Титульные листы по курсовой работе

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Принятие решений основывается на целях, поставленных лицом, принимающим решения. Научное изучение данного вопроса дает возможность отыскать наиболее подходящие методы исследования и сформировать теоретический и опытный фундамент для решения задач. Эффективные действия позволяют достичь поставленной цели при наименьших расходах трудящийся, вещественных и сырьевых ресурсов.

Математическое программирование применяется для решения распределительных задач, в которых существующих ресурсов недостаточно для выполнения запланированных работ в соответствии с подобранным критерием оптимальности. Решение проблем с использованием математического программирования применяют в различных сферах, требующих выбор одного из предложенных процессов деятельности. В частности, линейное программирование применяется при решении экономических задач, в таких задачах как управление и планирование производства; в задачах определения оптимального плана перевозок груза; в задачах оптимального распределения кадров.

В данной работе рассматриваются некоторые методы нахождения оптимального решения для многокритеальных задач, способы разрешения проблем в задачах линейного программирования.

1 МЕТОД ПАРЕТО

Метод Парето — это подход к решению задач многокритериальной оптимизации, который позволяет находить наиболее выгодные решения, учитывая несколько критериев одновременно.

Основная идея метода заключается в том, чтобы выделить такие решения, которые нельзя улучшить по одному критерию, не ухудшив при этом другой. Эти решения называются Парето-оптимальными, а их совокупность образует множество Парето.

Алгоритм метода:

1. Определяются альтернативы и критерии, по которым они будут оцениваться.
2. Каждая альтернатива попарно сравнивается с другими по всем критериям. Если одна альтернатива лучше или равна по всем критериям и строго лучше хотя бы по одному, то она называется доминирующей.
3. Альтернативы, которые не доминируются никакой другой альтернативой, включаются в множество Парето. В результате, это множество содержит только те решения, которые не могут быть улучшены по всем критериям одновременно.
4. После построения множества Парето лицо, принимающее решение, может выбрать одну из альтернатив на основе дополнительных предпочтений или других критериев.

Часто решение многокритериальной задачи состоит в построении множества Парето-оптимальных точек и дальнейшем выборе одной из них с помощью какого-либо критерия. Во всех случаях задача многокритериальной оптимизации каким-то способом сводится к задаче с одним критерием.

1.1 Выбор Парето-оптимального множества

Оптимизируем выбор VPN – приложения. Выделим 10 альтернатив (вариантов решений) и 4 наиболее важных критериев оценки: стоимость подписки (руб./мес.), рейтинг приложения, скорость соединения (Mbps), задержка соединения (мс). Также определим стремления критериев. Знаком (-) указывается отрицательное стремление критерия (чем меньше, тем лучше), а знаком (+) – положительное (чем больше, тем лучше). Все данные сведены в таблицу 1.1.

Таблица 1.1. Альтернативы и критерии

№	Варианты решений	Критерии			
		Стоимость подписки (руб./мес.) (-)	Рейтинг приложения (+)	Скорость соединения (Mbps) (+)	Задержка соединения (мс) (-)
A1	Outline VPN	300	4.5	50	50
A2	ExpressVPN	100	4.8	100	30
A3	TunnelBear	90	4.2	90	70
A4	NordVPN	80	4.7	80	40
A5	CyberGhost	105	4.6	95	60
A6	Surfshark	85	4.7	85	45
A7	Privare Internet Access	60	4.3	60	55
A8	ProtonVPN	90	4	90	80
A9	IPVanish	95	4.4	70	65
A10	Windscribepo	50	4.5	60	50

Следуя алгоритму Парето, попарно сравниваем все альтернативы: несравнимые критерии отметим в таблице как «н», доминирующие критерии указываем в ячейках. Далее исключаем те, которые доминируются другими. В результате остается множество Парето, которое содержит только те альтернативы, которые не могут быть улучшены по всем критериям одновременно. Весь процесс сравнения сведем в таблицу (Таблица 1.2).

Таблица 1.2. Сравнения альтернатив

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

Примечание: Знаком (н) указываются несравнимые альтернативы.

Парето-оптимальное множество определено альтернативами {1, 3, 5, 10} представлено в Таблице 1.3.

Таблица 1.3. Парето-оптимальное множество

№	Варианты решений	Критерии			
		Стоимость подписки (руб./мес.) (-)	Рейтинг приложения (+)	Скорость соединения (Mbps) (+)	Задержка соединения (мс) (-)
A2	ExpressVPN	100	4.8	100	30
A3	TunnelBear	90	4.2	90	70
A4	NordVPN	80	4.7	80	40
A6	Surfshark	85	4.7	85	45
A10	Windscribepo	50	4.5	60	50

1.2 Указание верхних/нижних границ критериев.

Для метода верхних/нижних критериев необходимо установить верхнюю и нижнюю границы:

- Скорость соединения не менее 85 Mbps;
- Стоимость подписки не более 90 рублей в месяц.

В результате установки границ остается единственная альтернатива, представленная в Таблице 1.4.

Таблица 1.4. Результат верхних и нижних границ критериев

№	Варианты решений	Критерии			
		Стоимость подписки (руб./мес.) (-)	Рейтинг приложения (+)	Скорость соединения (Mbps) (+)	Задержка соединения (мс) (-)
A3	TunnelBear	90	4.2	90	70
A4	NordVPN	80	4.7	80	40

Основной недостаток метода состоит в том, что оптимальное решение становится здесь субъективным, так как зависит, от величин верхних/нижних границ критериев, назначаемых лицом, принимающим решения.

1.3 Субоптимизация

Для метода субоптимизации необходимо выделить один из критериев, а по всем остальным критериям назначать нижние границы. Оптимальным при этом считается исход, максимизирующий выделенный критерий на множестве исходов, оценки которых по остальным критериям не ниже назначенных.

Выберем главный критерий: средний чек. Установим верхние/нижние границы для остальных критериев: рейтинг не ниже 4,7; Задержка соединения не более 50 мс. Отбросим варианты, которые не удовлетворяют данным ограничениям и составим таблицу (Таблица 1.5).

Таблица 1.5. Результат субоптимизации

№	Варианты решений	Критерии			
		Стоимость подписки (руб./мес.) (-)	Рейтинг приложения (+)	Скорость соединения (Mbps) (+)	Задержка соединения (мс) (-)
A2	ExpressVPN	100	4.8	100	30

Из (табл. 1.5) видно, остаётся единственный вариант – ExpressVPN. С помощью метода субоптимизации задача многокритериальной оптимизации превращается в задачу скалярной оптимизации на суженном допустимом множестве. Выделение одного из критериев, а также указание нижних границ для остальных критериев основано на дополнительной информации, получаемой от ЛПР (лицо принимающее решение). Следовательно, окончательное решение здесь также имеет субъективный характер.

1.4 Лексикографическая оптимизация

Лексикографическая оптимизация основана на упорядочении критериев по их относительной важности. На первом шаге отбираются исходы, которые имеют максимальную оценку по важнейшему критерию. Если такой исход единственный, то он и считается оптимальным. Если же таких исходов несколько, то среди них отбираются те, которые имеют максимальную оценку по следующему за важнейшим критерию. В результате такой процедуры всегда остается единственный исход — он и будет оптимальным.

Упорядочим критерии по их относительной важности:

1. Стоимость подписки
2. Скорость соединения
3. Задержка соединения

4. Рейтинг приложения

Таблица 1.6. Результат лексикографиского метода

№	Варианты решений	Критерии			
		Стоимость подписки (руб./мес.) (-)	Рейтинг приложения (+)	Скорость соединения (Mbps) (+)	Задержка соединения (мс) (-)
A10	Windscribepo	50	4.5	60	50

Из Таблицы 1.6 видно, что осталась одна альтернатива. Таким образом, наглядно проявляется недостаток лексикографической оптимизации — фактический учет одного важнейшего критерия.

1.5 Результаты работы программы

Результат работы программы показан ниже (Рисунки 1.1 – 1.5).

Исходная таблица с альтернативами и критериями:

№	Варианты решений	Стоимость подписки (руб./мес.) (-)	Рейтинг приложения (+)	Скорость соединения (Mbps) (+)	Задержка соединения (мс) (-)
A1	Outline VPN	300	4.5	50	50
A2	ExpressVPN	100	4.8	100	30
A3	TunnelBear	90	4.2	90	70
A4	NordVPN	80	4.7	80	40
A5	CyberGhost	105	4.6	95	60
A6	Surfshark	85	4.7	85	45
A7	Privare Internet Access	60	4.3	60	55
A8	ProtonVPN	90	4.0	90	80
A9	IPVanish	95	4.4	70	65
A10	Windscribepo	50	4.5	60	50

Рисунок 1.1 – Исходная таблица с альтернативами и критериями

Оптимальное множество Парето:

№	Варианты решений	Стоимость подписки (руб./мес.) (-)	Рейтинг приложения (+)	Скорость соединения (Mbps) (+)	Задержка соединения (мс) (-)
A2	ExpressVPN	100	4.8	100	30
A3	TunnelBear	90	4.2	90	70
A4	NordVPN	80	4.7	80	40
A6	Surfshark	85	4.7	85	45
A10	Windscribepo	50	4.5	60	50

Рисунок 1.2 – Таблица оптимального множества Парето

Метод указания верхних/нижних границ:

№	Варианты решений	Стоимость подписки (руб./мес.) (-)	Рейтинг приложения (+)	Скорость соединения (Mbps) (+)	Задержка соединения (мс) (-)
A3	TunnelBear	90	4.2	90	70
A6	Surfshark	85	4.7	85	45

Рисунок 1.3 – Метод указания верхних/нижних границ

Субоптимизация:

№	Варианты решений	Стоимость подписки (руб./мес.) (-)	Рейтинг приложения (+)	Скорость соединения (Mbps) (+)	Задержка соединения (мс) (-)
A2	ExpressVPN	100	4.8	100	30

Рисунок 1.4 – Метод субоптимизации

Рисунок 1.5 – Метод лексикографической оптимизации

Лексикографическая оптимизация:

№	Варианты решений	Стоимость подписки (руб./мес.) (-)	Рейтинг приложения (+)	Скорость соединения (Mbps) (+)	Задержка соединения (мс) (-)
A10	Windscribepo	50	4.5	60	50

1.6 Вывод по методу Парето

Можно выделить несколько преимуществ метода Парето. Он позволяет исключить заведомо неудачные решения, что упрощает процесс принятия решений. Метод Парето достаточно гибкий: после построения множества Парето можно использовать дополнительные методы для выбора окончательного решения. Однако, при использовании метода Парето можно столкнуться с некоторыми трудностями. Принцип основан на субъективных наблюдениях и обобщениях, поэтому может содержать ошибки и оценочные суждения. Метод плохо применим в условиях неопределённости или там, где системы являются хаотичными. В таких случаях взаимосвязи между элементами настолько сложны и непредсказуемы, что выделение ключевых 20% становится практически невозможным.

