞ |  |
| 6 |  |  |  |  |  | x | ∞ | 45/30 | ∞ |  |
| 7 |  |  |  |  |  |  | x |  |  |  |
| 8 |  |  |  |  |  |  | ∞ | x | 30/10 |  |
| 9 |  |  |  |  |  |  | ∞ |  | x |  |
| 10 |  |  |  |  | 35/20 | ∞ | ∞ | ∞ | ∞ | x |

По матрице построен граф предпочтений (Рисунок 2.1).



**Рисунок 2.1 – Вид графа предпочтений**

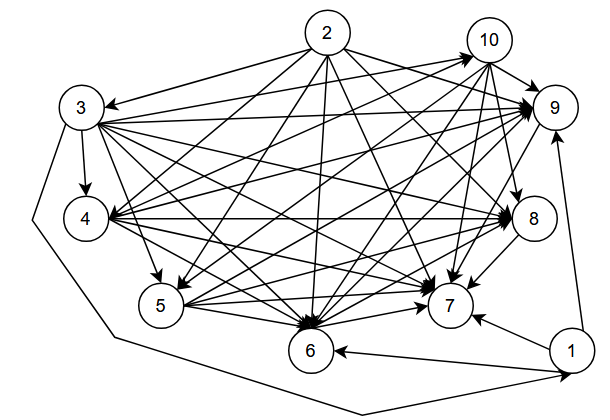
Назначен порог отбора предпочтений C = 1.5 (это соответствует тому, что учитываются только более сильные связи в графе).

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

*Таблица 2.4* **–** *Матрица предпочтений проектов, при пороге С=* 1.5

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 1 | x |  |  |  | 30/15 | ∞ | ∞ |  | ∞ |  |
| 2 |  | x | 55/25 | 60/30 | 55/15 | 65/10 | ∞ | ∞ | ∞ | 75/15 |
| 3 | 45/30 |  | x | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ |
| 4 |  |  |  | x |  | 50/10 | ∞ | 70/10 | 90/15 | 50/15 |
| 5 |  |  |  |  | x | 25/15 | ∞ | 35/20 | ∞ |  |
| 6 |  |  |  |  |  | x | ∞ | 45/30 | ∞ |  |
| 7 |  |  |  |  |  |  | x |  |  |  |
| 8 |  |  |  |  |  |  | ∞ | x |  |  |
| 9 |  |  |  |  |  |  | ∞ |  | x |  |
| 10 |  |  |  |  | 35/20 | 35/10 | ∞ | ∞ | ∞ | x |

По этой матрице построен граф предпочтений (Рисунок 2.2).

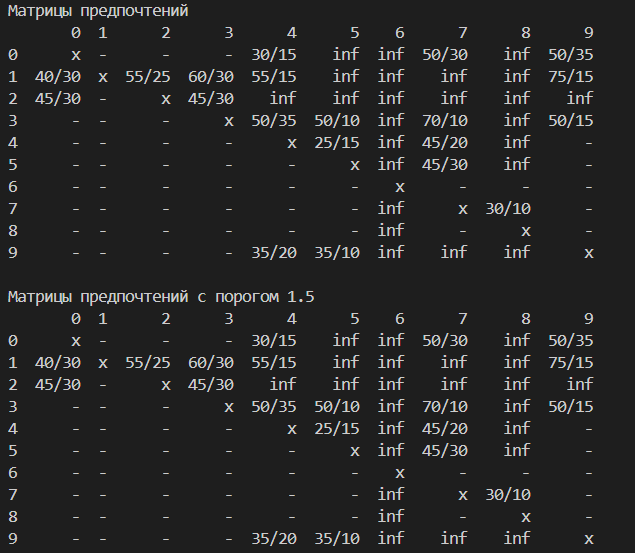


**Рисунок 2.2 – Вид графа предпочтений для случая порога принятия решений C =** 1.5

Петель в графе нет, при этом граф остался целостным.

Решение говорит что лучший вариант — 2. На втором месте — 3. На третьем — 10 вариант. На четвертом — 4. На пятом — 5 и 1 варианты. На шестом — 6. На седьмом — 8 и 9 варианты. На восьмом — 7 вариант.

## **2.3 Результат работы программы**



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

## **2.4 Вывод по методу Электра II**

В данной работе был изучен метод Электро II и использован на примере выбора самого лучшего космического корабля. Алгоритм был реализован как вручную так и программно. Данный алгоритм прост а алгоритмах, но имеет большое количество операций и сравнений. Также если выбрать слишком высокий порог, граф предпочтений превратится в лес, что увеличит количество оптимальных вариантов.

3 МЕТОД АНАЛИЗА ИЕРАРХИЙ

Метод анализа иерархий (Аналитический иерархический процесс, AHP) является одним из наиболее распространённых методов принятия решений в условиях многокритериальности. Он был разработан американским математиком Томасом Саати и основан на структурировании сложной задачи в виде иерархии целей, критериев и альтернатив. Основная идея метода заключается в попарном сравнении элементов на каждом уровне иерархии с целью определения их относительной важности. Оценки проводятся с использованием экспертных суждений, что позволяет учитывать как количественные, так и качественные характеристики. Результатом метода является взвешенная оценка альтернатив и выбор наилучшего варианта. Благодаря своей гибкости и наглядности, метод широко применяется в экономике, управлении, проектировании и других сферах.

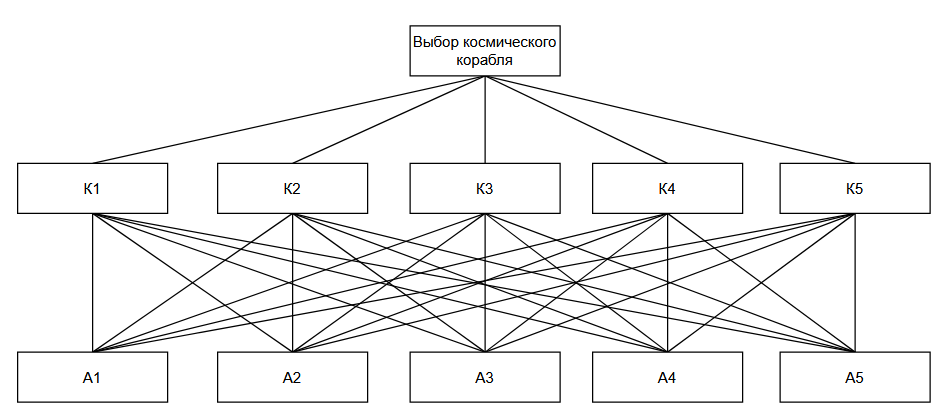
## **3.1 Постановка задачи**

Задача практической работы: выбрать лучший космический корабль.

## **3.2 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 – Цена;

К 2 – Скорость;

К 3 – Время входа в гиперпространство;

К 4 – Кол-во орудий;

К 5 – Мощность щитов.

Альтернативы:

А 1 – TIE Fighter;

А 2 – S-100;

А 3 – CR90;

А 4 – FT-6;

А 5 – S-13.

## **3.3 Установка приоритетов критериев**

Таблица данных:

*Таблица.3.1.1 – Шкала относительной важности.*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Космический корабль** | **Цена**  **(кредиты) (-)** | **Скорость**  **(км/ч) (+)** | **Время входа в гиперпространство**  **(сек) (-)** | **Количество орудий**  **(шт) (+)** | **Мощность щитов**  **(Вт) (+)** |
| 1 | TIE Fighter | 20000 | 5000 | 3.0 | 2 | 100 |
| 2 | S-100 | 21000 | 4800 | 3.1 | 3 | 150 |
| 3 | CR90 | 25000 | 4600 | 3.5 | 2 | 130 |
| 4 | FT-6 | 35000 | 4400 | 4.5 | 2 | 100 |
| 5 | S-13 | 33000 | 4600 | 4.1 | 2 | 105 |

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 3.1.2).

*Таблица.3,1.2 – Шкала относительной важности.*

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

Шкала содержит соответствующие обратные значения.

## **3.4 Синтез приоритетов**

После построения иерархии и определения величин парных субъективных суждений следует этап, на котором иерархическая декомпозиция и относительные суждения объединяются для получения осмысленного решения многокритериальной задачи принятия решений. Из групп парных сравнений формируется набор локальных критериев, которые выражают относительное влияние элементов на элемент, расположенный на уровне выше. Составлена обратно симметричная матрица для парного сравнения критериев (Таблица 2).

*Таблица 3.2 – Матрица парного сравнения критериев.*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Цель | К 1 | К 2 | К 3 | К 4 | К 5 | Vi | W2i |
| К 1 | 1 | 1/7 | 3 | 2 | 2 | 1,113 | 0,170 |
| К 2 | 7 | 1 | 3 | 5 | 5 | 3,500 | 0,534 |
| К 3 | 1/3 | 1/3 | 1 | 1 | 1 | 0,644 | 0,098 |
| К 4 | 1/2 | 1/5 | 1 | 1 | 3 | 0,786 | 0,120 |
| К 5 | 1/2 | 1/5 | 1 | 1/3 | 1 | 0,506 | 0,077 |
| ∑Vi | | | | | | 6,549 |

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

V1=(1 х 1/7 х 3 х 2 х 2)1/5 = 1,113;

Строка № 2

V2=(7 х 1 х 3 х 5 х 5 )1/5 = 3,500;

Строка № 3

V3=( 1/3 х 1/3 х 1 х 1 х 1)1/5 = 0,644;

Строка № 4

V4=( 1/2 х 1/5 х 1 х 1 х 3)1/5 = 0,786;

Строка № 5

V5=( 1/2 х 1/5 х 1 х 1/3 х 1)1/5 = 0,506.

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

∑Vi = V1 + V2 + V3 + V4 + V5 = 1,113 + 3,500 + 0,644 + 0,786 + 0,506 = 6,549.

Найдена важность приоритетов W2i, для этого каждое из чисел Vi разделено на ∑Vi.

***Строка № 1***

W21= 1,113 / ∑Vi =Y21 = 0,170;

Строка № 2

W22= 3,500 / ∑Vi = Y22 = 0,534;

***Строка № 3***

W23= 0,644 / ∑Vi = Y23 = 0,098;

***Строка № 4***

W24= 0,786 / ∑Vi = Y24 = 0,120;

***Строка № 5***

W25= 0,506 / ∑Vi = Y25 = 0,077.

В результате получен вектор приоритетов:

W2i = (0,170 = Y21; 0,534 = Y22; 0,098 = Y23; 0,120 = Y24; 0,077 = Y25), где индекс 2 означает, что вектор приоритетов относится ко второму уровню иерархии.

К 1 – цена(Таблица 3);

*Таблица 3.3 – Матрица сравнения по критерию 1.*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| К1 | А1 | А2 | А3 | А4 | А5 | VК1Y | W3К1Y |
| А1 | 1 | 3 | 3 | 5 | 5 | 2,954 | 0,436 |
| А2 | 1/3 | 1 | 3 | 5 | 5 | 1,904 | 0,281 |
| А3 | 1/3 | 1/3 | 1 | 5 | 3 | 1,107 | 0,164 |
| А4 | 1/5 | 1/5 | 1/5 | 1 | 1 | 0,380 | 0,056 |
| А5 | 1/5 | 1/5 | 1/3 | 1 | 1 | 0,422 | 0,062 |
| ∑VК1Y | | | | | | 6,767 |

Определена относительная ценность каждого элемента.

Строка № 1

VК11=(1 x 3 x 3 x 5 x 5)1/5= 2,954;

Строка № 2

VК12=( 1/3 x 1 x 3 x 5 x 5)1/5= 1,904;

Строка № 3

VК13=( 1/3 x 1/3 x 1 x 5 x 3)1/5= 1,107;

Строка № 4

VК14=( 1/5 x 1/5 x 1/5 x 1 x 1)1/5= 0,380;

Строка № 5

VК15=( 1/5 x 1/5 x 1/3 x 1 x 1)1/5= 0,422.