2 МЕТОД ЭЛЕКТРА II

Метод Электра II - это алгоритм выбора наилучшего решения из множества альтернативных вариантов. Алгоритм основан на использовании методов анализа и принятия решений, а также на оценке возможных последствий каждого варианта.

Этапы алгоритма:

1. Определение целей и критериев оценки альтернатив.
2. Оценка каждой альтернативы по каждому критерию.
3. Ранжирование альтернатив по совокупной оценке.
4. Принятие решения на основе полученных результатов.

Отличительной особенностью этого метода является наличие веса у каждого критерия. Вес – число, характеризующее важность того или иного свойства альтернативы, может определяться различными способами.

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
Безопасность	5	Высокий Средний Низкий	12 8 4	(-)
Рейтинг сервисов	4	5.0 4.8 – 4.9 4.6 – 4.7	12 8 4	(+)
Стоимость подписки	2	Больше 200 руб 200 руб или меньше	12 6	(-)
Скорость соединения (Mbps)	4	25 – 27 23 – 25 21 - 23	12 8 4	(-)
Количество отзывов	1	Очень много Много Средне Мало	12 9 6 3	(+)

Составлена таблица оценок выбора лучшего (VPN сервиса). Для 10-ти альтернатив заполнена Таблицу 2.

Таблица 2.2 – Таблица оценок по критериям

№	Варианты решений	Критерии				
		Безопастность	Рейтинг сервиса	Скорость соединения (Mbps)	Задержка соединения (мс)	Количество отзывов
1	Outline VPN	8	12	6	4	9
2	ExpressVPN	12	8	12	8	9
3	TunnelBear	12	4	6	8	9
4	NordVPN	12	4	6	8	6
5	CyberGhost	4	8	6	4	6
6	Surfshark	4	12	6	12	12
7	Privare Internet Access	12	12	6	8	3
8	ProtonVPN	4	4	6	8	9
9	IPVanish	8	4	6	12	6
10	Windscribepo	4	4	6	8	9
Вес		5	4	2	4	1

Стремление	(-)	(+)	(-)	(-)	(+)

2.2 Веса предпочтений

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

P12 = 5 + 4 + 2 + 4 + 0 = 15;

N12 = 0 + 0 + 0 + 0 + 0 = 0;

D12 = P12/N12 = 15/0 = ∞ – принимаем.

P21 = 0 + 0 + 0 + 0 + 0 = 0;

N21 = 5 + 4 + 2 + 4 + 0 = 15;

D21 = P21/N21 = 0/15 = 0 < 1 – отбрасываем.

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

P13 = 5 + 4 + 0 + 4 + 0 = 13;

N13 = 0 + 0 + 0 + 0 + 0 = 0;

D13 = P13/N13 = 13/0 = ∞ – принимаем.

P31 = 0 + 0 + 0 + 0 + 0 = 0;

N31 = 5 + 4 + 0 + 4 + 0 = 13;

D31 = P31/N31 = 0/13 = 0 < 1 – отбрасываем.

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

P14 = 5 + 4 + 0 + 4 + 1 = 14;

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

D14 = P14/N14 = 14/0 = ∞ – принимаем.

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

N41 = 5 + 4 + 0 + 4 + 1 = 14;

D41 = 0 < 1 – отбрасываем.

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

P15 = 0 + 4 + 0 + 0 + 1 = 5;

N15 = 5 + 0 + 0 + 0 + 0 = 5;

D15 = P15/N15 = 5/5 ≤ 1 – отбрасываем.

P51 = 5 + 0 + 0 + 0 + 0 = 5;

N51 =0 + 4 + 0 + 0 + 1 = 5;

D51 = P51/N51= 5/5 ≤ 1 – отбрасываем.

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

P16 = 0 + 0 + 0 + 4 + 0 = 4;

N16 = 5 + 0 + 0 + 0 + 1 = 6;

D16 = P16/N16 = 4/6 ≤ 1 – отбрасываем.

P61 = 5 + 0 + 0 + 0 + 1 = 6;

N61 = 0 + 0 + 0 + 4 + 0 = 4;

D61 = P61/N61 = 6/4 = 1.5 > 1 – принимаем.

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

P17 = 5 + 0 + 0 + 4 + 1 = 10;

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

D17 = P17/N17 = 10/0 = ∞ – принимаем.

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

N71 = 5 + 0 + 0 + 4 + 1 = 10;

D71 = P71/N71 = 0/10 < 1 – отбрасываем.

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

P18 = 0 + 4 + 0 + 4 + 0 = 8;

N18 = 5 + 0 + 0 + 0 + 0 = 5;

D18 = P18/N18 = 8/5 = 1.6 > 1 – принимаем.

P81 = 5 + 0 + 0 + 0 + 0 = 5;

N81 = 0 + 4 + 0 + 4 + 0 = 8;

D81 = P81/N81 = 5/8 < 1 – отбрасываем.

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

P19 = 0 + 4 + 0 + 4 + 1 = 9;

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

D19 = P19/N19 = 9/0 = ∞ – принимаем.

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

N91 = 0 + 4 + 0 + 4 + 1 = 9;

D91 = P91/N91 = 0/9 < 1 – отбрасываем.

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

P110 = 0 + 4 + 0 + 4 + 0 = 8;

N110 = 5 + 0 + 0 + 0 + 0 = 5;

D110 = P110/N110 = 8/5 = 1.6 > 1 – принимаем.

P101 = 5 + 0 + 0 + 0 + 0 = 5;

$N101 = 0 + 4 + 0 + 4 + 0 = 8$;

$D101 = P101/N101 = 5/8 < 1$ – отбрасываем.

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

$P23 = 0 + 4 + 0 + 0 + 0 = 4$;

$N23 = 0 + 0 + 2 + 0 + 0 = 2$;

$D23 = P23/N23 = 4/2 = 2 > 1$ – принимаем.

$P32 = 0 + 0 + 2 + 0 + 0 = 2$;

$N32 = 0 + 4 + 0 + 0 + 0 = 4$;

$D32 = P32/N32 = 2/4 < 1$ - отбрасываем.

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

$P24 = 0 + 4 + 0 + 0 + 1 = 5$;

$N24 = 0 + 0 + 2 + 0 + 0 = 2$;

$D24 = P24/N24 = 5/2 = 2.5 > 1$ – принимаем.

$P42 = 0 + 0 + 2 + 0 + 0 = 2$;

$N42 = 0 + 4 + 0 + 0 + 1 = 5$;

$D42 = P42/N42 = 2/5 < 1$ - отбрасываем.

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

$P25 = 0 + 0 + 0 + 0 + 1 = 1$;

$N25 = 5 + 0 + 2 + 4 + 0 = 11$;

$D25 = P25/N25 = 1/11 < 1$ - отбрасываем.

$P52 = 0 + 0 + 0 + 0 + 1 = 1$;

$N52 = 5 + 0 + 2 + 4 + 0 = 11$;

$D52 = P52/N52 = 11/1 = 11 > 1$ – принимаем.

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

$P26 = 0 + 0 + 0 + 4 + 0 = 4$;

$N26 = 5 + 4 + 2 + 0 + 1 = 12$;

$D26 = P26/N26 = 4/12 < 1$ - отбрасываем.

$P62 = 5 + 4 + 2 + 0 + 1 = 12$;

$N62 = 0 + 0 + 0 + 4 + 0 = 4$;

$D62 = P62/N62 = 12/4 = 3 > 1$ – принимаем.

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

$P27 = 0 + 0 + 0 + 0 + 1 = 1$;

$N27 = 0 + 4 + 2 + 0 + 0 = 6;$

$D27 = P27/N27 = 1/6 < 1$ - отбрасываем.

$P72 = 0 + 4 + 2 + 0 + 0 = 6;$

$N72 = 0 + 0 + 0 + 0 + 1 = 1;$

$D72 = P72/N72 = 6/1 = 6 > 1$ – принимаем.

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

$P28 = 0 + 4 + 0 + 0 + 0 = 4;$

$N28 = 5 + 0 + 2 + 0 + 0 = 7;$

$D28 = P28/N28 = 4/7 < 1$ - отбрасываем.

$P82 = 5 + 0 + 2 + 0 + 0 = 7;$

$N82 = 0 + 4 + 0 + 0 + 0 = 4;$

$D82 = P82/N82 = 7/4 = 1.75 > 1$ – принимаем.

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

$P29 = 0 + 4 + 0 + 4 + 1 = 9;$

$N29 = 5 + 0 + 2 + 0 + 0 = 7;$

$D29 = P29/N29 = 9/7 = 1.3 > 1$ – принимаем.

$P92 = 5 + 0 + 2 + 0 + 0 = 7;$

$N92 = 0 + 4 + 0 + 4 + 1 = 9;$

$D92 = P92/N92 = 7/9 = 1.75 < 1$ - отбрасываем.

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

$P210 = 0 + 4 + 0 + 4 + 0 = 8;$

$N210 = 5 + 0 + 2 + 0 + 0 = 7;$

$D210 = P210/N210 = 8/7 = 1.1$ с

$P102 = 5 + 0 + 2 + 0 + 0 = 7;$

$N102 = 0 + 4 + 0 + 4 + 0 = 8;$

$D102 = P102/N102 = 7/8 < 1$ - отбрасываем.

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

$P34 = 0 + 0 + 0 + 0 + 1 = 1;$

$N34 = 0 + 0 + 0 + 0 + 0 = 0;$

$D34 = P34/N34 = 1/0 = \infty$ – принимаем.

$P43 = 0 + 0 + 0 + 0 + 0 = 0;$

$N43 = 0 + 0 + 0 + 0 + 1 = 1;$

$D43 = P43/N43 = 0/1 < 1$ - отбрасываем.

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

$P35 = 0 + 0 + 0 + 0 + 1 = 1$;

$N35 = 5 + 4 + 0 + 4 + 0 = 13$;

$D35 = P35/N35 = 1/13 < 1$ - отбрасываем.

$P53 = 5 + 4 + 0 + 4 + 0 = 13$;

$N53 = 0 + 0 + 0 + 0 + 1 = 1$;

$D53 = P53/N53 = 13/1 = 13 > 1$ – принимаем.

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

$P36 = 0 + 0 + 0 + 4 + 0 = 4$;

$N36 = 5 + 4 + 0 + 0 + 1 = 10$;

$D36 = P36/N36 = 4/10 < 1$ - отбрасываем.

$P63 = 5 + 4 + 0 + 0 + 1 = 10$;

$N63 = 0 + 0 + 0 + 4 + 0 = 4$;

$D63 = P63/N63 = 10/4 = 2.5 > 1$ – принимаем.

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

$P37 = 0 + 0 + 0 + 0 + 1 = 1$;

$N37 = 0 + 4 + 0 + 0 + 0 = 4$;

$D37 = P37/N37 = 1/4 < 1$ - отбрасываем.

$P73 = 0 + 4 + 0 + 0 + 0 = 4$;

$N73 = 0 + 0 + 0 + 0 + 1 = 1$;

$D73 = P73/N73 = 4/1 = 4 > 1$ – принимаем.

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

$P38 = 0 + 0 + 0 + 0 + 0 = 0$;

$N38 = 5 + 0 + 0 + 0 + 0 = 5$;

$D38 = P38/N38 = 0/5 < 1$ - отбрасываем.

$P83 = 5 + 0 + 0 + 0 + 0 = 5$;

$N83 = 0 + 0 + 0 + 0 + 0 = 0$;

$D83 = P83/N83 = 5/0 = \infty$ – принимаем.

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

$P39 = 0 + 0 + 0 + 4 + 1 = 5$;

$N39 = 5 + 0 + 0 + 0 + 0 = 5$;

D39 = P39/N39 = 5/5 = 1 ≤ - отбрасываем.

P93 = 5 + 0 + 0 + 0 + 0 = 5;

N93 = 0 + 0 + 0 + 4 + 1 = 5;

D93 = P93/N93 = 5/5 = 1 ≤ - отбрасываем.

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

P310 = 0 + 0 + 0 + 0 + 0 = 0;

N310 = 5 + 0 + 0 + 0 + 0 = 5;

D310 = P310/N310 = 0/5 < 1 - отбрасываем.

P103 = 5 + 0 + 0 + 0 + 0 = 5;

N103 = 0 + 0 + 0 + 0 + 0 = 0;

D103 = P103/N103 = 5/0 = ∞ – принимаем.

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

P45 = 0 + 0 + 0 + 0 + 0 = 0;

N45 = 5 + 4 + 0 + 4 + 0 = 13;

D45 = P45/N45 = 0/13 < 1 - отбрасываем.

P54 = 5 + 4 + 0 + 4 + 0 = 13;

N54 = 0 + 0 + 0 + 0 + 0 = 0;

D54 = P54/N54 = 13/0 = ∞ – принимаем.

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

P46 = 0 + 0 + 0 + 4 + 0 = 4;

N46 = 5 + 4 + 0 + 0 + 1 = 10;

D46 = P46/N46 = 4/10 < 1 - отбрасываем.

P64 = 5 + 4 + 0 + 0 + 1 = 10;

N64 = 0 + 0 + 0 + 4 + 0 = 4;

D64 = P64/N64 = 10/4 = 2.5 > 1 – принимаем.

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

P47 = 0 + 0 + 0 + 0 + 1 = 1;

N47 = 0 + 4 + 0 + 0 + 0 = 4;

D47 = P47/N47 = 1/4 < 1 - отбрасываем.

P74 = 0 + 4 + 0 + 0 + 0 = 4;

N74 = 0 + 0 + 0 + 0 + 1 = 1;

D74 = P74/N74 = 4/1 = 4 > 1 – принимаем.

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

P48 = 0 + 0 + 0 + 0 + 0 = 0;

N48 = 5 + 0 + 0 + 0 + 1 = 6;

D48 = P48/N48 = 0/6 < 1 - отбрасываем.

P84 = 5 + 0 + 0 + 0 + 1 = 6;

N84 = 0 + 0 + 0 + 0 + 0 = 0;

D84 = P84/N84 = 6/0 = ∞ – принимаем.

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

P49 = 0 + 0 + 0 + 4 + 0 = 4;

N49 = 5 + 0 + 0 + 0 + 0 = 5;

D49 = P49/N49 = 4/5 < 1 - отбрасываем.

P94 = 5 + 0 + 0 + 0 + 0 = 5;

N94 = 0 + 0 + 0 + 4 + 0 = 4;

D94 = P94/N94 = 5/4 = 1.25 > 1 – принимаем.

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

P410 = 0 + 0 + 0 + 0 + 0 = 0;

N410 = 5 + 0 + 0 + 0 + 1 = 6;

D410 = P410/N410 = 0/6 < 1 - отбрасываем.

P104 = 5 + 0 + 0 + 0 + 1 = 6;

N104 = 0 + 0 + 0 + 0 + 0 = 0;

D104 = P104/N104 = 6/0 ∞ – принимаем.

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

P56 = 0 + 0 + 0 + 4 + 0 = 4;

N56 = 0 + 4 + 0 + 0 + 1 = 5;

D56 = P56/N56 = 4/5 < 1 - отбрасываем.

P65 = 0 + 4 + 0 + 0 + 1 = 5;

N65 = 0 + 0 + 0 + 4 + 0 = 4;

D65 = P65/N65 = 5/4 = 1.25 > 1 – принимаем.

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

P57 = 5 + 0 + 0 + 4 + 1 = 10;

N57 = 0 + 4 + 0 + 0 + 0 = 4;

D57 = P57/N57 = 10/4 = 2.5 > 1 – принимаем.

P75 = 0 + 4 + 0 + 0 + 0 = 4;

N75 = 5 + 0 + 0 + 4 + 1 = 10;

D75 = P75/N75 = 4/10 < 1 - отбрасываем.

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

P58 = 0 + 4 + 0 + 4 + 0 = 8;

N58 = 0 + 0 + 0 + 0 + 1 = 1;

D58 = P58/N58 = 8/1 = 8 >1 – принимаем.

P85 = 0 + 0 + 0 + 0 + 1 = 1;

N85 = 0 + 4 + 0 + 4 + 0 = 8;

D85 = P85/N85 = 1/8 < 1 - отбрасываем.

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

P59 = 5 + 4 + 0 + 4 + 0 = 13;

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

D59 = P59/N59 = 13/0 = ∞ – принимаем.

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

N95 = 5 + 4 + 0 + 4 + 0 = 13;

D95 = P95/N95 = 0/13 < 1 - отбрасываем.

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

P510 = 0 + 4 + 0 + 4 + 0 = 8;

N510 = 0 + 0 + 0 + 0 + 1 = 1;

D510 = P510/N510 = 8/1 = 8 >1 – принимаем.

P105 = 0 + 0 + 0 + 0 + 1 = 1;

N105 = 0 + 4 + 0 + 4 + 0 = 8;

D105 = P105/N105 = 1/8 < 1 - отбрасываем.

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

P67 = 5 + 0 + 0 + 0 + 1 = 6;

N67 = 0 + 0 + 0 + 4 + 0 = 4;

D67 = P67/N67 = 6/4 = 1.5 > 1 – принимаем.

P76 = 0 + 0 + 0 + 4 + 0 = 4;

N76 = 5 + 0 + 0 + 0 + 1 = 6;

D76 = P76/N76 = 6/4 < 1 - отбрасываем.

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

P68 = 0 + 4 + 0 + 0 + 1 = 5;

N68 = 0 + 0 + 0 + 4 + 0 = 4;

D68 = P68/N68 = 5/4 = 1.25 > 1 – принимаем.

P86 = 0 + 0 + 0 + 4 + 0 = 4;

N86 = 0 + 4 + 0 + 0 + 1 = 5;

D86 = P86/N86 = 4/5 < 1 - отбрасываем.

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

P69 = 5 + 4 + 0 + 0 + 1 + 10;

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

D69 = P69/N69 = 10/0 = ∞ – принимаем.

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

N96 = 5 + 4 + 0 + 0 + 1 + 10;

D96 = P96/N96 = 0/10 < 1 - отбрасываем.

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

P610 = 0 + 4 + 0 + 0 + 1 = 5;

N610 = 0 + 0 + 0 + 4 + 0 = 4;

D610 = P610/N610 = 5/4 = 1.25 > 1 – принимаем.

P106 = 0 + 0 + 0 + 4 + 0 = 4;

N106 = 0 + 4 + 0 + 0 + 1 = 5;

D106 = P106/N106 = 4/5 < 1 - отбрасываем.

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

P78 = 0 + 4 + 0 + 0 + 0 = 4;

N78 = 5 + 0 + 0 + 0 + 1 = 6;

D78 = P78/N78 = 4/6 < 1 - отбрасываем.

P87 = 5 + 0 + 0 + 0 + 1 = 6;

N87 = 0 + 4 + 0 + 0 + 0 = 4;

D87 = P87/N87 = 6/4 = 1.5 > 1 – принимаем.

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

P79 = 0 + 4 + 0 + 4 + 0 = 8;

N79 = 5 + 0 + 0 + 0 + 1 = 6;

D79 = P79/N79 = 8/6 =1.3 > 1 – принимаем.

P97 = 5 + 0 + 0 + 0 + 1 = 6;

N97 = 0 + 4 + 0 + 4 + 0 = 8;

D97 = P97/N97 = 6/8 < 1 - отбрасываем.

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

P710 = 0 + 4 + 0 + 0 + 0 = 4;

N710 = 5 + 0 + 0 + 0 + 1 = 6;

D710 = P710/N710 = 4/6 < 1 - отбрасываем.

P107 = 5 + 0 + 0 + 0 + 1 = 6;

N107 = 0 + 4 + 0 + 0 + 0 = 4;

D107 = P107/N107 = 6/4 = 1.5 > 1 – принимаем.

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

P89 = 5 + 0 + 0 + 4 + 1 = 10;

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

D89 = P89/N89 = 10/0 = ∞ – принимаем.

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

N98 = 5 + 0 + 0 + 4 + 1 = 10;

D98 = P98/N98 = 0/10 < 1 - отбрасываем.

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

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

N810 = 0 + 0 + 0 + 0 + 0 = 0;

D810 = P810/N810 = 0/0 = 0 < 1 - отбрасываем.

P108 = 0 + 0 + 0 + 0 + 0 = 0;

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

D108 = P108/N108 = 0/0 = 0 < 1 - отбрасываем.

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

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

N910 = 5 + 0 + 0 + 4 + 1 = 10;

D910 = P910/N910 = 0/10 < 1 - отбрасываем.