Проведена нормализация полученных чисел. Для этого определен нормирующий коэффициент ∑VK1Y.

∑VК1Y=VК11+VК12+VК13+VК14+VК15=2,954+1,904+1,107+0,380+0,422 = 6,767.

Найдена важность приоритетов W3К1Y, для этого каждое из чисел VK1Y разделено на ∑VK1Y.

Строка № 1

W3К11= 2,954 / ∑Vi = Y311 = 0,436

Строка № 2

W3К12= 1,904 / ∑Vi = Y312 = 0,281

Строка № 3

W3К13= 1,107 / ∑Vi = Y313 = 0,164

Строка № 4

W3К14= 0,380 / ∑Vi = Y314 = 0,056

Строка № 5

W3К15= 0,422 / ∑Vi = Y315 = 0,062

В результате получаем вектор приоритетов:

W3К1Y = (Y311 = 0,436; Y312 = 0,281; Y313 = 0,164; Y314 = 0,056; Y315 = 0,062),

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

К 2 – скорость(Таблица 4);

*Таблица 3.4. – Матрица сравнения по критерию 2.*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| К2 | А1 | А2 | А3 | А4 | А5 | VК2Y | W3К2Y |
| А1 | 1 | 1/3 | 1/3 | 1/5 | 1/3 | 0,375 | 0,064 |
| А2 | 3 | 1 | 1/3 | 1/3 | 1/3 | 0,644 | 0,110 |
| А3 | 3 | 3 | 1 | 1 | 1 | 1,552 | 0,266 |
| А4 | 5 | 3 | 1 | 1 | 1 | 1,719 | 0,294 |
| А5 | 3 | 3 | 1 | 1 | 1 | 1,552 | 0,266 |
| ∑VК2Y | | | | | | 5,842 |  |

Определена относительная ценность каждого элемента.

Строка № 1

VК21=( 1 x 1/3 x 1/3 x 1/5 x 1/3)1/5= 0,375;

Строка № 2

VК22=( 3 x 1 x 1/3 x 1/3 x 1/3)1/5= 0,644;

Строка № 3

VК23=(3 x 3 x 1 x 1 x 1)1/5= 1,552;

Строка № 4

VК24=( 5 x 3 x 1 x 1 x 1)1/5= 1,719;

Строка № 5

VК25=( 3 x 3 x 1 x 1 x 1)1/5= 1,552.

Проведена нормализация полученных чисел. Для этого определен нормирующий коэффициент ∑VK2Y.

∑VК2Y = VК21 + VК22 + VК23 + VК24 + VК25 = 0,375+0,644+1,552+1,719+1,552 = 5,842.

Найдена важность приоритетов W3К2Y, для этого каждое из чисел VK2Y разделено на ∑VK2Y.

Строка № 1

W3К21= 0,375 / ∑Vi = 0,064;

Строка № 2

W3К22= 0,644 / ∑Vi = 0,110;

Строка № 3

W3К23= 1,552 / ∑Vi = 0,266;

Строка № 4

W3К24= 1,719 / ∑Vi = 0,294;

Строка № 5

W3К25= 1,552 / ∑Vi = 0,266.

В результате получаем вектор приоритетов:

W3К2Y = (Y321 = 0,064; Y322 =0,110; Y323 =0,266; Y324 =0,294; Y325 =0,266),

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

К 3 – время входа в гиперпространство (Таблица 5);

*Таблица 3.5 – Матрица сравнения по критерию 5.*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| К3 | А1 | А2 | А3 | А4 | А5 | VК3Y | W3К3Y |
| А1 | 1 | 1 | 3 | 5 | 5 | 2,371 | 0,372 |
| А2 | 1 | 1 | 1 | 5 | 5 | 1,904 | 0,299 |
| А3 | 1/3 | 1 | 1 | 3 | 3 | 1,246 | 0,195 |
| А4 | 1/5 | 1/5 | 1/3 | 1 | 1 | 0,422 | 0,066 |
| А5 | 1/5 | 1/5 | 1/3 | 1 | 1 | 0,422 | 0,066 |
| VК35 | | | | | | 6,365 |  |

Определена относительная ценность каждого элемента.

Строка № 1

VК31=( 1 x 1 x 3 x 5 x 5)1/5= 2,371;

Строка № 2

VК32=( 1 x 1 x 1 x 5 x 5)1/5= 1,904;

Строка № 3

VК33=( 1/3 x 1 x 1 x 3 x 3)1/5= 1,246;

Строка № 4

VК34=( 1/5 x 1/5 x 1/3 x 1 x 1)1/5= 0,422;

Строка № 5

VК35=( 1/5 x 1/5 x 1/3 x 1 x 1)1/5= 0,422.

Проведена нормализация полученных чисел. Для этого определен нормирующий коэффициент ∑VK3Y.

∑VК3Y = VК31 + VК32 + VК33 + VК34 + VК35 = 2,371+1,904+1,246+0,422+0,422 = 6,365.

Найдена важность приоритетов W3К2Y, для этого каждое из чисел VK2Y разделено на ∑VK2Y.

Строка № 1

W3К31= 2,371 / ∑Vi = 0,372;

Строка № 2

W3К32= 1,904 / ∑Vi = 0,299;

Строка № 3

W3К33= 1,246 / ∑Vi = 0,195;

Строка № 4

W3К34= 0,422 / ∑Vi = 0,066;

Строка № 5

W3К35= 0,422 / ∑Vi = 0,066.

В результате получаем вектор приоритетов:

W3К3Y = (Y331=0,372; Y332=0,299; Y333=0,195; Y334=0,066; Y335=0,066),

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

К 4 – кол-во орудий (Таблица 6);

*Таблица 3.6 – Матрица сравнения по критерию 4.*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| К4 | А1 | А2 | А3 | А4 | А5 | VК4Y | W3К4Y |
| А1 | 1 | 3 | 1 | 1 | 1 | 1,246 | 0,312 |
| А2 | 1/3 | 1 | 1/3 | 1/3 | 1/3 | 0,333 | 0,083 |
| А3 | 1 | 1/3 | 1 | 1 | 1 | 0,803 | 0,201 |
| А4 | 1 | 1/3 | 1 | 1 | 1 | 0,803 | 0,201 |
| А5 | 1 | 1/3 | 1 | 1 | 1 | 0,803 | 0,201 |
| ∑VК4Y | | | | | | 3,988 |

Определена относительная ценность каждого элемента.

Строка № 1

VК41=( 1 x 3 x 1 x 1 x 1)1/5 = 1,246;

Строка № 2

VК42=( 1/3 x 1/3 x 1/3 x 1/3 x 1/3)1/5 =0,333 ;

Строка № 3

VК43=( 1 x 1/3 x 1 x 1 x 1)1/5 = 0,803;

Строка № 4

VК44=( 1 x 1/3 x 1 x 1 x 1)1/5 = 0,803;

Строка № 5

VК45=( 1 x 1/3 x 1 x 1 x 1)1/5 = 0,803.

Проведена нормализация полученных чисел. Для этого определен нормирующий коэффициент ∑VK4Y.

∑VК4Y = VК41 + VК42 + VК43 + VК44 + VК45 =1,246+0,333+0,803+0,803+0,803 = 3,988 .

Найдена важность приоритетов W3К4Y, для этого каждое из чисел VK4Y разделено на ∑VK4Y.

Строка № 1

W3К41= 1,246 / ∑Vi =0,312 ;

Строка № 2

W3К42= 0,333 / ∑Vi = 0,083;

Строка № 3

W3К43= 0,803 / ∑Vi = 0,201;

Строка № 4

W3К44= 0,803 / ∑Vi = 0,201;

Строка № 5

W3К45= 0,803 / ∑Vi = 0,201.

В результате получаем вектор приоритетов:

W3К4Y = (Y341=0,312; Y342=0,083; Y343=0,201; Y344=0,201; Y345=0,201),

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

К 5 – мощность шитов (Таблица 7).

*Таблица 3.7 – Матрица сравнения по критерию 5.*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| К5 | А1 | А2 | А3 | А4 | А5 | VК5Y | W3К5Y |
| А1 | 1 | 5 | 5 | 3 | 3 | 2,954 | 0,454 |
| А2 | 1/5 | 1 | 3 | 5 | 5 | 1,719 | 0,264 |
| А3 | 1/5 | 1/3 | 1 | 3 | 3 | 0,902 | 0,138 |
| А4 | 1/3 | 1/5 | 1/3 | 1 | 1 | 0,467 | 0,072 |
| А5 | 1/3 | 1/5 | 1/3 | 1 | 1 | 0,467 | 0,072 |
| ∑VК5Y | | | | | | 6,509 |

Определена относительная ценность каждого элемента.

Строка № 1

VК51=( 1 x 5 x 5 x 3 x 3)1/5 = 2,954;

Строка № 2

VК52=( 1/5 x 1 x 3 x 5 x 5)1/5 = 1,719;

Строка № 3

VК53=( 1/5 x 1/3 x 1 x 3 x 3)1/5 = 0,902;

Строка № 4

VК54=( 1/3 x 1/5 x 1/3 x 1 x 1)1/5 = 0,467;

Строка № 5

VК55=( 1/3 x 1/5 x 1/3 x 1 x 1)1/5 = 0,467.

Проведена нормализация полученных чисел. Для этого определен нормирующий коэффициент ∑VK5Y.

∑VК5Y = VК51 + VК52 + VК53 + VК54 + VК55 = 2,954+1,719+0,902+0,467+0,467=6,509.

Найдена важность приоритетов W3К5Y, для этого каждое из чисел VK5Y разделено на ∑VK5Y.

Строка № 1

W3К51= 2,954 / ∑Vi = 0,454;

Строка № 2

W3К52= 1,719 / ∑Vi = 0,264;

Строка № 3

W3К53= 0,902 / ∑Vi = 0,138;

Строка № 4

W3К54= 0,467 / ∑Vi = 0,072;

Строка № 5

W3К55= 0,467 / ∑Vi = 0,072.

В результате получаем вектор приоритетов:

W3К5Y = (Y351=0,454; Y352=0,264; Y353=0,138; Y354=0,072; Y355=0,072),

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

## **3.5 Согласованность локальных приоритетов**

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

Метод анализа иерархий дает возможность провести такую оценку.

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

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

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

Определены индекс согласованности и отношение согласованности для матрицы «Выбор космического коробля» (Таблица 8).

*Таблица 3.8 – Матрица «Выбор лучшего отеля».*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Цель | К 1 | К 2 | К 3 | К 4 | К 5 | W2i |
| К 1 | 1 | 1/7 | 3 | 2 | 2 | 0,170 |
| К 2 | 7 | 1 | 3 | 5 | 5 | 0,534 |
| К 3 | 1/3 | 1/3 | 1 | 1 | 1 | 0,098 |
| К 4 | 1/2 | 1/5 | 1 | 1 | 3 | 0,120 |
| К 5 | 1/2 | 1/5 | 1 | 1/3 | 1 | 0,077 |

Определена сумма каждого столбца матрицы суждений.

S1 = 1 + 7 + 1/3 + 1/2 + 1/2 = 9,3;

S2 = 1/7 + 1 + 1/3 + 1/5 + 1/5 = 1,876;

S3 = 3 + 3 + 1 + 1 + 1 = 9;

S4 = 2 + 5 + 1 + 1 + 1/3 = 9,333;

S5 = 2 + 5 + 1 + 3 + 1 = 12.

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.

Р1 = S1 х W21 = 9,3 x 0,170 = 1,581;

Р2 = S2 х W22 = 1,876 x 0,534 = 1,001;

Р3 = S3 х W23 = 9 x 0,098 = 0,882;

Р4 = S4 х W24 = 9,333 x 0,120 = 1,120;

Р5 = S5 х W25 = 12 x 0,077 = 0,824.

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

λmax = Р1 + Р2 + Р3 + Р4 + Р5 = 5,405.

Отклонение от согласованности выражается индексом согласованности.

ИС = (λmax - n)/(n - 1) = 0,405/4 = 0,101.

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

ОС = ИС/СИ = 0,101/1,12 = 0,090.

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

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

*Таблица 3.9 – Матрица сравнения по критерию 1.*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| К1 | А1 | А2 | А3 | А4 | А5 | W3К1Y |
| А1 | 1 | 3 | 3 | 5 | 5 | 0,436 |
| А2 | 1/3 | 1 | 3 | 5 | 5 | 0,281 |
| А3 | 1/3 | 1/3 | 1 | 5 | 3 | 0,164 |
| А4 | 1/5 | 1/5 | 1/5 | 1 | 1 | 0,056 |
| А5 | 1/5 | 1/5 | 1/3 | 1 | 1 | 0,062 |

Определяется сумма каждого столбца матрицы суждений.

S1К1 = 2.666;

S2 К1 = 5.333 ;

S3 К1 = 7.833;

S4 К1 = 17;

S5 К1 = 15.

Затем полученный результат умножен на компоненту нормализованного вектора приоритетов.

Р1 К1 = S1 х W3К11 = 1.133;

Р2 К1 = S2 х W3К12 = 1.489;

Р3 К1 = S3 х W3К13 = 1.284;

Р4 К1 = S1 х W3К14 = 0.652;

Р5 К1 = S1 х W3К15 = 0.93.

Найдена пропорциональность предпочтений.

λmax К1 = Р1К1 + Р2К1 + Р3К1 + Р4К1 + Р5К1 = 5.448.

Отклонение от согласованности выражается индексом согласованности.

ИС К1 = (λmax К1 - n)/(n - 1) = 0.112.

Найдено отношение согласованности ОС.

ОС К1 = ИС/СИ = 0.1.

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

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

*Таблица 3.10 – Матрица сравнения по критерию 2.*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| К2 | А1 | А2 | А3 | А4 | А5 | W3К2Y |
| А1 | 1 | 1/3 | 1/3 | 1/5 | 1/3 | 0,064 |
| А2 | 3 | 1 | 1/3 | 1/3 | 1/3 | 0,110 |
| А3 | 3 | 3 | 1 | 1 | 1 | 0,266 |
| А4 | 5 | 3 | 1 | 1 | 1 | 0,294 |
| А5 | 3 | 3 | 1 | 1 | 1 | 0,266 |

Определена сумма каждого столбца матрицы суждений.

S1К2 =17;

S2 К2 =10.3;

S3 К2 =3.6;

S4 К2 =3.8;

S5 К2 =3.6.

Затем полученный результат умножен на компоненту нормализованного вектора приоритетов.

Р1 К2 = S1 х W3 К21 =1.088;

Р2 К2 = S2 х W3 К22 =1.133;

Р3 К2 = S3 х W3 К23 =0.957;

Р4 К2 = S4 х W3 К24 =1.117;

Р5 К2 = S5 х W3 К25 =0.957.

Найдена пропорциональность предпочтений.

λmax К2 = Р1К2 + Р2К2 + Р3К2 + Р4К2 + Р5К2 = 5.252.

Отклонение от согласованности выражается индексом согласованности.

ИС К2 = (λmax К2 - n)/(n - 1) = 0.063.

Найдено отношение согласованности ОС.

ОС К2 = ИС/СИ =0.056.

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

Определены индекс согласованности и отношение согласованности для матрицы К 3 – время входа в гиперпространство (Таблица 3.11).

*Таблица 3.11 – Матрица сравнения по критерию 3.*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| К3 | А1 | А2 | А3 | А4 | А5 | W3К3Y |
| А1 | 1 | 1 | 3 | 5 | 5 | 0,372 |
| А2 | 1 | 1 | 1 | 5 | 5 | 0,299 |
| А3 | 1/3 | 1 | 1 | 3 | 3 | 0,195 |
| А4 | 1/5 | 1/5 | 1/3 | 1 | 1 | 0,066 |
| А5 | 1/5 | 1/5 | 1/3 | 1 | 1 | 0,066 |

Определена сумма каждого столбца матрицы суждений.

S1К3 =3.3;

S2 К3 =4;

S3 К3 =5.6;

S4 К3 =15;

S5 К3 =15.

Затем полученный результат умножен на компоненту нормализованного вектора приоритетов.

Р1 К3 = S1 х W3 К31 =1.227;

Р2 К3 = S2 х W3 К32 =1.196;

Р3 К3 = S3 х W3 К33 =1.092;

Р4 К3 = S4 х W3 К34 =0.99;

Р5 К3 = S5 х W3 К35 =0.99.

Найдем пропорциональность предпочтений.

λmax К3 = Р1К3 + Р2К3 + Р3К3 + Р4К3 + Р5К3 =5.495.

Отклонение от согласованности выражается индексом согласованности.

ИС К3 = (λmax К3 - n)/(n - 1) =0.123.

Найдено отношение согласованности ОС.

ОС К3 = ИС/СИ =0.10.

Значение ОС меньше или равное 0.10 считается приемлемым, значит матрица К 3 ( время входа в гиперпространство) согласована.

Определены индекс согласованности и отношение согласованности для матрицы К 4 – кол-во орудий (Таблица 3.12).

*Таблица 3.12 – Матрица сравнения по критерию 4.*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| К4 | А1 | А2 | А3 | А4 | А5 | W3К4Y |
| А1 | 1 | 3 | 1 | 1 | 1 | 0,312 |
| А2 | 1/3 | 1 | 1/3 | 1/3 | 1/3 | 0,083 |
| А3 | 1 | 1/3 | 1 | 1 | 1 | 0,201 |
| А4 | 1 | 1/3 | 1 | 1 | 1 | 0,201 |
| А5 | 1 | 1/3 | 1 | 1 | 1 | 0,201 |

Определена сумма каждого столбца матрицы суждений.

S1К4 =4.3;

S2К4 =5;

S3К4 =4.3;

S4К4 =4.3;

S5К4 =4.3.

Затем полученный результат умножен на компоненту нормализованного вектора приоритетов.

Р1К4 = S1 х W3 К41 =1.341;

Р2К4 = S2 х W3 К42 =0.415;

Р3К4 = S3 х W3 К43 =0.864;

Р4К4 = S4 х W3 К44 =0.864;

Р5К4 = S5 х W3 К45 =0.864.

Найдена пропорциональность предпочтений.

λmax К4 = Р1К4 + Р2К4 + Р3К4 + Р4К4 + Р5К4 = 4.348.

Отклонение от согласованности выражается индексом согласованности.

ИС К4 = (λmax К4 - n)/(n - 1) =0.087.

Найдено отношение согласованности ОС.

ОС К4 = ИС/СИ =0.077.

Значение ОС меньше или равное 0,10 считается приемлемым, значит матрица К 4 (кол-во орудий) согласована.

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

*Таблица 3.13 – Матрица сравнения по критерию 5.*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| К5 | А1 | А2 | А3 | А4 | А5 | W3К5Y |
| А1 | 1 | 5 | 5 | 3 | 3 | 0,454 |
| А2 | 1/5 | 1 | 3 | 5 | 5 | 0,264 |
| А3 | 1/5 | 1/3 | 1 | 3 | 3 | 0,138 |
| А4 | 1/3 | 1/5 | 1/3 | 1 | 1 | 0,072 |
| А5 | 1/3 | 1/5 | 1/3 | 1 | 1 | 0,072 |

Определена сумма каждого столбца матрицы суждений.

S1К5 =2.6;

S2К5 =7.3;

S3К5 =9.6;

S4К5 =13;

S5К5 =13.

Затем полученный результат умножен на компоненту нормализованного вектора приоритетов.

Р1К5 = S1 х W3 К41 =0.180;

Р2К5 = S2 х W3 К42 =1.927;

Р3К5 = S3 х W3 К43 =1.296;

Р4К5 = S1 х W3 К44 =0.936;

Р5К5 = S1 х W3 К45 =0.936.

Найдена пропорциональность предпочтений.

λmax К5 = Р1К5 + Р2К5 + Р3К5 + Р4К5 + Р5К5 =5.275.

Отклонение от согласованности выражается индексом согласованности.

ИС К5 = (λmax К5 - n)/(n - 1) =0.068.

Найдено отношение согласованности ОС.

ОС К5 = ИС/СИ =0.061.

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

## **3.6 Синтез альтернатив**

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

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

W2i = (0,170 = Y21; 0,534 = Y22; 0,098 = Y23; 0,120 = Y24; 0,077 = Y25);

W3К1Y = (Y311 = 0,436; Y312 = 0,281; Y313 = 0,164; Y314 = 0,056; Y315 = 0,062);

W3К2Y = (Y321 = 0,064; Y322 =0,110; Y323 =0,266; Y324 =0,294; Y325 =0,266);

W3К3Y = (Y331=0,372; Y332=0,299; Y333=0,195; Y334=0,066; Y335=0,066);

W3К4Y = (Y341=0,312; Y342=0,083; Y343=0,201; Y344=0,201; Y345=0,201);

W3К5Y = (Y351=0,454; Y352=0,264; Y353=0,138; Y354=0,072; Y355=0,072).

Приоритеты альтернатив получены следующим образом:

W1 = W21 х W3К11 + W22 х W3К21 + W23 х W3К31 + W24 х W3К41 + W25 х W3К51 =0,216.

W2 = W21 х W3К12 + W22 х W3К22 + W23 х W3К32 + W24 х W3К42 + W25 х W3К52 =0,162.

W3 = W21 х W3К13 + W22 х W3К23 + W23 х W3К33 + W24 х W3К43 + W25 х W3К53 =0,221

W4 = W21 х W3К14 + W22 х W3К24 + W23 х W3К34 + W24 х W3К44 + W25 х W3К54 =0,119.

W5 = W21 х W3К15 + W22 х W3К25 + W23 х W3К35 + W24 х W3К45 + W25 х W3К55 =0,201.

Таким образом, приоритеты альтернатив равны:

альтернатива А1 (название) - W1 приоритет равен =0,216;

альтернатива А2 (название)- W2 приоритет равен =0,162;

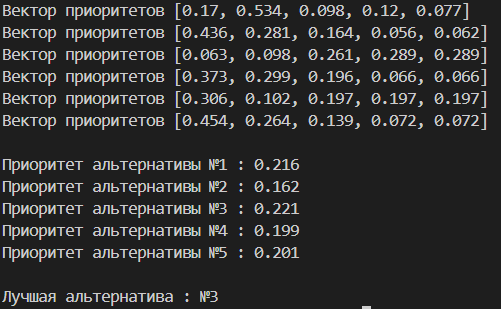
альтернатива А3 (название) - W3 приоритет равен =0,221;

альтернатива А4 (название) – W4 приоритет равен =0,119;

альтернатива А5 (название) - W5 приоритет равен =0,201.

Самая оптимальная альтернатива имеет имеет наибольший приоритет, это альтернатива №3.

## **3.7 Результаты работы программы**



**Рисунок 3.2 – Вывод программы**

## **Вывод по методу МАИ**

В ходе данной работы изучен метод анализа иерархий, проведён его ручной расчёт для пяти критериев и пяти альтернатив. Преимуществом метода является гарантированное получение единственного оптимального решения, а недостатком является требование соблюдать согласованность матриц приоритетов, из-за чего необходимо проводить повторные расчёты в случае, если матрица не согласована.