P109 = 5 + 0 + 0 + 4 + 1 = 10;

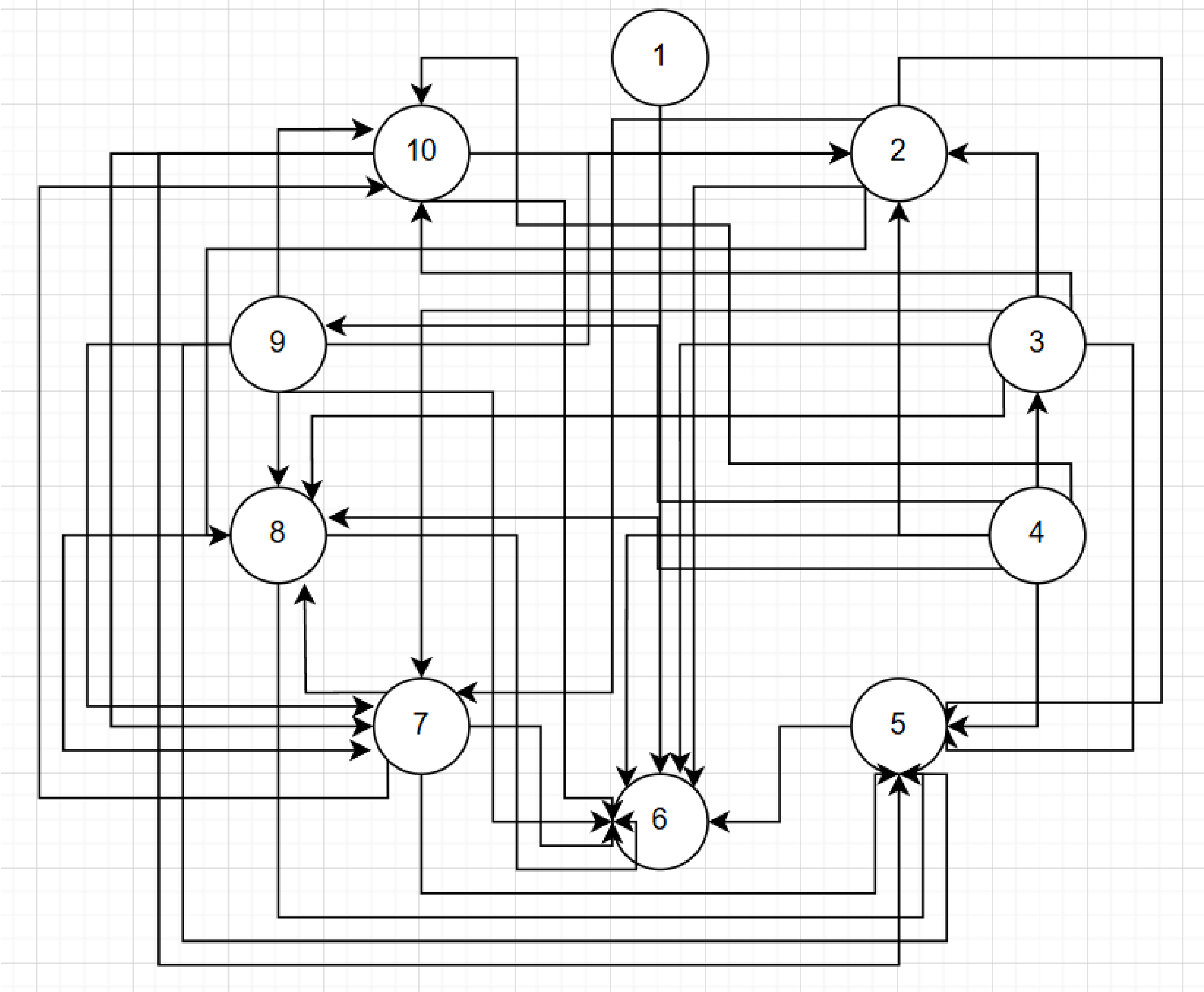
N109 = 0 + 0 + 0 + 0 + 0 = 0;

D109 = P109/N109 = 10/0 = ∞ – 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	-	-	-	-	1.5	-	-	-	-
2	∞	x	-	-	11	3	6	1.75	-	-
3	∞	2	x	-	13	2.5	4	∞	-	∞
4	∞	2.5	∞	x	∞	2.5	4	∞	1.25	∞
5	-	-	-	-	x	1.25	-	-	-	-
6	-	-	-	-	-	x	-	-	-	-
7	∞	-	-	-	2.5	1.5	x	1.5	-	1.5
8	1.6	-	-	-	8	1.25	-	x	-	-
9	∞	1.3	-	-	∞	∞	1.3	∞	x	∞
10	1.6	1.1	-	-	8	1.25	-	-	-	x



По матрице построен граф предпочтений (Рисунок 2.1).

Рисунок 2.1 – Вид графа предпочтений

Назначен порог отбора предпочтений C = 4 (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=4

	1	2	3	4	5	6	7	8	9	10
1	x	-	-	-	-	-	-	-	-	-
2	∞	x	-	-	11	-	6	-	-	-
3	∞	-	x	-	13	-	4	∞	-	∞
4	∞	-	∞	x	∞	-	4	∞	-	∞
5	-	-	-	-	x	-	-	-	-	-
6	-	-	-	-	-	x	-	-	-	-
7	∞	-	-	-	-	-	x	-	-	-
8	-	-	-	-	8	-	-	x	-	-
9	∞	-	-	-	∞	∞	-	∞	x	∞
10	-	-	-	-	8	-	-	-	-	x

По этой матрице построен граф предпочтений (Рисунок 2.2).

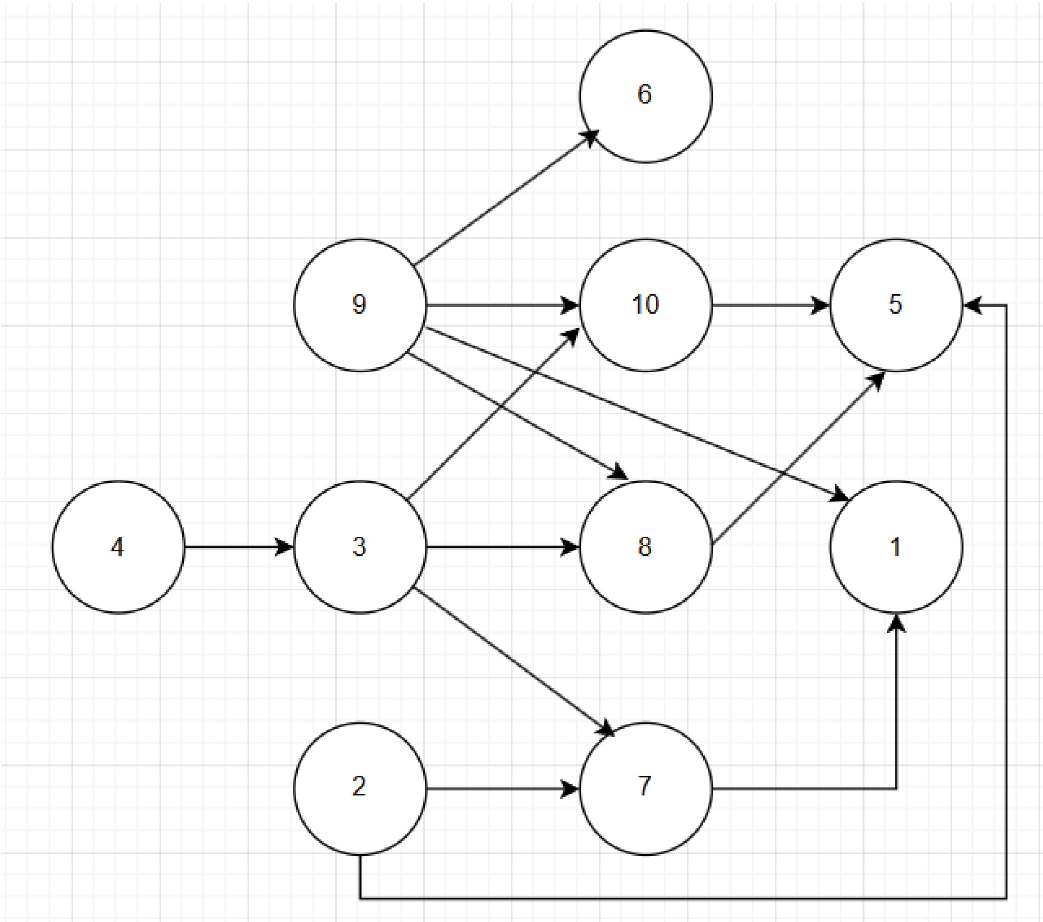


Рисунок 2.2 – Вид графа предпочтений для случая порога принятия решений С = 4

Петель в графе нет, при этом граф остался целостным.

Решение говорит нам о том, что первое место делят VPN-сервисы 4, 2 и 9. На втором месте расположены VPN-сервисы 3 и 6, на третьем– 7, на четвертом - 8 и 10. На пятом месте находится VPN-сервис с номером 1, а на шестом - 5.

2.3 Результат работы программы

Результат работы программы показан ниже (Рисунки 2.3, 2.4).

Матрица предпочтений:

	1	2	3	4	5	6	7	8	9	10
1	0.0	0.0	0.0	0.0	0.0	1.5	0.0	0.0	0.0	0.0
2	inf	0.0	0.0	0.0	11.0	3.0	6.0	1.75	0.0	1.75
3	inf	2.0	0.0	0.0	13.0	2.5	4.0	inf	0.0	inf
4	inf	2.5	inf	0.0	inf	2.5	4.0	inf	1.25	inf
5	0.0	0.0	0.0	0.0	0.0	1.25	0.0	0.0	0.0	0.0
6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
7	inf	0.0	0.0	0.0	2.5	1.5	0.0	1.5	0.0	1.5
8	1.6	0.0	0.0	0.0	8.0	1.25	0.0	0.0	0.0	inf
9	inf	1.29	0.0	0.0	inf	inf	1.33	inf	0.0	inf
10	1.6	0.0	0.0	0.0	8.0	1.25	0.0	inf	0.0	0.0

Рисунок 2.3 – Вывод матрицы предпочтений.

Матрица предпочтений с порогом C = 4

	1	2	3	4	5	6	7	8	9	10
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	inf	0.0	0.0	0.0	11.0	0.0	6.0	0.0	0.0	0.0
3	inf	0.0	0.0	0.0	13.0	0.0	4.0	inf	0.0	inf
4	inf	0.0	inf	0.0	inf	0.0	4.0	inf	0.0	inf
5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
7	inf	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
8	0.0	0.0	0.0	0.0	8.0	0.0	0.0	0.0	0.0	inf
9	inf	0.0	0.0	0.0	inf	inf	0.0	inf	0.0	inf
10	0.0	0.0	0.0	0.0	8.0	0.0	0.0	inf	0.0	0.0

Рисунок 2.4 – Вывод матрицы предпочтений с порогом, вывод альтернатив от лучшей к худшей.

2.4 Вывод по методу Электра II

В результате работы можно выделить ряд преимуществ метода Электра II. Метод позволяет учитывать, как количественные, так и качественные критерии. Что делает его подходящим для большего количества задач, в отличие от метода Парето. В методе используются пороговые значения для критериев, что помогает избавится от циклов, исключить слабые альтернативы и сосредоточиться на более значимых.

Однако, метод Электра II не лишен недостатков. Субъективная оценка весов критериев может привести к искажению результатов. К тому же метод требует значительных вычислительных ресурсов, особенно при большом количестве альтернатив и критериев.

3 МЕТОД АНАЛИЗА ИЕРАРХИЙ

Метод анализа иерархий основан на декомпозиции задачи и представлении ее в виде иерархической структуры: цель – критерии – альтернативы. Такой подход позволяет включить в иерархию все знания лица, принимающего решение знания. В методе может быть выявлена относительная степень взаимодействия элементов в иерархии, которые затем выражаются численно. МАИ включает процедуры синтеза множественных суждений, получения приоритетности критериев и нахождения альтернативных решений.

Весь процесс решения подвергается проверке на каждом этапе, что позволяет проводить оценку качества полученного решения. Решение многокритериального выбора основано на трех основных этапах:

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

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

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

3.1 Постановка задачи

Задача практической работы: выбрать лучший VPN-сервис.

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. Представим задачу по выбору VPN-сервиса в виде иерархии на Рисунке 3.1.

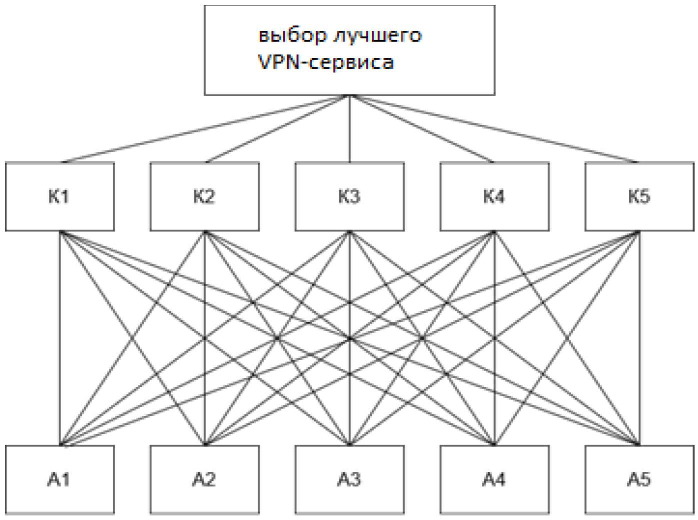


Рисунок 3.1 – Полная доминантная иерархия.

Критерии:

К 1 – Средняя цена (руб.) (-);

К 2 – Среднее время ответа сервиса (мин) (-);

К 3 – Рейтинг на VirusTotal (+);

К 4 – Количество отзывов (+);

К 5 – Удаленность от дата-центра (м) (-).

Альтернативы:

А 1 - OutlineVPN;

А 2 – ДядяВаня VPN;

А 3 – TunelBear VPN;

А 4 – VPN Secure;

А 5 – VPN Master.

3.3 Установка приоритетов критериев

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

Таблица 3.1 – Шкала относительной важности.

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

Шкала содержит соответствующие обратные значения.

3.4 Синтез приоритетов

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

Таблица 2 – Матрица парного сравнения критериев.

Цель	К 1	К 2	К 3	К 4	К 5	Vi	W_{2i}
К 1	1	1	3	6	8	2.702	0.39
К 2	1	1	2	5	6	2.268	0.327
К 3	1/3	1/2	1	3	5	1.201	0.173
К 4	1/6	1/5	1/3	1	2	0.467	0.067
К 5	1/8	1/6	1/5	1/2	1	0.382	0.042
$\sum$Vi						7.048	

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

Строка № 1

$V_1=(1x1x3x6x8)^{1/5} = 2.702$;

Строка № 2

$V_2=(1x1x2x5x6)^{1/5} = 2.268$;

Строка № 3

$V_3=(1/3x1/2x1x3x5)^{1/5} = 1.201$;

Строка № 4

$V_4=(1/6x1/5x1/3x1x2)^{1/5} = 0.467$;

Строка № 5

$V_5=(1/8x1/6x1/5x1/2x1\ )^{1/5} = 0.401$.

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

$\sum Vi = V_1 + V_2 + V_3 + V_4 + V_5 = 2.702 + 2.268 + 1.201 + 0.476 + 0.401 = 7.048$.

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

Строка № 1

$W_{21}= 2.702 / 7.048= 0.39 =Y_{21}$;

Строка № 2

$W_{22}= 2.268 / 7.048= 0.327 = Y_{22}$;

Строка № 3

$W_{23}= 1.201 / 7.048= 0.173 = Y_{23}$;

Строка № 4

$W_{24}= 0.476 / 7.048= 0.067 = Y_{24}$;

Строка № 5

$W_{25}= 0.401 / 7.048= 0.042 = Y_{25}$.

В результате получен вектор приоритетов:

$W_{2i} = (Y_{21}; Y_{22}; Y_{23}; Y_{24}; Y_{25}) = (0.39; 0.327; 0.173; 0.067; 0{,}042)$, где индекс 2 означает, что вектор приоритетов относится ко второму уровню иерархии.

Таблица 3.3 – Матрица сравнения по критерию 1.

К1	А1	А2	А3	А4	А5	V_{K1Y}	W_{3K1Y}

A1	1	5	6	9	2	3.519	0.444
A2	1/5	1	3	4	1/7	0.807	0.102
A3	1/6	1/3	1	3	1/8	0.461	0.058
A4	1/9	1/4	1/3	1	1/7	0.256	0.034
A5	1/2	7	8	7	1	2.952	0.363
$\sum V_{K1Y}$						7.996	

Определена относительная ценность каждого элемента.

Строка № 1

$V_{K11}=(1x5x6x9x2)^{1/5}= 3.519$;

Строка № 2

$V_{K12}=(1/5x1x3x4x1/7)^{1/5}= 0.807$;

Строка № 3

$V_{K13}=(1/6x1/3x1x3x1/8)^{1/5}= 0.461$;

Строка № 4

$V_{K14}=(1/9x1/4x1/3x1x1/8)^{1/5}= 0.256$;

Строка № 5

$V_{K15}=(1/2x7x8x8x1)^{1/5}= 2.952$.

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

$\sum V_{K1Y} = V_{K11} + V_{K12} + V_{K13} + V_{K14} + V_{K15} = 3.519 + 0.807 + 0.461 + 0.256 + 2.952 = 7.996$

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

Строка № 1

$W_{3K11}= 3.519 / 7.996= 0.444$

Строка № 2

$W_{3K12}= 0.807 / 7.996= 0.102$

Строка № 3

$W_{3K13}= 0.461 / 7.996= 0.058$

Строка № 4

$W_{3K14}= 0.265 / 7.996= 0.034$

Строка № 5

$W_{3K15}= 2.722 / 7.996= 0.363$

В результате получаем вектор приоритетов:

W_{3K1Y} = (Y_{311}; Y_{312}; Y_{313}; Y_{314}; Y_{315}) = (0.444; 0.102; 0.058; 0.034; 0.363),

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

Таблица 3.4 – Матрица сравнения по критерию 2.

К2	A1	A2	A3	A4	A5	VК2Y	W3К2Y
A1	1	2	4	4	2	2.297	0.407
A2	1/2	1	2	2	1	1.149	0.203
A3	1/4	1/2	1	1	2	0.758	0.134
A4	1/4	1/2	1	1	1/2	0.574	0.102
A5	1/2	1	1/2	2	1	0.871	0.154
∑VК2Y						5.649	

Определена относительная ценность каждого элемента.

Строка № 1

V_{K21}= $(1x2x4x4x2)^{1/5}$= 2.297;

Строка № 2

V_{K22}= $(1/2x1x2x2x1)^{1/5}$= 1.149;

Строка № 3

V_{K23}=$(1/4x1/2x1x1x2)^{1/5}$= 0.758;

Строка № 4

V_{K24}=$(1/4x1/2x1x1x1/2)^{1/5}$= 0.574;

Строка № 5

V_{K25}=$(1/2x1x1/2x2x1)^{1/5}$= 0.871.

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

$\sum V_{K2Y}$ = V_{K21} + V_{K22} + V_{K23} + V_{K24} + V_{K25} = 2.297+1.149+0.758+0.574+0.871 = 5.649.

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

Строка № 1

W_{3K21}= 2.297 / 5.649 = 0.407;

Строка № 2

W_{3K22}= 1.149/ 5.649 = 0.203;

Строка № 3

W_{3K23}= 0.758 / 5.649 = 0.134;

Строка № 4

W_{3K24}= 0.574 / 5.649 = 0.102;

Строка № 5

W_{3K25}= 0.871/ 5.649 = 0.154.

В результате получаем вектор приоритетов:

W_{3K2Y} = (Y_{321}; Y_{322}; Y_{323}; Y_{324}; Y_{325}) = (0.407; 0.203; 0.134; 0.102; 0.154),

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

К 3 – рейтинг по отзывам пользователей (Таблица 3.5);

Таблица 3.5 – Матрица сравнения по критерию 3

К3	А1	А2	А3	А4	А5	V_{K3Y}	W_{3K3Y}
А1	1	7	2	3	3	2.63	0.402
А2	1/7	1	1/7	1/6	1/6	0.224	0.034
А3	1/2	7	1	2	2	1.695	0.259
А4	1/3	6	1/2	1	1	1	0.153
А5	1/3	6	1/2	1	1	1	0.153
V_{K35}						6.549	

Определена относительная ценность каждого элемента.

Строка № 1

V_{K31}=(1x7x2x3x3)$^{1/5}$= 2.63 ;

Строка № 2

V_{K32}=(1/7x1x1/7x1/6x1/6)$^{1/5}$= 0.224;

Строка № 3

V_{K33}=(1/2x7x1x2x2)$^{1/5}$= 1.695;

Строка № 4

V_{K34}=(1/3x6x1/2x1x1)$^{1/5}$= 1;

Строка № 5

V_{K35}=(1/3x6x1/2x1x1)$^{1/5}$= 1.

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

$\sum V_{K3Y}$ = V_{K31} + V_{K32} + V_{K33} + V_{K34} + V_{K35} = 2.63+0.224+1.695+1+1 = 6.549.