**4 ГРАФИЧЕСКИЙ МЕТОД**

Линейное программирование является важным разделом математического программирования, позволяющим находить оптимальные решения задач, связанных с распределением ограниченных ресурсов. Одним из наиболее наглядных и интуитивно понятных методов решения задач линейного программирования является графический метод.

Графический метод применяется в случаях, когда число переменных в задаче не превышает двух, поскольку его суть заключается в геометрической интерпретации системы ограничений и нахождении оптимального значения целевой функции на построенной области допустимых решений. Метод позволяет не только получить решение, но и визуально проанализировать, какие ограничения оказывают влияние на результат, а также оценить чувствительность оптимального решения к изменению параметров задачи.

Область применения графического метода охватывает широкий спектр задач, связанных с управлением ресурсами, планированием производства, оптимизацией затрат и максимизацией прибыли. В частности, он используется для решения задач оптимального использования сырья в производстве, распределения финансовых средств, планирования логистических маршрутов и управления запасами.

**4.1 Постановка задачи**

Решить задачу линейного программирования с двумя переменными графическим методом.

**4.2 Данные индивидуального варианта**

**4.3 Подготовка данных**

В среде Microsoft Excel добавим 5 столбцов:

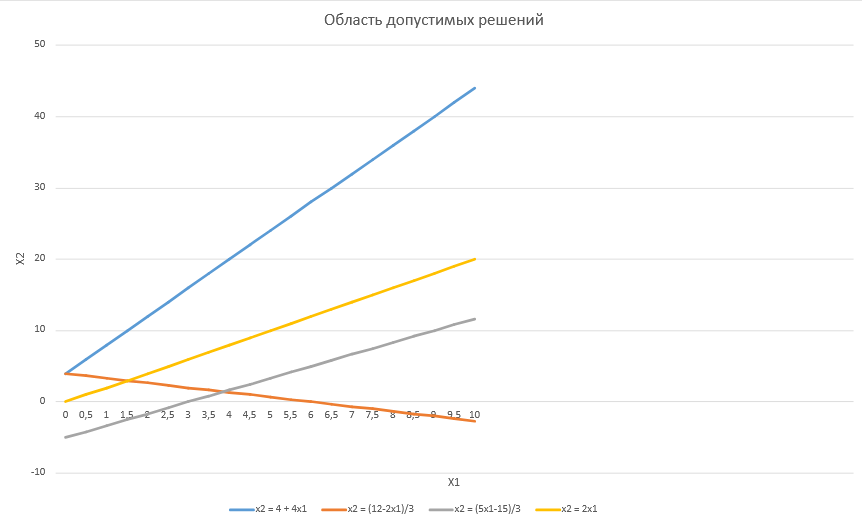
1. – значения от 0 до 10 с шагом 0,5;
2. – значения ограничения ;
3. – значения ограничения ;
4. – значения ограничения ;
5. – значения = 0.

*Таблица 4,1.1 – Данные для графика*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| x1 | x2= 4 + 4x1 | x2 = (12-2x1)/3 | x2=(5x1-15)/3 | x2=2x1 |
| 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 |
| 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 |
| 5,5 | 26 | 0,33 | 4,17 | 11 |
| 6 | 28 | 0 | 5 | 12 |
| 6,5 | 30 | -0,33 | 5,83 | 13 |
| 7 | 32 | -0,67 | 6,67 | 14 |
| 7,5 | 34 | -1 | 7,5 | 15 |
| 8 | 36 | -1,33 | 8,33 | 16 |
| 8,5 | 38 | -1,67 | 9,17 | 17 |
| 9 | 40 | -2 | 10 | 18 |
| 9,5 | 42 | -2,33 | 10,83 | 19 |
| 10 | 44 | -2,67 | 11,67 | 20 |

**4.4 Построение графика**

Выделим таблицу подготовленных данных и построим гладкий график. Произведем настройку шага координатной оси x1 и получим следующий график (Рисунок 4.1.1)

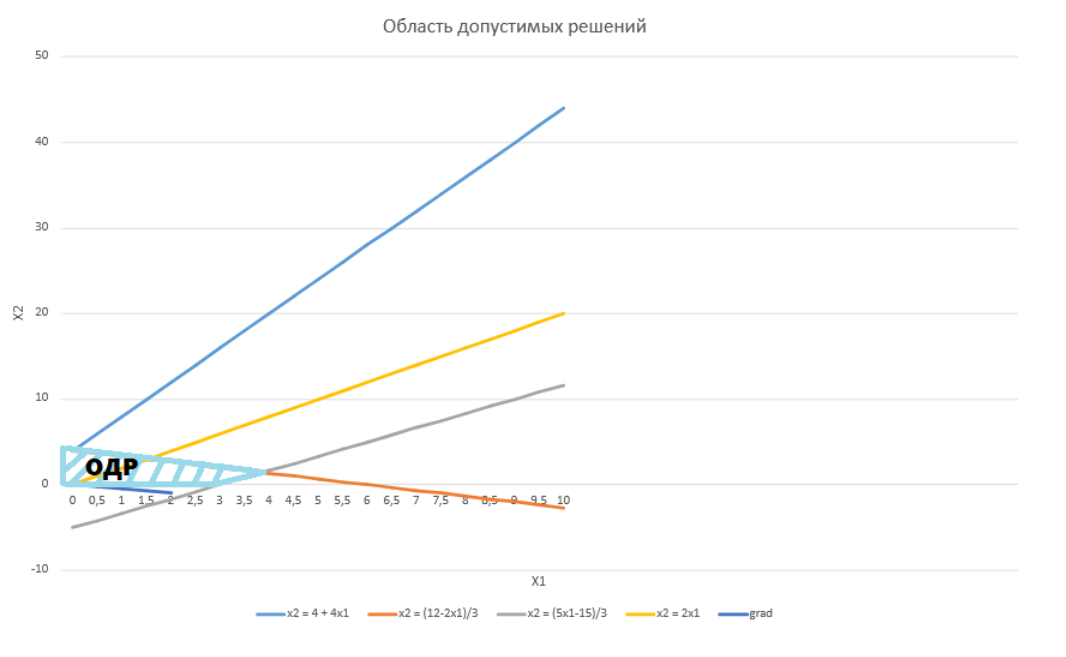


**Рисунок 4.1.1 – Построение графиков по данным**

**4.5 Выделение области допустимых решений**

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

В результате получим область допустимых решений, представленную на Рисунке 4.1.2.



**Рисунок 4.1.2 – Выделение области допустимых решений**

**4.6 Максимум функции**

Для нахождения максимума функции найдем её градиент по формуле 4.1.1:

(4.1.1)

Для нахождения минимума функции найдем её градиент по формуле 4.1.1:

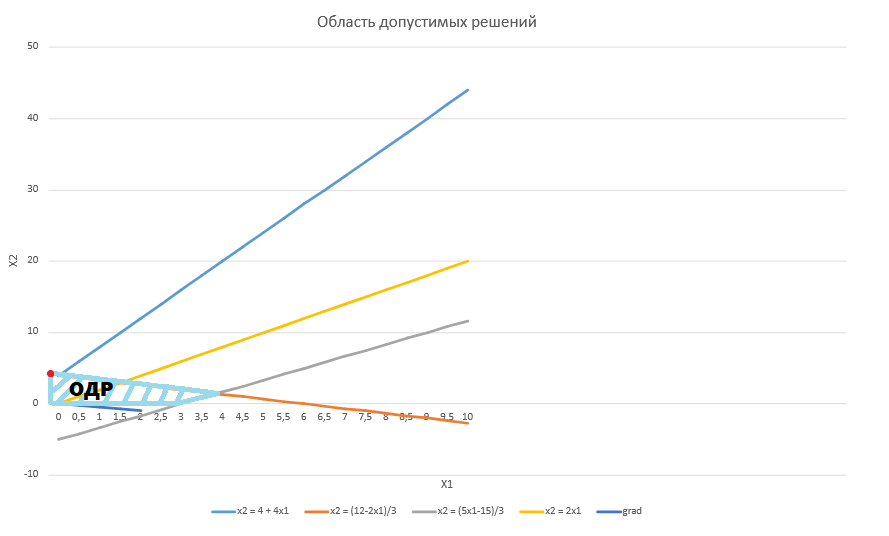
(4.1.2)

Градиент функции будет равен {-2,1}, а антиградиент функции будет равен {2,-1}. Изобразим эти вектора на графике (Рисунок 4.1.4).

Точка находится на пересечении двух прямых, поэтому найдём её значение при помощи системы уравнений:

Решив систему, получаем, что и

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



**Рисунок 4.1.4 – Точка максимума функции**

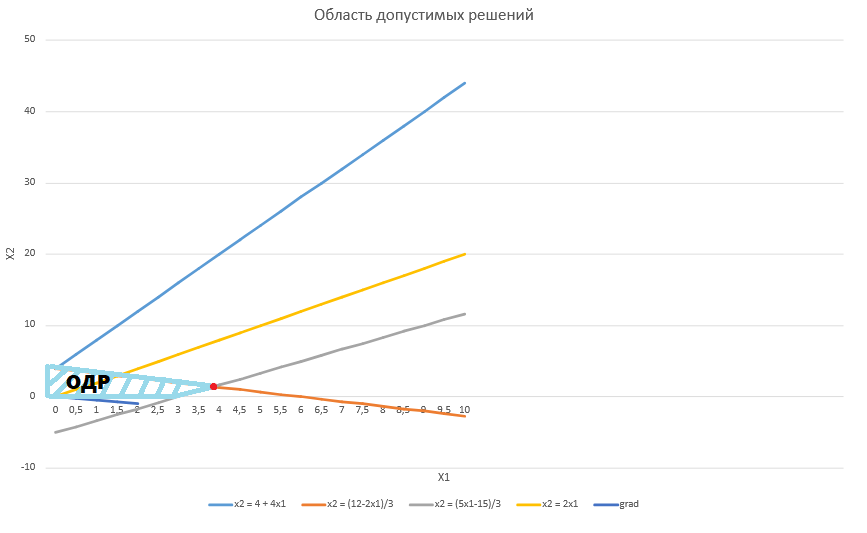
Найдем значение функции в точке максимума.

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

Получим значение равное F(x)max = 4.

**4.7 Минимум функции**

Для нахождения минимума функции будем перемещать прямую в сторону антиградиента. Отметим на графике найденную точку (Рисунок 4.1.5).



**Рисунок 4.1.5 – Точка минимума функции**

Найдем координаты точки минимума:

Точка находится на пересечении двух прямых, поэтому найдём её значение при помощи системы уравнений:

Решив систему, получаем, что и

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:

Получим результат F(x)min =

Ответ:

F(x)max = 4.

F(x)min =

**4.8 Вывод по графическому методу**

В ходе выполнения данной работы был изучен и применен графический метод решения задач линейного программирования. Данный метод позволяет наглядно представить процесс поиска оптимального решения, используя геометрическую интерпретацию ограничений и целевой функции.

Графический метод удобен для анализа задач с двумя переменными, так как позволяет не только находить оптимальное решение, но и визуально оценивать влияние различных ограничений на область допустимых решений. Кроме того, он дает возможность проводить анализ чувствительности, определяя, какие ограничения являются критическими для достижения оптимального результата.

Достоинства графического метода: наглядность и интуитивная понятность; простота вычислений для задач с двумя переменными; возможность визуального анализа области допустимых решений и влияния ограничений.

Недостатки графического метода: ограниченность применения (работает только при количестве переменных, не превышающем двух); сложность точного определения оптимального решения в случаях, когда оно находится в угловой точке с дробными значениями; неэффективность при большом числе ограничений, поскольку построение множества пересечений становится трудоемким.

Таким образом, графический метод является удобным инструментом для решения простых задач линейного программирования, однако для более сложных моделей, включающих большое количество переменных, необходимо использовать другие методы.

**5 СИМПЛЕКСНЫЙ МЕТОД**

Симплексный метод — это один из наиболее распространённых и эффективных алгоритмов решения задач линейного программирования. Он был разработан Джорджем Данцигом в 1947 году и используется для нахождения оптимального решения линейной целевой функции при наличии системы линейных ограничений.

Метод применяется в задачах, где требуется найти наилучшее распределение ограниченных ресурсов (времени, материалов, труда и т.д.) между различными видами деятельности с целью максимизации прибыли или минимизации затрат. Классические примеры применения симплексного метода — это задачи планирования производства, логистики, распределения ресурсов, управления проектами и другие экономико-математические задачи.

Суть метода заключается в последовательном переходе от одной допустимой вершины области решений к другой, улучшая значение целевой функции на каждом шаге, пока не будет достигнут оптимум. Симплексный метод является итерационным и хорошо масштабируется для задач с большим числом переменных и ограничений.

* 1. **Постановка задачи**

***Задание 8.*** Решить прямую ЗЛП с помощью симплексного метода и обратную с помощью теорем двойственности. Определить интервалы устойчивости.

***Задача.*** Для изготовления четырех видов продукции А, В, С и D используются три вида ресурсов I, II, III. Дальнейшее условие задачи в таблице П.8.

*Таблица П.5.8.* Исходные данные задачи.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Ресурсы | Нормы расхода сырья на  единицу продукции, ед. | | | | Запасы  ресурсов, ед. |
| А | В | С | D |
| 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 | 12 |  |

Требуется определить план выпуска продукции, при котором прибыль от ее реализации будет максимальной.

**5.2 Математическая модель задачи**

Пусть х1 – тип ресурса A, х2 –тип ресурса B, х3 –тип ресурса C, x4 – тип ресурса D. Прибыль от продажи шкафов составит 7.5х1 + 3х2 + 6х3+12x4, прибыль требуется максимизировать.

Ограничения задачи:

Таким образом, переходим к задаче линейного программирования:

Приведем задачу к канонической форме. Для этого в левые части ограничений вводим дополнительные переменные: х5 ≥ 0, х6 ≥ 0, х7 ≥ 0. Эти переменные выбираются так, чтобы они обращали неравенства в равенства.

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

Векторы 𝐴5, 𝐴6, 𝐴7 являются линейно независимыми единичными векторами 3х-мерного пространства и образуют базис этого пространства.

Поэтому за базисные переменные выбираем переменные 𝑥5, 𝑥6, 𝑥7. Небазисными переменными являются 𝑥1, 𝑥2, *х*3, *x*4. Разложение позволяет найти первое базисное допустимое решение.

Для этого свободные переменные 𝑥1, 𝑥2, *х*3, *x*4 приравниваем нулю. В результате получим разложение

Которому соответствует первоначальный опорный план

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

В левый столбец Таблицы 5.5.2 запишем переменные 𝑥5, 𝑥6, 𝑥7 образующие базис, в верхней строке – небазисные переменные 𝑥1, 𝑥2, *x*3, *x*4. В строке 𝑐j запишем коэффициенты целевой функции, соответствующие небазисным переменным с1 = 9, с2 = 3, c3 = 6, c4 = 12. В столбце запишем коэффициенты целевой функции, соответствующие базисным переменным Столбец, определяемый переменной 𝑥1, состоит из коэффициентов вектора . Аналогично, столбец, определяемый переменной 𝑥2, состоит из коэффициентов вектора . Аналогично, столбец, определяемый переменной 𝑥3, состоит из коэффициентов вектора . И столбец, определяемый переменной *х*4 состоит из коэффициентов вектора Крайний правый столбец заполняется элементами столбца , в нем же в результате вычислений получаем оптимальный план.

Заполнение f-строки (Таблица 5.1.3). Найдем относительные оценки ∆1, ∆2, ∆3, ∆4 и значение целевой функции 𝑄.

*Таблица 5.1.2 – Начальная симплекс-таблица задачи о максимальном доходе*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | сj | 7.5 | 3 | 6 | 12 |  |
|  |  | X1 | X2 | X3 | X4 |  |
| 0 | X5 | 2 | 1 | 0.5 | 4 | 3400 |
| 0 | X6 | 1 | 5 | 3 | 0 | 1200 |
| 0 | X7 | 3 | 0 | 6 | 1 | 3000 |
|  | f |  |  |  |  |  |
|  |  | Δ1 | Δ2 | Δ3 | Δ4 | Q |

*Таблица 5.1.3 – Заполнение f-строки*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | сj | 7.5 | 3 | 6 | 12 |  |  |
|  |  | X1 | X2 | X3 | X4 |  |  |
| 0 | X5 | 2 | 1 | 0.5 | 4 | 3400 | 3400 / 4 = 850 min |
| 0 | X6 | 1 | 5 | 3 | 0 | 1200 | не имеет смысла |
| 0 | X7 | 3 | 0 | 6 | 1 | 3000 | 3000 / 1 = 3000 |
|  | f | –7.5 | –3 | –6 | –12 | 0 |  |
|  |  | Δ1 | Δ2 | Δ3 | Δ4 | Q |  |

Для оптимальности опорного решения в задаче на максимум требуется выполнение неотрицательности всех относительных оценок ∆i ≥ 0. Так как оценки ∆1= −9, ∆2= −11, ∆3= −15 и ∆4= –12 в f-строке отрицательны, то это свидетельствуют о возможности улучшения полученного решения. Наибольшая по модулю отрицательная оценка ∆4= −12. В базис будет включена соответствующая ей небазисная переменная 𝑥4. Составим отношения свободных членов к положительным элементам разрешающего столбца. Данные отношения приведены справа от таблицы. Наименьшему частному соответствует строка с переменной 𝑥5. Эта переменная исключается из базиса. В Таблице 5.1.3 разрешающий столбец и разрешающая строка выделены. Разрешающим элементом является число 𝑎14 = 4.

Далее построим новую симплекс-таблицу. Ниже поэтапно демонстрируется процесс заполнения новой симплекс-таблицы (Таблицы 5.1.4 ).

*Таблица 5.1.4 – Новая симплекс-таблица*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | сj | 7.5 | 3 | 6 | 0 |  |
|  |  | X1 | X2 | X3 | X5 |  |
| 12 | X4 |  |  |  | 1/4 |  |
| 0 | X6 |  |  |  |  |  |
| 0 | X7 |  |  |  |  |  |
|  | f |  |  |  |  |  |
|  |  | Δ1 | Δ2 | Δ3 | Δ4 | Q |

В Таблице 5.1.4 переменные 𝑥4 и 𝑥5 меняются местами вместе с коэффициентами 𝑐𝑗. Разрешающий элемент заменяется на обратный. В Таблице 5.1.5 элементы разрешающей строки делятся на разрешающий элемент. Элементы разрешающего столбца делятся на разрешающий элемент и меняют знак.

*Таблица 5.1.5 – Симплекс преобразования*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | сj | 7.5 | 3 | 6 | 0 |  |
|  |  | X1 | X2 | X3 | X5 |  |
| 12 | X4 | 1/2 | 1/4 | 1/8 | 1/4 | 850 |
| 0 | X6 |  |  |  | 0 |  |
| 0 | X7 |  |  |  | -1/4 |  |
|  | f |  |  |  | 3 |  |
|  |  | Δ1 | Δ2 | Δ3 | Δ4 | Q |

*Таблица 5.1.6 – Итерация 0*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | сj | 7.5 | 3 | 6 | 0 |  |  |
|  |  | X1 | X2 | X3 | X5 |  |  |
| 12 | X4 | 1/2 | 1/4 | 1/8 | 1/4 | 850 | 850 / 1/8 = 6800 |
| 0 | X6 | 1 | 5 | 3 | 0 | 1200 | 1200 / 3 = 400 |
| 0 | X7 | 2.5 | -0.25 | 5.875 | -1/4 | 2150 | 2150 / 5.875 = 365 *min* |
|  | f | -1.5 | 0 | -4.5 | 3 |  |  |
|  |  | Δ1 | Δ2 | Δ3 | Δ4 | Q |  |

Остальные элементы (Таблица 5.1.6) рассчитываются по «правилу прямоугольника».

Базисное решение, которое дает последняя таблица

Это решение не является оптимальным, так как в f-строке имеются отрицательные оценки ∆1, ∆3.

*Таблица 5.5.7 – Итерация 1*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | сj | 7.5 | 3 | 0 | 0 |  |  |
|  |  | X1 | X2 | X7 | X5 |  |  |
| 12 | X4 | 0,45 | 0,25 | -0,02 | 0,25 | 804 | 804/0.25 = 3654 |
| 0 | X6 | -0,28 | 5,13 | -0,51 | 0,126 | 102 | 102/5 = 51 min |
| 6 | X3 | 0,42 | -0,042 | 0,17 | -0,042 | 365 | -1 |
|  | f | 0,41 | -0,19 | 0,77 | 2,811 | 11847 |  |
|  |  | Δ1 | Δ2 | Δ3 | Δ4 | Q |  |

*Таблица 5.5.7 – Итерация 4*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | сj | 7.5 | 3 | 0 | 0 |  |
|  |  | X1 | X6 | X7 | X5 |  |
| 12 | X4 | 0,45 | -0.05 | 0 | 0,25 | 799 |
| 3 | X2 | -0,05 | 0.2 | -0,1 | 0,02 | 20 |
| 6 | X3 | 0,42 | 0 | 0,17 | -0,04 | 367 |
|  | f | 0,40 | 0.04 | 0,75 | 2,81 | 11851 |
|  |  | Δ1 | Δ2 | Δ3 | Δ4 | Q |

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

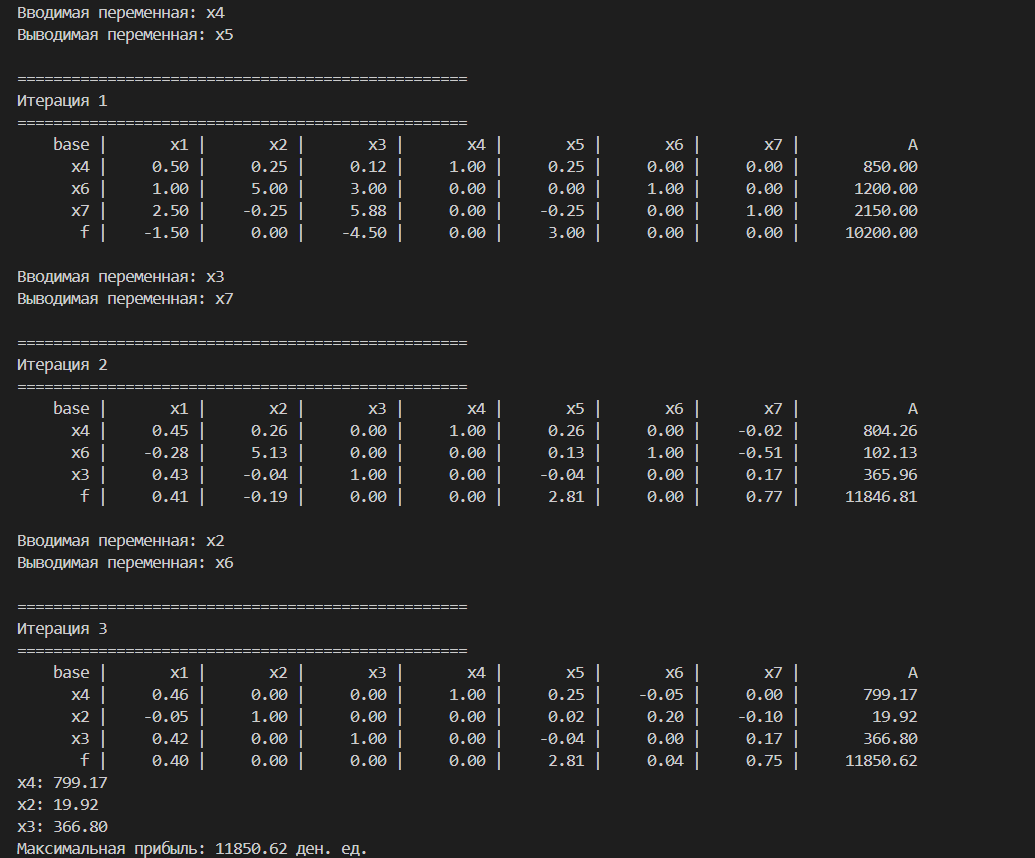
Подставляем базисное решение, которое дает последняя таблица

Где n – количество итераций

Проверим решение по «правилу прямоугольника».