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

Строка № 1

W_{3K31}= 2.63/6.549= 0.402;

Строка № 2

W_{3K32}= 0.224/6.549= 0.034;

Строка № 3

W_{3K33}= 1.695/6.549= 0.259;

Строка № 4

W_{3K34}= 1/6.549= 0.153;

Строка № 5

W_{3K35}= 1/6.549= 0.153.

В результате получаем вектор приоритетов:

W_{3K3Y} = (Y331; Y332; Y333; Y334; Y335) = (0.402; 0.034; 0.259; 0.153; 0.153), где индекс 3 означает, что вектор приоритетов относится к третьему уровню иерархии критерия К3.

К 4 – удаленность от ближайшего сервера (Таблица 3.6);

Таблица 3.6 – Матрица сравнения по критерию 4

К4	A1	A2	A3	A4	A5	V_{K4Y}	W_{3K4Y}
A1	1	5	4	5	2	2.885	0.471
A2	1/5	1	1	1	1/2	0.631	0.103
A3	1/4	1	1	1	1/2	0.66	0.108
A4	1/5	1	1	1	1/2	0.631	0.103
A5	1/2	2	2	2	1	1.32	0.215
$\sum V_{K4Y}$						6.127	

Определена относительная ценность каждого элемента.

Строка № 1

$V_{K41}=(1x5x4x5x2)^{1/5} = 2.885$;

Строка № 2

$V_{K42}=(1/5x1x1x1x1/2)^{1/5} = 0.631$;

Строка № 3

$V_{K43}=(1/4x1x1x1x1/2)^{1/5} = 0.66$;

Строка № 4

$V_{K44}=(1/5x1x1x1x1/2)^{1/5} = 0.631$;

Строка № 5

$V_{K45}=(1/2x2x2x2x1)^{1/5} = 1.32$.

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

$\sum V_{K4Y} = V_{K41} + V_{K42} + V_{K43} + V_{K44} + V_{K45} = 2.885 + 0.631 + 0.66 + 0.631 + 1.32 = 6.127$.

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

Строка № 1

$W_{3K41}= 2.885/ 6.127= 0.47$

Строка № 2

$W_{3K42}=0.631/ 6.127= 0.103$;

Строка № 3

$W_{3K43}= 0.66 / 6.127= 0.108$;

Строка № 4

$W_{3K44}= 0.631/ 6.127= 0.103$;

Строка № 5

$W_{3K45}=1.32/ 6.127= 0.215$.

В результате получаем вектор приоритетов:

$W_{3K4Y} = (Y_{341}; Y_{342}; Y_{343}; Y_{344}; Y_{345}) = (0.47; 0.103; 0.108; 0.103; 0.215)$, где индекс 3 означает, что вектор приоритетов относится к третьему уровню иерархии критерия К4.

К 5 – удаленность от сервера (Таблица 3.7).

Таблица 3.7 – Матрица сравнения по критерию 5

К5	А1	А2	А3	А4	А5	V_{K5Y}	W_{3K5Y}
А1	1	3	3	5	2	2.46	0.432
А2	1/3	1	2	1/2	1	0.803	0.141
А3	1/3	1/2	1	2	1	0.803	0.141
А4	1/5	2	1/2	1	1/2	0.631	0.111
А5	1/2	1	1	2	1	1	0.176
$\sum V_{K5Y}$				5.697			

Определена относительная ценность каждого элемента.

Строка № 1

$V_{K51}=(1x3x3x5x2)^{1/5} = 2.46$;

Строка № 2

$V_{K52}=(1/3x1x2x1/2x1)^{1/5} = 0.803$;

Строка № 3

$V_{K53}=(1/3x1/2x1x2x1)^{1/5} = 0.803$;

Строка № 4

$V_{K54}=(1/5x2x1/2x1x1/2)^{1/5} = 0.631$;

Строка № 5

$V_{K55}=(1/2x1x1x2x1)^{1/5} = 1$.

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

$\sum V_{K5Y} = V_{K51} + V_{K52} + V_{K53} + V_{K54} + V_{K55} = 2.46+0.803+0.803+0.631+1= 5.697$.

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

Строка № 1

$W_{3K51}= 2.46 / 5.697= 0.432$;

Строка № 2

$W_{3K52}=0.803/ 5.697= 0.141$;

Строка № 3

$W_{3K53}= 0.803/ 5.697= 0.141$;

Строка № 4

$W_{3K54}= 0.631/ 5.697= 0.111$;

Строка № 5

$W_{3K55}= 1/ 5.697= 0.176$.

В результате получаем вектор приоритетов:

$W_{3K5Y} = (Y_{351}; Y_{352}; Y_{353}; Y_{354}; Y_{355}) = (0.432; 0.141; 0.141; 0.111; 0.176)$,

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 лучшего VPN-сервиса» (Таблица 3.8).

Таблица 3.8 – Матрица «Выбор лучшего VPN-сервиса»

Цель	К 1	К 2	К 3	К 4	К 5	W_{2i}
К 1	1	1	3	6	8	0.383
К 2	1	1	2	5	6	0.327
К 3	1/3	1/2	1	3	5	0.173
К 4	1/6	1/5	1/3	1	2	0.067
К 5	1/8	1/6	1/5	1/2	1	0.042

Определена сумма каждого столбца матрицы суждений.

$S_1 = 1 + 1 + 1/3 + 1/6 + 1/8 = 2.625$;

$S_2 = 1 + 1 + 1/2 + 1/5 + 1/6 = 2.867$;

$S_3 = 3 + 2 + 1 + 1/3 + 1/ 5= 6.533$;

$S_4 = 6 + 5 + 3 + 1 + 1/2 = 15.5$;

$S_5 = 8 + 6 + 5 + 2 + 1= 22$.

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

$P_1 = S_1 x W_{21} = 2.625x 0.383 = 1.015$;

$P_2 = S_2 x W_{22} = 2.867x0.327= 0.938$;

$P_3 = S_3 x W_{23} =6.533x0.173= 1.13$;

$P_4 = S_4 x W_{24} =15.5x0.067 = 1.039$;

$P_5 = S_5 x W_{25} =22x0.042 = 0.924$.

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

$$\lambda_{max} = P_1 + P_2 + P_3 + P_4 + P_5 = 1.015+0.938+1.13+1.039+0.924 = 5.046.$$

Отклонение от согласованности выражается индексом согласованности.

$$ИС = (\lambda_{max} - n)/(n - 1) = (5.046-5)/(5-1) = 0.012.$$

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

$$ОС = ИС/СИ = 0.012/1.12 = 0.013.$$

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

Определены индекс согласованности и отношение согласованности для матрицы К 1 Таблица 3.9).

Таблица 3.9 – Матрица сравнения по критерию 1

К1	А1	А2	А3	А4	А5	W_{3K1Y}
А1	1	5	6	9	2	0.444
А2	1/5	1	3	4	1/7	0.102
А3	1/6	1/3	1	3	1/8	0.058
А4	1/9	1/4	1/3	1	1/7	0.034
А5	1/2	7	8	7	1	0.363

Определяется сумма каждого столбца матрицы суждений.

$S_{1K1} = 1 + 1/5 + 1/6 + 1/9 + 1/2 = 1.978;$

$S_{2\,K1} = 5 + 1 + 1/3 + 1/4 + 7 = 13.583;$

$S_{3\,K1} = 6 + 3 + 1 + 1/3 + 8 = 18.333;$

$S_{4\,K1} = 9 + 4 + 3 + 1 + 8 = 25;$

$S_{5\,K1} = 2 + 1/7 + 1/8 + 1/8 + 1 = 3.393.$

Затем полученный результат умножен на компоненту нормализованного вектора приоритетов.

$P_{1\,K1} = S_1 \text{ x } W_{3K11} = 1.811\text{x } 0.444= 0.796;$

$P_{2\,K1} = S_2 \text{ x } W_{3K12} = 13.583\text{x}0.102 = 1.372;$

$P_{3\,K1} = S_3 \text{ x } W_{3K13} =18.333\text{x}0.058 = 1.063;$

$P_{4\,K1} = S_1 \text{ x } W_{3K14} = 25\text{x}0.034 = 0.825;$

$P_{5\,K1} = S_1 \text{ x } W_{3K15} = 3.393\text{x}0.363 = 1.154.$

Найдена пропорциональность предпочтений.

$$\lambda_{max\ K1} = P_{1K1} + P_{2K1} + P_{3K1} + P_{4K1} + P_{5K1} = 0.796+1.372+1.063+0.825+1.154 = 5.21.$$

Отклонение от согласованности выражается индексом согласованности.

$$ИС_{К1} = (\lambda_{max\ К1} - n)/(n - 1) = (5.21 - 5)/(5-1) = 0.053.$$

Найдено отношение согласованности ОС.

$$ОС_{К1} = ИС/СИ = 0.053/1.12 = 0.084.$$

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

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

Таблица 3.10 – Матрица сравнения по критерию 2

К2	А1	А2	А3	А4	А5	W_{3K2Y}
А1	1	2	4	4	2	0.407
А2	1/2	1	2	2	1	0.203
А3	1/4	1/2	1	1	2	0.134
А4	1/4	1/2	1	1	1/2	0.102
А5	1/2	1	1/2	2	1	0.154

$S_{1K2} = 1 + 1/2 + 1/4 + 1/4 + 1/2 = 2.5;$

$S_{2\ K2} = 2 + 1 + 1/2 + 1/2 + 1 = 5;$

$S_{3\ K2} = 4 + 2 + 1 + 1 + 1/2 = 8.5;$

$S_{4\ K2} = 4 + 2 + 1 + 1 + 2 = 10;$

$S_{5\ K2} = 2 + 1 + 2 + 1/2 + 1 = 6.5.$

Затем полученный результат умножен на компоненту нормализованного вектора приоритетов.

$P_{1\ K2} = S_1 \text{ x } W_{3\ K21} = 2.5\text{x}0.407 = 1.018;$

$P_{2\ K2} = S_2 \text{ x } W_{3\ K22} = 5\text{x}0.203 = 1.015;$

$P_{3\ K2} = S_3 \text{ x } W_{3\ K23} = 8.5\text{x}0.134 = 1.139;$

$P_{4\ K2} = S_4 \text{ x } W_{3\ K24} =10\text{x}0.102 = 1.02;$

$P_{5\ K2} = S_5 \text{ x } W_{3\ K25} = 6.5\text{x}0.154 = 1.001.$

Найдена пропорциональность предпочтений.

$$\lambda_{max\ K2} = P_{1K2} + P_{2K2} + P_{3K2} + P_{4K2} + P_{5K2} = 1.018 + 1.015 + 1.139 + 1.02 + 1.001 = 5.193.$$

Отклонение от согласованности выражается индексом согласованности.

$$ИС_{К2} = (\lambda_{max\ К2} - n)/(n - 1) = (5.193 - 5)/(5-1) = 0.048.$$

Найдено отношение согласованности ОС.

$$ОС_{К2} = ИС/СИ = 0.048/1.12 = 0{,}054.$$

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

Определены индекс согласованности и отношение согласованности для матрицы К 3 – рейтинг по отзывам пользователей (Таблица 3.11).

Таблица 3.11 – Матрица сравнения по критерию 3

К3	А1	А2	А3	А4	А5	$W_{3К3Y}$
А1	1	7	2	3	3	0.402
А2	1/7	1	1/7	1/6	1/6	0.034
А3	1/2	7	1	2	2	0.259
А4	1/3	6	1/2	1	1	0.153
А5	1/3	6	1/2	1	1	0.153

Определена сумма каждого столбца матрицы суждений.

$S_{1К3} = 1 + 1/7 + 1/2 + 1/3 + 1/3 = 2.31;$

$S_{2\ К3} = 7 + 1 + 7 + 6 + 6 = 27;$

$S_{3\ К3} = 2 + 1/7 + 1 + 1/2 + 1/2 =4.143;$

$S_{4\ К3} = 3 + 1/6 + 2 + 1 + 1 =7.167;$

$S_{5\ К3} = 3 + 1/6 + 2 + 1 + 1 = 7.167.$

Затем полученный результат умножен на компоненту нормализованного вектора приоритетов.

$P_{1\ К3} = S_1 \times W_{3\ К31} = 2.31 \times 0.402 = 0.929;$

$P_{2\ К3} = S_2 \times W_{3\ К32} =27 \times 0.034 = 0.918;$

$P_{3\ К3} = S_3 \times W_{3\ К33} =4.143 \times 0.259 = 1.073;$

$P_{4\ К3} = S_4 \times W_{3\ К34} =7.167 \times 0.153=1.097;$

$P_{5\ К3} = S_5 \times W_{3\ К35} = 7.167 \times 0.153=1.097.$

Найдем пропорциональность предпочтений.

$$\lambda_{max\ К3} = P_{1К3} + P_{2К3} + P_{3К3} + P_{4К3} + P_{5К3} = 0.929 + 0.918+ 1.073+ 1.097+ 1.097= 5.114.$$

Отклонение от согласованности выражается индексом согласованности.

$$ИС_{К3} = (\lambda_{max\ К3} - n)/(n - 1) = (5.114 - 5)/(5 - 1) = 0.029.$$

Найдено отношение согласованности ОС.

$$ОС_{К3} = ИС/СИ = 0.029/1.12 = 0.026.$$

Значение ОС меньше или равное 0.10 считается приемлемым, значит матрица К 3 (рейтинг по отзывам пользователей) согласована.

Определены индекс согласованности и отношение согласованности для матрицы К 4 – удаленность от ближайшей станции метро (Таблица 12).

Таблица 3.12 – Матрица сравнения по критерию 4

К4	А1	А2	А3	А4	А5	W_{3K4Y}
А1	1	5	4	5	2	0.47
А2	1/5	1	1	1	1/2	0.103
А3	1/4	1	1	1	1/2	0.108
А4	1/5	1	1	1	1/2	0.103
А5	1/2	2	2	2	1	0.215

Определена сумма каждого столбца матрицы суждений.

$S_{1К4} = 1 + 1/5 + 1/4 + 1/5 + 1/2 = 2.15;$

$S_{2К4} = 5 + 1 + 1 + 1 + 2 = 10;$

$S_{3К4} = 4 + 1 + 1 + 1 + 2 = 9;$

$S_{4К4} = 5 + 1 + 1 + 1 + 2 = 10;$

$S_{5К4} = 2 + 1/2 + 1/2 + 1/2 + 1 = 4.5.$

Затем полученный результат умножен на компоненту нормализованного вектора приоритетов.

$P_{1К4} = S_1 x W_{3\ К41} = 2.15x0.47 = 1.011;$

$P_{2К4} = S_2 x W_{3\ К42} = 10x0.103 = 1.03;$

$P_{3К4} = S_3 x W_{3\ К43} =9x0.108= 0.972;$

$P_{4К4} = S_4 x W_{3\ К44} = 10x0.103 = 1.03;$

$P_{5К4} = S_5 x W_{3\ К45} = 4.5x0.215 = 0.968.$

Найдена пропорциональность предпочтений.

$\lambda_{max\ К4} = P_{1К4} + P_{2К4} + P_{3К4} + P_{4К4} + P_{5К4} = 1.011 + 1.03 + 0.972 + 1.03 + 0.968 = 5.011.$

Отклонение от согласованности выражается индексом согласованности.

$$ИС_{К4} = (\lambda_{max\ К4} - n)/(n - 1) = (5.011 - 5)/(5 - 1) = 0.028.$$

Найдено отношение согласованности ОС.

$$ОС_{К4} = ИС/СИ = 0.028/1.12 = 0.025.$$

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

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

Таблица 3.13 – Матрица сравнения по критерию 5

К5	А1	А2	А3	А4	А5	W_{3K5Y}
А1	1	3	3	5	2	0.432
А2	1/3	1	2	1/2	1	0.141
А3	1/3	1/2	1	2	1	0.141
А4	1/5	2	1/2	1	1/2	0.111
А5	1/2	1	1	2	1	0.176

Определена сумма каждого столбца матрицы суждений.

$S_{1K5} = 1 + 1/3 + 1/3 + 1/5 + 1/2 = 2.367$;

$S_{2K5} = 3 + 1 + 1/2 + 2 + 1 = 7.5$;

$S_{3K5} = 3 + 2 + 1 + 1/2 + 1 = 7.5$;

$S_{4K5} = 5 + 1/2 + 2 + 1 + 2 = 10.5$;

$S_{5K5} = 2 + 1 + 1 + 1/2 + 1 = 5.5$.

Затем полученный результат умножен на компоненту нормализованного вектора приоритетов.

$P_{1K5} = S_1 \text{ x } W_{3\ K41} = 2.367\text{x}0.432 = 1.023$;

$P_{2K5} = S_2 \text{ x } W_{3\ K42} = 7.5\text{x}0.141 = 1.058$;

$P_{3K5} = S_3 \text{ x } W_{3\ K43} = 7.5\text{x}0.141 = 1.058$;

$P_{4K5} = S_1 \text{ x } W_{3\ K44} = 10.5\text{x}0.111 = 1.166$;

$P_{5K5} = S_1 \text{ x } W_{3\ K45} = 5.5\text{x}0.176 = 0.968$.

Найдена пропорциональность предпочтений.

$$\lambda_{max\ K5} = P_{1K5} + P_{2K5} + P_{3K5} + P_{4K5} + P_{5K5} = 1.023 + 1.058 + 1.058 + 1.666 + 0.968 = 5.273.$$

Отклонение от согласованности выражается индексом согласованности.

$$ИС_{K5} = (\lambda_{max\ K5} - n)/(n - 1) = (5.273 - 5)/(5 - 1) = 0.068.$$

Найдено отношение согласованности ОС.

$$ОС_{K5} = ИС/СИ = 0.068/1.12 = 0.061.$$

Значение ОС меньше или равное 0,10 считается приемлемым, значит матрица К 5 (удаленность от центра города) согласована.

3.6 Синтез альтернатив

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

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

W_{2i} = (0.39; 0.327; 0.173; 0.067; 0,042);

W_{3K1Y} = (0.444; 0.102; 0.058; 0.034; 0.363);

W_{3K2Y} = (0.407; 0.203; 0.134; 0.102; 0.154);

W_{3K3Y} = (0.402; 0.034; 0.259; 0.153; 0.153);

W_{3K4Y} = (0.47; 0.103; 0.108; 0.103; 0.215);

W_{3K5Y} = (0.432; 0.141; 0.141; 0.111; 0.176).

Приоритеты альтернатив получены следующим образом:

$W_1 = W_{21} \times W_{3K11} + W_{22} \times W_{3K21} + W_{23} \times W_{3K31} + W_{24} \times W_{3K41} + W_{25} \times W_{3K51} =$ 0.426.

$W_2 = W_{21} \times W_{3K12} + W_{22} \times W_{3K22} + W_{23} \times W_{3K32} + W_{24} \times W_{3K42} + W_{25} \times W_{3K52} =$ 0.24.

$W_3 = W_{21} \times W_{3K13} + W_{22} \times W_{3K23} + W_{23} \times W_{3K33} + W_{24} \times W_{3K43} + W_{25} \times W_{3K53} =$ 0.125

$W_4 = W_{21} \times W_{3K14} + W_{22} \times W_{3K24} + W_{23} \times W_{3K34} + W_{24} \times W_{3K44} + W_{25} \times W_{3K54} =$ 0.125.

$W_5 = W_{21} \times W_{3K15} + W_{22} \times W_{3K25} + W_{23} \times W_{3K35} + W_{24} \times W_{3K45} + W_{25} \times W_{3K55} =$ 0.084.

Таким образом, приоритеты альтернатив равны:

альтернатива А1 (название) - W_1 приоритет равен = 0.426;

альтернатива А2 (название)- W_2 приоритет равен = 0.24;

альтернатива А3 (название) - W_3 приоритет равен = 0.125;

альтернатива А4 (название) – W_4 приоритет равен = 0.125;

альтернатива А5 (название) - W_5 приоритет равен =0.084.

Лучшая альтернатива – А1 с приоритетом, равным 0,426.