Таким образом, фабрика должна выпускать в течении недели 𝑥4 = 800 шт. шкафов продукции D, 𝑥2 = 20 шт. шкафов продукции B, и 𝑥3 = 367 шт. продукции типа С. Тогда фабрика получит максимальный доход от продажи 11851 [тыс. ден.ед].

**5.3 Консольный результат работы**



**Рисунок 5.1 – Результат работы программы**

**5.4 Вывод по симплексному методу**

В ходе выполнения практической работы была решена задача линейного программирования, связанная с оптимизацией производства подшипников двух типов. На основании исходных данных была составлена математическая модель: целевая функция, отражающая прибыль, и система ограничений, описывающих доступное время работы оборудования. Затем задача была приведена к канонической форме, после чего решена с использованием симплексного метода.

Симплексный метод продемонстрировал свою эффективность для данной задачи, позволив найти оптимальное распределение ресурсов, максимизирующее прибыль предприятия.

К основным преимуществам симплексного метода можно отнести: высокую точность при решении линейных задач; чёткую пошаговую процедуру, удобную для алгоритмизации; возможность применения к задачам с большим числом переменных и ограничений.

Однако у метода есть и недостатки: чувствительность к численным ошибкам при большом количестве итераций; неэффективность при решении задач с огромным числом переменных; невозможность применения к задачам с нелинейными ограничениями или целевой функцией.

В целом, симплексный метод остаётся универсальным и надёжным инструментом для решения широкого класса прикладных задач линейного программирования.

**6 ДВОЙСТВЕННАЯ ЗАДАЧА**

Двойственная задача является важным понятием в линейном программировании и теории оптимизации. Она формулируется на основе исходной (прямой) задачи и отражает её структуру с иной точки зрения — через ограничения и оценки ресурсов. Каждой переменной прямой задачи соответствует ограничение двойственной, и наоборот. Анализ двойственной задачи позволяет получить дополнительную информацию о прямой задаче, включая экономические интерпретации, такие как теневая цена ресурсов. Сильная двойственность гарантирует равенство оптимальных значений целевых функций прямой и двойственной задач при наличии оптимальных решений. Использование двойственности облегчает решение задач и повышает эффективность вычислительных методов.

**6.1 Постановка задачи**

***Задание 8.*** Решить прямую ЗЛП с помощью симплексного метода и обратную с помощью теорем двойственности. Определить интервалы устойчивости.

***Задача.*** Для изготовления четырех видов продукции А, В, С и D используются три вида ресурсов I, II, III. Дальнейшее условие задачи в таблице П.8.

*Таблица П.6.8.* Исходные данные задачи.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Ресурсы | Нормы расхода сырья на  единицу продукции, ед. | | | | Запасы  ресурсов, ед. |
| А | В | С | D |
| 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 | 12 |  |

Требуется определить план выпуска продукции, при котором прибыль от ее реализации будет максимальной.

**6.2 Математическая модель исходной задачи**

Пусть х1 – тип шкафа А, х2 –тип шкафа В, х3 –тип шкафа С. Прибыль от продажи шкафов составит 9х1 + 11х2 + 15х3, прибыль требуется максимизировать.

Векторный вид:

В ходе решения прямой задачи было определено, что максимальный доход от продажи составляет тыс. ден.ед., оптимальный план

**6.3 Соответствующая исходной двойственная задача**

Найдем соответствующую двойственную задачу. Введем вектор двойственных переменных размерности три . Соответствующие векторы и матрица ограничений имеет вид:

.

Запишем двойственную задачу. Найти минимум функции.

При ограничениях:

**6.4 Первая теорема двойственности**

Если одна из пары двойственных задач имеет оптимальный план, то и другая имеет оптимальный план, причем экстремальные значения целевых функций равны. В ходе решения прямой задачи было определено, что максимальный доход от продажи составляет тыс. ден.ед., оптимальный план

Оптимальное решение двойственной задачи может быть получено из оптимального решения прямой задачи. Так как прямая задача имеет решение, то на основании первой теоремы о двойственности задача также разрешима. Ее решение может быть найдено из формулы:

где D – матрица, составленная из компонентов векторов входящих в последний базис, при котором получен оптимальный план исходной задачи.

В последней симплекс-таблице базисными переменными являются 𝑥4, 𝑥2, 𝑥3. Соответствующие этим переменным векторы , в разложении используются для формирования столбцов матрицы D.

Тогда,

*.*

Для вычисления обратной матрицы 𝐷-1 запишем матрицу 𝐷 дописав к ней справа единичную матрицу.

.

Для нахождения обратной матрицы 𝐷-1 используем элементарные преобразования над строками матрицы. Таким образом, преобразуются левая часть полученной матрицы в единичную.

Переставим 1 и 3 строку;

.

Из третьей строки вычтем первую строку умноженную на два;

,

Делим вторую строку на два;

,

Из третьей строки вычтем вторую строку;

.

Разделим третью строку на -24,1;

.

Из первой вычтем третью умноженную на 6 и из второй третью умноженную на 0.6;

Запишем обратную матрицу.

.

Базисными переменными в симплекс-таблице являются , тогда

.

При этом минимальное значение целевой функции двойственной задачи

совпадает с максимальным значением 𝑓𝑚𝑎𝑥 = [ден.ед.] прямой задачи, что является результатом взаимодвойственности. Таким образом:

**6.5 Вторая теорема двойственности**

Для того, чтобы планы и ЗЛП двойственной пары были оптимальными, необходимо и достаточно, чтобы эти планы удовлетворяли условиям дополняющей нежесткости.

.

.

Итак, имеем оптимальное решение прямой задачи: недельный объем производства продукции типа A – 𝑥1 = 0; недельный объем производства продукции типа В – 𝑥2 = 20; недельный объем производства продукции типа С – 𝑥3 = 367; недельный объем производства продукции типа D – 𝑥3 = 799; максимальный доход от продажи 𝑓𝑚𝑎𝑥 = [ден.ед. / неделю]. Рассмотрим выполнение неравенств прямой задачи при подстановке 𝑥4, 𝑥2, *х*3 в систему ограничений (Таблица 6.1).

Согласно Таблице 6.1 имеем следующую систему уравнений:

Решим данную систему уравнений

1 = 678/241, y2 = 9/241, y3 = 180/241

Решение, найденное из первой теоремы двойственности, равнозначно решению из второй теоремы.

Таким образом, вторая теорема дает нахождение оптимального решения двойственной задачи, пользуясь условием обращения в равенство сопряженных неравенств в системах ограничения.

*Таблица 6.1 – Выполнение неравенств прямой задачи*

|  |  |  |
| --- | --- | --- |
| Ограничение | Расчет | Вывод |
| 2x1 + х2 + 0.5х3 + 4x4 ≤ 3400 | 2\*0 + 20 + 0.5\*366 + 4\*799 < 3400  3399 < 3400 | Первое ограничение прямой задачи выполняется как строгое неравенство, остается спрос на продукцию. Значит, этот ресурс не является дефицитным и его оценка в оптимальном плане равна нулю (𝑦1 = 0). |
| х1 + 5х2 + 3х3 +0x4 ≤ 1200 | 0 + 5\*20 + 3\*366 + 0\*799 = 1200  1200 = 1200 | Второе ограничение прямой задачи выполняется как равенство. Это означает что шкафы полностью используется в оптимальном плане, является дефицитным и его оценка согласно второй теоремы двойственности отлична от нуля (𝑦3 ≠ 0). |
| 3х1 + 0х2 + 6х3 +x4 ≤ 3000 | 3\*0 + 0\*20 + 6\*366 + 799= 3000  3000 = 3000 | Третье ограничение прямой задачи выполняется как равенство. Это означает что шкафы полностью используется в оптимальном плане, является дефицитным и его оценка согласно второй теоремы двойственности отлична от нуля (𝑦3 ≠ 0). |
| х1 = 0 | 0 = 0 | Первое ограничение в двойственной задаче не будем учитывать |
| х2 ≥ 0 | 19 > 0 | Второе ограничение в двойственной задаче будет равенством |
| х3 ≥ 0 | 366 > 0 | Третье ограничение в двойственной задаче будет равенством , т.е. |
| Х4 ≥ 0 | 799 > 0 | Третье ограничение в двойственной задаче будет равенством |

**6.6 Третья теорема двойственности**

Третью теорему двойственности иногда называют теоремой об оценках. Рассматривая ограничения ЗЛП, можно констатировать: изменение правых частей ограничений исходной задачи приводит к изменению максимального значения целевой функции 𝑍𝑚𝑎𝑥.

Выпишем необходимые элементы из прямой задачи о максимальном доходе. Обратная матрица базиса оптимального плана:

Свободные члены неравенств (ограничений) прямой задачи:

Теперь воспользуемся формулами для нахождения нижней и верхней границ интервалов устойчивости оценок по видам ресурсов.

*Ресурс 3 (Тип C)*. Найдем нижнюю границу. В третьем столбце обратной матрицы есть положительные элементы (0,17 и 0,004), им соответствуют индексы базисных переменных оптимального плана.

Выбираем минимальное значение 17647.

Найдем верхнюю границу. В третьем столбце единственное отрицательное значение (–0,09), которое соответствует индексу базисной переменной оптимального плана (1200).

Таким образом, получаем

Тогда первый ресурс может изменяться в интервале:

При таком значении оптимальный план двойственной задачи остается неизменным. Аналогичные рассуждения позволяют найти интервалы устойчивости оценок для остальных ресурсов.

*Ресурс 2 (Ингредиент В).* Рассматриваем второй столбец обратной матрицы, в котором один положительный элемент (1/2) и два отрицательных (−1/2, −1/2). Данным элементам соответствуют следующие индексы базисных переменных оптимального плана: положительного элемента – 520; для отрицательных – 360, 220.

Тогда находим нижнюю границу.

Выбираем наименьшее значение, равное 24.

Найдем верхнюю границу.

Получаем

Тогда второй ресурс может изменяться в интервале:

*Ресурс 3 (Ограничение по недельному объему производства шкафов типа C по сравнению с объемом производства шкафов типа В).* Рассматриваем первый столбец обратной матрицы, в котором два положительных элемента. Данным элементам соответствует индекс соответствующих базисных переменных оптимального плана.

Находим нижнюю границу.

Найдем верхнюю границу.

Получаем

Получаем интервал устойчивости оценок по отношению к третьему ограничению:

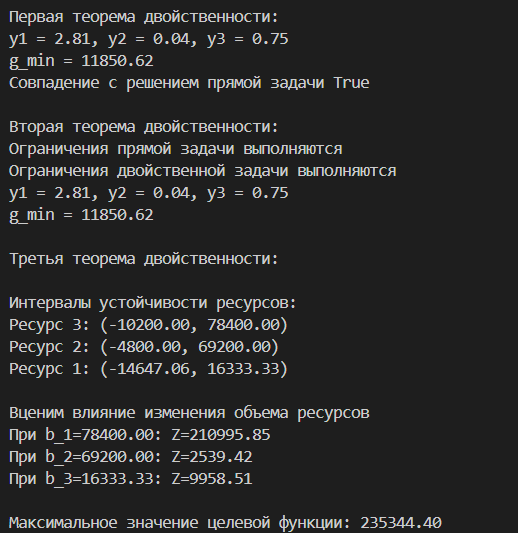
Далее оценим влияние изменения объема ресурсов на величину максимальной стоимости продукции. Как известно, это дефицитные ресурсы Введем верхние границы в формулу:

Совместное влияние изменений этих ресурсов приводит к изменению максимальной стоимости продукции 𝐺𝑚𝑎𝑥 на величину:

Следовательно, оптимальное значение целевой функции при максимальном изменении ресурсов:

Таким образом, двойственные оценки позволяют судить о чувствительности решения к изменениям.

**6.7 Консольный результат программы**



**Рисунок 6.1 – Результат работы программы**

**6.8 Вывод по двойственному методу**

Таким образом, двойственная задача является мощным инструментом в анализе и решении задач линейного программирования. Она не только дополняет прямую задачу, но и позволяет глубже понять экономический смысл ограничений и переменных. Использование двойственности даёт возможность оценить эффективность использования ресурсов и выявить скрытые зависимости между параметрами задачи. Кроме того, проверка соответствия решений прямой и двойственной задач служит критерием оптимальности. Благодаря этим свойствам, двойственная задача широко применяется в экономике, логистике, управлении и других областях. Её использование способствует более обоснованному и эффективному принятию решений.

**ЗАКЛЮЧЕНИЕ**

В заключение, теория принятия решений предоставляет широкий арсенал методов для обоснованного выбора оптимального варианта в сложных ситуациях. Методы Паретто и Электро II позволяют эффективно работать с многокритериальными задачами, учитывая предпочтения и компромиссы, но метод Паретто не учитывает важность критериев, а Электро сложен в выполнении, что делает его реализацию не эффективной. Метод анализа иерархий способствует структурированию проблемы и учёту экспертных суждений при сравнении альтернатив, но сильно зависит от субъекта, который занимается методом. Графический метод нагляден и удобен для задач с двумя переменными, позволяя визуализировать множество допустимых решений, но не эффективен для многомерных задач. Симплексный метод и двойственная задача обеспечивают эффективное решение задач линейного программирования и дают ценные экономические интерпретации. Совместное использование этих подходов позволяет принимать более точные, обоснованные и устойчивые решения в различных сферах деятельности.

**СПИСОК ИНФОРМАЦИОННЫХ ИСТОЧНИКОВ**

1. Болотова Л. С. Многокритериальная оптимизация. Болотова Л. С., Сорокин А. Б. [Электронный ресурс] / Метод. указания по вып. курсовой работы — М.: МИРЭА, 2015.
2. Сорокин А. Б. Методы оптимизации: гибридные генетические алгоритмы. Сорокин А. Б. [Электронный ресурс] / Учебно-метод. пособие — М.: МИРЭА, 2016.
3. Сорокин А. Б. Линейное программирование: практикум. Сорокин А. Б., Бражникова Е. В., Платонова О. В. [Электронный ресурс] / Учебно-метод. пособие — М.: МИРЭА, 2017.
4. McKinney, W. (2022). Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython (3rd ed.). O'Reilly Media.  
   — Подробное руководство по обработке данных в Python с использованием pandas и NumPy.
5. Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., ... & Oliphant, T. E. (2020). *Array programming with NumPy*. *Nature*, 585(7825), 357–362. https://doi.org/10.1038/s41586-020-2649-2.
6. Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., ... & van der Walt, S. J. (2020). *SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python*. *Nature Methods*, 17, 261–272. https://doi.org/10.1038/s41592-019-0686-2.
7. The pandas development team. (2024). *pandas documentation*. Retrieved from https://pandas.pydata.org/docs.
8. NumPy Developers. (2024). *NumPy documentation*. Retrieved from https://numpy.org/doc/.
9. SciPy Developers. (2024). *SciPy documentation*. Retrieved from https://docs.scipy.org/doc/.

**ПРИЛОЖЕНИЯ**

Приложение А – Код реализации метода Парето на языке Python.

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

Приложение В – Код реализации метода МАИ на языке Python.

Приложение Г – Код реализации Симплексного метода на языке Python.

Приложение Д – Код реализации Двойственной задачи на языке Python.

**Приложение А**

Код реализации метода Парето на языке Python.

*Листинг А.1. . Реализация Парето.*

import pandas as pd

alts = pd.DataFrame([

{"name": "TIE Fighter", "credits": 20000, "speed": 5000, "hyper": 3.0, "weapons": 2, "shields": 100},

{"name": "TZ-24", "credits": 22000, "speed": 4900, "hyper": 3.2, "weapons": 4, "shields": 120},

{"name": "S-100", "credits": 21000, "speed": 4800, "hyper": 3.1, "weapons": 3, "shields": 150},

{"name": "F-T2", "credits": 30000, "speed": 5100, "hyper": 4.0, "weapons": 3, "shields": 110},

{"name": "CR90", "credits": 25000, "speed": 4600, "hyper": 3.5, "weapons": 2, "shields": 130},

{"name": "IL-5", "credits": 26000, "speed": 4700, "hyper": 3.7, "weapons": 2, "shields": 100},

{"name": "FT-6", "credits": 35000, "speed": 4400, "hyper": 4.5, "weapons": 2, "shields": 110},

{"name": "FT-8", "credits": 34000, "speed": 4500, "hyper": 4.3, "weapons": 3, "shields": 115},

{"name": "S-13", "credits": 33000, "speed": 4600, "hyper": 4.1, "weapons": 2, "shields": 105},

{"name": "S-SC4", "credits": 32000, "speed": 4700, "hyper": 3.9, "weapons": 3, "shields": 125},

])

min\_crit = ["credits", "hyper"]

plus\_crit = ["speed", "weapons", "shields"]

def dom(a, b):

crits = sum(a[min\_crit].values <= b[min\_crit].values) + sum(a[plus\_crit].values >= b[plus\_crit].values)

return crits == len(min\_crit) + len(plus\_crit)

def pareto(alts):

pareto = []

for i in range (len(alts)):

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

if dom(alts.iloc[i], alts.iloc[j]):

pareto.append(alts.iloc[i])

break

return pd.DataFrame(pareto)

opt\_alt = pareto(alts)

print("\nОптимальное множество:")

print(opt\_alt["name"].to\_list())

*Продолжение листинга А.1.*

fil\_opt\_alt = opt\_alt[

(opt\_alt["credits"] <= 22000) &

(opt\_alt["speed"] >= 4800) &

(opt\_alt["hyper"] >= 3.0) &

(opt\_alt["weapons"] >= 3) &

(opt\_alt["shields"] > 100)

]

print("\nУказание верхних/нижних границ критериев:")

print(fil\_opt\_alt["name"].to\_list())

fil\_opt\_alt = opt\_alt[

(opt\_alt["credits"] <= 25000) &

(opt\_alt["hyper"] >= 3.2) &

(opt\_alt["weapons"] > 2) &

(opt\_alt["shields"] >= 120)

]

fil\_opt\_alt = fil\_opt\_alt.sort\_values(by="speed", ascending=False)

print("\nСубоптимизация:")

print(fil\_opt\_alt["name"].to\_list())

criteria\_priority = ["speed", "weapons", "shields", "credits", "hyper"]

filtered\_opt\_alt = alts.sort\_values(by=criteria\_priority, ascending=[False, False, False, True, True])

print("\nЛексикографическая оптимизация:")

print(filtered\_opt\_alt["name"].to\_list()[0])

*Конец листинга А.1.*

**Приложение Б**

Код реализации метода Электра II на языке Python.

*Листинг Б.1. Реализация метода Электра II.*

import pandas as pd

alts = pd.DataFrame([

{"name": "TIE Fighter", "credits": 10, "speed": 30, "hyper": 10, "weapons": 10, "shields": 5},

{"name": "TZ-24", "credits": 10, "speed": 25, "hyper": 10, "weapons": 30, "shields": 10},

{"name": "S-100", "credits": 10, "speed": 20, "hyper": 10, "weapons": 20, "shields": 25},

{"name": "F-T2", "credits": 20, "speed": 30, "hyper": 20, "weapons": 20, "shields": 5},

{"name": "CR90", "credits": 10, "speed": 10, "hyper": 10, "weapons": 10, "shields": 15},

{"name": "IL-5", "credits": 10, "speed": 15, "hyper": 20, "weapons": 10, "shields": 5},

{"name": "FT-6", "credits": 30, "speed": 5, "hyper": 30, "weapons": 10, "shields": 5},

{"name": "FT-8", "credits": 30, "speed": 5, "hyper": 30, "weapons": 20, "shields": 10},

{"name": "S-13", "credits": 30, "speed": 10, "hyper": 30, "weapons": 10, "shields": 5},

{"name": "S-SC4", "credits": 30, "speed": 15, "hyper": 20, "weapons": 20, "shields": 15},

])

min\_crit = [1, 3]

plus\_crit = [2, 4, 5]

N = len(alts)

pref\_table = pd.DataFrame()

def compare(s1, s2):

P = 0

N = 0

for crit in min\_crit:

P += s1.iloc[crit] if s1.iloc[crit] < s2.iloc[crit] else 0

N += s2.iloc[crit] if s1.iloc[crit] > s2.iloc[crit] else 0

for crit in plus\_crit:

P += s1.iloc[crit] if s1.iloc[crit] > s2.iloc[crit] else 0

N += s2.iloc[crit] if s1.iloc[crit] < s2.iloc[crit] else 0

if (N == 0):

return "inf"

if P/N > 1:

return str(P) + "/" + str(N)

else:

return "-"

for i in range(N):

for j in range(N):

if i == j:

pref\_table.loc[i,j] = "x"

else:

pref\_table.loc[i,j] = compare(alts.loc[i], alts.loc[j])

print("Матрицы предпочтений")

*Продолжение листинга Б.1.*

print(pref\_table)

N = len(alts)

pref\_table = pd.DataFrame()

def compare(s1, s2):

P = 0

N = 0

for crit in min\_crit:

P += s1.iloc[crit] if s1.iloc[crit] < s2.iloc[crit] else 0

N += s2.iloc[crit] if s1.iloc[crit] > s2.iloc[crit] else 0

for crit in plus\_crit:

P += s1.iloc[crit] if s1.iloc[crit] > s2.iloc[crit] else 0

N += s2.iloc[crit] if s1.iloc[crit] < s2.iloc[crit] else 0

if (N == 0):

return "inf"

if P/N > 1.3:

return str(P) + "/" + str(N)

else:

return "-"

for i in range(N):

for j in range(N):

if i == j:

pref\_table.loc[i,j] = "x"

else:

pref\_table.loc[i,j] = compare(alts.loc[i], alts.loc[j])

print("\nМатрицы предпочтений с порогом 1.5")

print(pref\_table)

*Конец листинга Б.1.*

**Приложение В**

Код реализации метода МАИ на языке Python.

*Листинг В.1. Реализация МАИ.*

import pandas as pd

import numpy as np

all\_crit = pd.DataFrame([

{"K1": 1,"K2": 1/7,"K3": 3,"K4": 2,"K5": 2},

{"K1": 7,"K2": 1,"K3": 3,"K4": 5,"K5": 5},

{"K1": 1/3,"K2": 1/3,"K3": 1,"K4": 1,"K5": 1},

{"K1": 1/2,"K2": 1/5,"K3": 1,"K4": 1,"K5": 3},

{"K1": 1/2,"K2": 1/5,"K3": 1,"K4": 1/3,"K5": 1},

])

crit1 = pd.DataFrame([

{"A1": 1,"A2": 3,"A3": 3,"A4": 5,"A5": 5},

{"A1": 1/3,"A2": 1,"A3": 3,"A4": 5,"A5": 5},

{"A1": 1/3,"A2": 1/3,"A3": 1,"A4": 5,"A5": 3},

{"A1": 1/5,"A2": 1/5,"A3": 1/5,"A4": 1,"A5": 1},

{"A1": 1/5,"A2": 1/5,"A3": 1/3,"A4": 1,"A5": 1},

])

crit2 = pd.DataFrame([

{"A1": 1,"A2": 1/3,"A3": 1/3,"A4": 1/5,"A5": 1/3},

{"A1": 3,"A2": 1,"A3": 1/3,"A4": 1/3,"A5": 1/5},

{"A1": 3,"A2": 3,"A3": 1,"A4": 1,"A5": 1},

{"A1": 5,"A2": 3,"A3": 1,"A4": 1,"A5": 1},

{"A1": 3,"A2": 5,"A3": 1,"A4": 1,"A5": 1},

])

crit3 = pd.DataFrame([

{"A1": 1,"A2": 1,"A3": 3,"A4": 5,"A5": 5},

{"A1": 1,"A2": 1,"A3": 1,"A4": 5,"A5": 5},

{"A1": 1/3,"A2": 1,"A3": 1,"A4": 3,"A5": 3},

{"A1": 1/5,"A2": 1/5,"A3": 1/3,"A4": 1,"A5": 1},

{"A1": 1/5,"A2": 1/5,"A3": 1/3,"A4": 1,"A5": 1},

])

crit4 = pd.DataFrame([

{"A1": 1,"A2": 3,"A3": 1,"A4": 1,"A5": 1},

{"A1": 1/3,"A2": 1,"A3": 1/3,"A4": 1/3,"A5": 1/3},

{"A1": 1,"A2": 1/3,"A3": 1,"A4": 1,"A5": 1},

{"A1": 1,"A2": 1/3,"A3": 1,"A4": 1,"A5": 1},

{"A1": 1,"A2": 1/3,"A3": 1,"A4": 1,"A5": 1},

])

crit5 = pd.DataFrame([

{"A1": 1,"A2": 5,"A3": 5,"A4": 3,"A5": 3},

{"A1": 1/5,"A2": 1,"A3": 3,"A4": 5,"A5": 5},

{"A1": 1/5,"A2": 1/3,"A3": 1,"A4": 3,"A5": 3},

{"A1": 1/3,"A2": 1/5,"A3": 1/3,"A4": 1,"A5": 1},

{"A1": 1/3,"A2": 1/5,"A3": 1/3,"A4": 1,"A5": 1},

])

*Продолжение листинга В.1.*

tabs = [crit1,crit2,crit3,crit4,crit5]

def vec\_prior(tab):

w = []

v1 = (tab.iloc[0].prod() \*\* (1/5)).round(3)

v2 = (tab.iloc[1].prod() \*\* (1/5)).round(3)

v3 = (tab.iloc[2].prod() \*\* (1/5)).round(3)

v4 = (tab.iloc[3].prod() \*\* (1/5)).round(3)

v5 = (tab.iloc[4].prod() \*\* (1/5)).round(3)

v\_all = v1 + v2 + v3 + v4 + v5

w.append((v1 / v\_all).round(3))

w.append((v2 / v\_all).round(3))

w.append((v3 / v\_all).round(3))

w.append((v4 / v\_all).round(3))

w.append((v5 / v\_all).round(3))

return w

def check(tab, vec):

p1 = tab["K1"].sum()

p2 = tab["K2"].sum()

p3 = tab["K3"].sum()

p4 = tab["K4"].sum()

p5 = tab["K5"].sum()

lamb = p1 + p2 + p3 + p4 +p5

return (((lamb - 4) / (5 - 1)) / 1.12).round(3)

alts\_vec = []

crit\_vec = []

vec = vec\_prior(all\_crit)

print("Вектор приоритетов", [float(x) for x in vec])

[crit\_vec.append(float(x)) for x in vec]

for i, tab in enumerate(tabs, 1):

alt = []

vec = vec\_prior(tab)

print("Вектор приоритетов", [float(x) for x in vec])

[alt.append(float(x)) for x in vec]

alts\_vec.append(alt)

best\_alt\_vec = []

print()

for i in range (5):

W = 0

for j in range (5):

W += crit\_vec[j] \* alts\_vec[j][i]

print(f"Приоритет альтернативы №{i + 1} : {round(W,3)}")

best\_alt\_vec.append(round(W,3))

print(f"\nЛучшая альтернатива : №{np.argmax(best\_alt\_vec) + 1}")

*Конец листинга В.1.*

**Приложение Г**

Код реализации Симплексного метода на языке Python.

*Листинг Г.1. Реализация симплексного метда.*

import numpy as np

def print\_table(table, basis, columns, iteration):

print(f"\n{'='\*50}\nИтерация {iteration}\n{'='\*50}")

header = ["base"] + columns

rows = []

for i in range(len(table)-1):

row = [basis[i]] + [f"{val:.2f}" if abs(val) >= 0.01 else "0.00" for val in table[i]]

rows.append(row)

rows.append(["f"] + [f"{val:.2f}" for val in table[-1]])

for row in [header] + rows:

print("{: >8} | {: >8} | {: >8} | {: >8} | {: >8} | {: >8} | {: >8} | {: >8} | {: >12}".format(\*row))

def simplex\_method():

num\_vars = 4

num\_slack = 3

columns = ['x1', 'x2', 'x3', 'x4', 'x5', 'x6', 'x7', 'A']

basis = ['x5', 'x6', 'x7']

iteration = 1

table = np.array([

[2, 1, 0.5, 4, 1, 0, 0, 3400],

[1, 5, 3, 0, 0, 1, 0, 1200],

[3, 0, 6, 1, 0, 0, 1, 3000],

[-7.5, -3, -6, -12, 0, 0, 0, 0]

], dtype=float)

print\_table(table, basis, columns, 0)

while True:

z\_row = table[-1, :-1]

if all(z\_row >= 0):

break

entering\_col = np.argmin(z\_row)

print(f"\nВводимая переменная: {columns[entering\_col]}")

ratios = []

for row in table[:-1]:

if row[entering\_col] > 0:

ratios.append(row[-1] / row[entering\_col])

else:

ratios.append(np.inf)

exiting\_row = np.argmin(ratios)

print(f"Выводимая переменная: {basis[exiting\_row]}")

pivot = table[exiting\_row, entering\_col]

table[exiting\_row] = table[exiting\_row] / pivot

for i in range(len(table)):

if i != exiting\_row:

factor = table[i, entering\_col]

table[i] -= factor \* table[exiting\_row]

basis[exiting\_row] = columns[entering\_col]

print\_table(table, basis, columns, iteration)

iteration += 1

for var, val in zip(basis, table[:-1, -1]):

print(f"{var}: {val:.2f}")

print(f"Максимальная прибыль: {table[-1, -1]:.2f} ден. ед.")

simplex\_method()

*Конец листинга Г.1.*

**Приложение Д**

Код реализации Двойственной задачи на языке Python.

Листинг Д.1 - Реализация Двойственной задачи

import numpy as np

from scipy.optimize import linprog

from sympy import symbols, Eq, solve

a\_primal = np.array([

[2, 1, 0.5, 4],

[1, 5, 3, 0],

[3, 0, 6, 1]

])

b\_primal = np.array([3400, 1200, 3000])

c\_primal = np.array([7.5, 3, 6, 12])

A\_dual = a\_primal.T

y1, y2, y3 = symbols('y1 y2 y3')

Eqs = []

for i in range(4):

Eqs.append(Eq(A\_dual[i][0]\*y1 + A\_dual[i][1]\*y2 + A\_dual[i][2]\*y3, c\_primal[i]))

B\_inv = np.array([

[0.25, -0.05, 0.004],

[0.02, 0.2, -0.09],

[-0.04, 0.008, 0.17]

])

# Первая теорема

print(f"Первая теорема двойственности:")

res\_primal = linprog(

-c\_primal,

A\_ub=a\_primal,

b\_ub=b\_primal,

bounds=[(0, None)] \* 4,

method='highs'

)

x\_opt = res\_primal.x

y\_opt = -res\_primal.ineqlin.marginals

g = b\_primal[0] \* (y\_opt[0]) + b\_primal[1] \* (y\_opt[1]) + b\_primal[2] \* (y\_opt[2])

print(f"y1 = {round(-res\_primal.ineqlin.marginals[0],2)}, y2 = {round(-res\_primal.ineqlin.marginals[1],2)}, y3 = {round(-res\_primal.ineqlin.marginals[2],2)}")

print(f"g\_min = {round(g,2)}")

print(f"Совпадение с решением прямой задачи {np.isclose(round(g,0), 11851)}")

# Вторая теорема

print(f"\nВторая теорема двойственности:")

lims = 0

for line in a\_primal:

check = [line[i] \* x\_opt[i] for i in range(4)]

*Продолжение листинга Д.1.*

|  |
| --- |
| check = sum(check) > b\_primal[lims]  lims += 1  if check:  print(f"Невыполение условия")  exit(0)  print(f"Ограничения прямой задачи выполняются")  fin\_eqs = []  for i in range(4):  if x\_opt[i] > 0:  fin\_eqs.append(Eqs[i])  print(f"Ограничения двойственной задачи выполняются")  solutions = solve((fin\_eqs), (y1, y2, y3))  g\_min = b\_primal[0] \* (solutions[y1]) + b\_primal[1] \* (solutions[y2]) + b\_primal[2] \* (solutions[y3])  print(f"y1 = {round(solutions[y1],2)}, y2 = {round(solutions[y2],2)}, y3 = {round(solutions[y3],2)}")  print(f"g\_min = {round(g\_min,2)}")  print(f"\nТретья теорема двойственности:")  def calculate\_limits(B\_inv, b):  limits = []  for col in range(B\_inv.shape[1]):  column = B\_inv[:, col]    pos\_mask = column > 0  if any(pos\_mask):  delta\_lower = min(b[pos\_mask]/column[pos\_mask])  else:  delta\_lower = np.inf    neg\_mask = column < 0  if any(neg\_mask):  delta\_upper = min(b[neg\_mask]/abs(column[neg\_mask]))  else:  delta\_upper = np.inf    limits.append((b[col] - delta\_lower, b[col] + delta\_upper))  return limits  def calculate\_max\_Z(y\_opt, limits, Z\_original):  delta\_Z = 0  for i in range(len(y\_opt)):  if y\_opt[i] > 1e-6:  delta\_b = limits[i][1] - b\_primal[i]  delta\_Z += y\_opt[i] \* delta\_b  return Z\_original + delta\_Z  resource\_limits = calculate\_limits(B\_inv, b\_primal)  Z\_max = calculate\_max\_Z(y\_opt, resource\_limits, g\_min)  print("\nИнтервалы устойчивости ресурсов:")  i = 3  for (lower, upper) in resource\_limits:  print(f"Ресурс {i}: ({lower:.2f}, {upper:.2f})")  i -= 1  print("\nВценим влияние изменения объема ресурсов")  for i in range(len(b\_primal)):  if y\_opt[i] > 1e-6:  print(f"При b\_{i+1}={resource\_limits[i][1]:.2f}: Z={y\_opt[i]\*(resource\_limits[i][1]-b\_primal[i]):.2f}")  print(f"\nМаксимальное значение целевой функции: {Z\_max:.2f}") |

*Конец листинга Д.1.*