3.7 Результаты работы программы

Результат работы программы показан ниже (Рисунки 3.2 – 3.4).

```
Матрица попарного сравнения критериев:
      K1    K2    K3    K4    K5      W
K1  1     1     3      6       8  0.39
K2  1     1     2      5       6  0.327
K3  0.33  0.5   1      3       5  0.173
K4  0.17  0.2   0.33   1       2  0.067
K5  0.12  0.17  0.2    0.5     1  0.042
Отношение согласованности: 0.014

Матрица сравнения альтернатив по критерию K1:
      A1    A2    A3    A4    A5      W
A1  1     5     6        9  2     0.444
A2  0.2   1     3        4  0.14  0.102
A3  0.17  0.33  1        3  0.12  0.058
A4  0.11  0.25  0.33     1  0.14  0.034
A5  0.5   7     8        7  1     0.363
Отношение согласованности: 0.084

Матрица сравнения альтернатив по критерию K2:
      A1    A2    A3    A4    A5      W
A1  1      2     4       4   2    0.407
A2  0.5    1     2       2   1    0.203
A3  0.25   0.5   1       1   2    0.134
A4  0.25   0.5   1       1   0.5  0.102
A5  0.5    1     0.5     2   1    0.154
Отношение согласованности: 0.053
```

Рисунок 3.2 – Вывод программы часть 1

```
Матрица сравнения альтернатив по критерию K3:
      A1    A2    A3    A4    A5      W
A1  1        7  2     3     3     0.402
A2  0.14     1  0.14  0.17  0.17  0.034
A3  0.5      7  1     2     2     0.259
A4  0.33     6  0.5   1     1     0.153
A5  0.33     6  0.5   1     1     0.153
Отношение согласованности: 0.027

Матрица сравнения альтернатив по критерию K4:
      A1    A2    A3    A4    A5      W
A1  1        5     4     5    2     0.471
A2  0.2      1     1     1    0.5   0.103
A3  0.25     1     1     1    0.5   0.108
A4  0.2      1     1     1    0.5   0.103
A5  0.5      2     2     2    1     0.215
Отношение согласованности: 0.002

Матрица сравнения альтернатив по критерию K5:
      A1    A2    A3    A4    A5      W
A1  1      3     3     5     2     0.432
A2  0.33   1     2     0.5   1     0.141
A3  0.33   0.5   1     2     1     0.141
A4  0.2    2     0.5   1     0.5   0.111
A5  0.5    1     1     2     1     0.176
Отношение согласованности: 0.085
```

Рисунок 3.3 – Вывод программы часть 2

```
Приоритеты:
1. A1: 0.426
2. A5: 0.240
3. A2: 0.125
4. A3: 0.125
5. A4: 0.084

Лучшая альтернатива: A1 (приоритет: 0.426)
```

Рисунок 3.4 – Вывод программы часть 3

1.8 Вывод по методу МАИ

Подводя итог, можно выделить ряд преимуществ метода анализа иерархий. Метод позволяет учитывать, как количественные, так и качественные критерии.

Также алгоритм включает расчет отношения согласованности, что помогает выявить противоречия в экспертных оценках и повысить достоверность результатов. Однако, метод анализа иерархий не лишен недостатков. Метод учитывает вес критериев, который определяется субъективно, что может привести к неточности и искажению результатов. Метод требует значительных вычислительных ресурсов, особенно при большом количестве альтернатив и критериев.

4 ГРАФИЧЕСКИЙ МЕТОД

В данной работе рассматривается метод линейного программирования, который используется для поиска оптимального решения задачи. В такой задаче целью является линейная функция, а условия ограничены системой линейных равенств и неравенств.

Алгоритм графического метода:

1. Необходимо привести систему уравнений, представленных как условия задачи, в равенства
2. Построить графики функций из уравнений ограничений;
3. Найти на графике область допустимых решений (ОДР);
4. Отобразит на графике целевую функцию;
5. Найти градиент и построить его на графике;
6. Найти с помощью градиента и целевой функции в ОДР точку (max) и точку (min);
7. Найти координаты точек максимума и минимума;
8. Сделать проверку и убедиться, что точки принадлежат ОДР;
9. Найти целевую функцию в точках максимума и минимума F(x)max, F(x)min.

4.1 Постановка задачи

Решить задачу линейного программирования с двумя переменными графическим методом.

4.2 Данные индивидуального варианта

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

$$\begin{cases} -4x_1 + 2x_2 \le 16 \\ 3x_1 + 2x_2 \le 14 \\ x_1, x_2 \ge 0 \end{cases}$$

4.3 Подготовка данных

В среде Microsoft Excel добавим 4 столбца:

1. x_1 – значения от 0 до 10 с шагом 0,5;
2. $x_2 = (\dots)$ – значения ограничения $(-4x_1 + 2x_2 \le 16)$;
3. $x_2 = (\dots)$ – значения ограничения $(3x_1 + 2x_2 \le 14)$;
4. $x_2 = (\dots)$ – значения целевой функции при условии $f(x)$ = 0.

Таблица 4.1 – Данные для графика

x1	x2= $\frac{16+4x1}{2}$	x2 = $\frac{14-3x1}{2}$	x2=x1
0	8	7	0
0,5	9	6,25	0,5
1	10	5,5	1
1,5	11	4,75	1,5
2	12	4	2
2,5	13	3,25	2,5
3	14	2,5	3
3,5	15	1,75	3,5
4	16	1	4
4,5	17	0,25	4,5
5	18	-0,5	5
5,5	19	-1,25	5,5
6	20	-2	6

Продолжение Таблицы 4.1

6,5	21	-2,75	6,5
7	22	-3,5	7
7,5	23	-4,25	7,5
8	24	-5	8
8,5	25	-5,75	8,5
9	26	-6,5	9
9,5	27	-7,25	9,5
10	28	-8	10

4.4 Построение графика

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

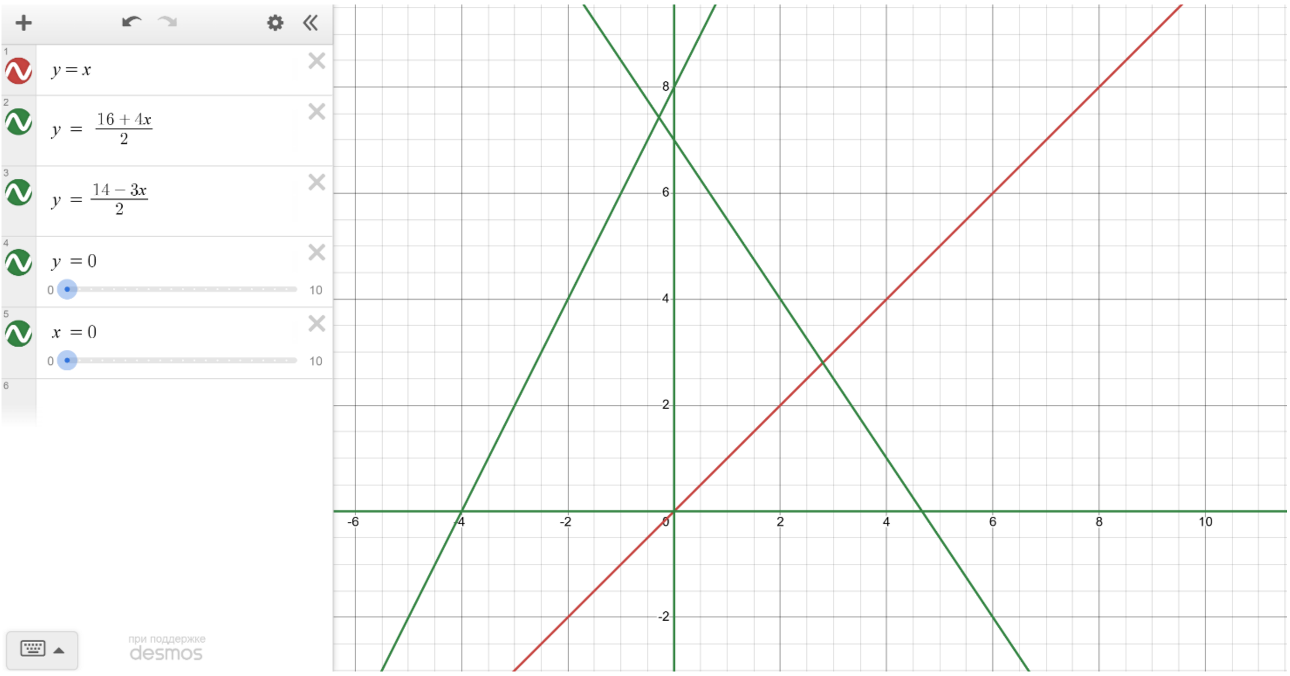


Рисунок 4.1 – Построение графиков по данным

4.5 Выделение области допустимых решений

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

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

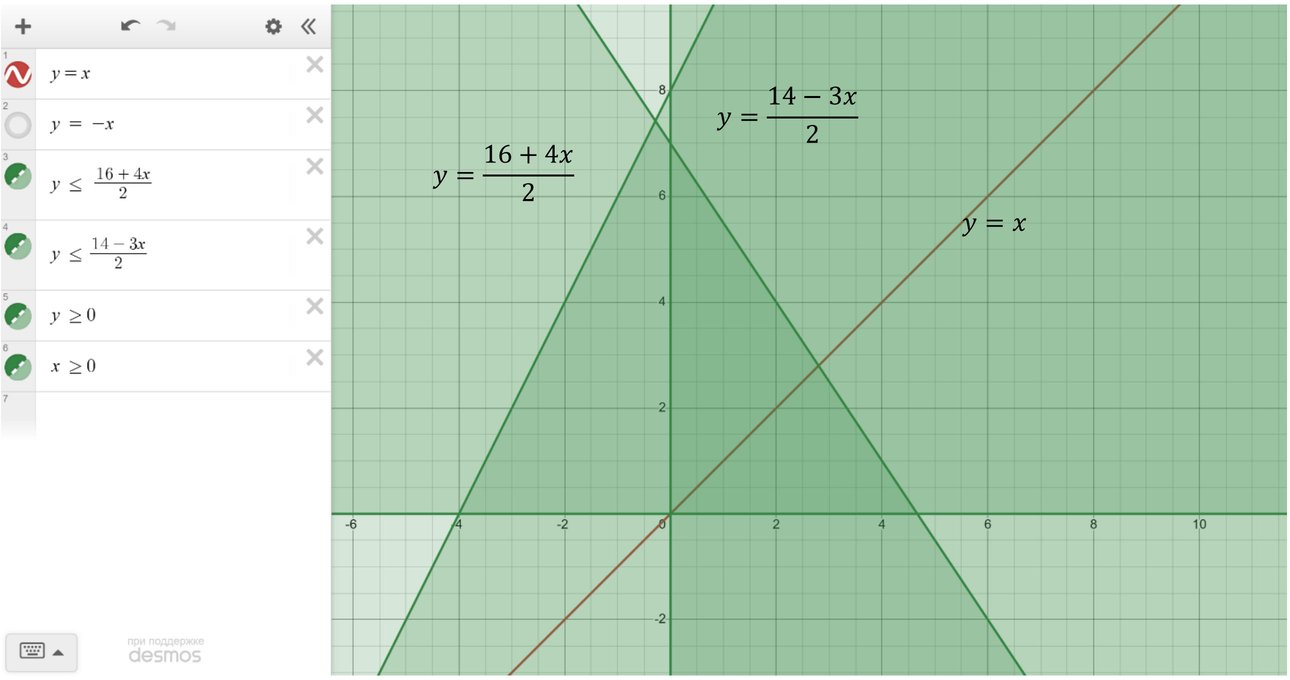


Рисунок 4.2 – Выделение области допустимых решений

4.6 Максимум функции

Для нахождения максимума функции найдем её градиент по формуле 1.1:

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

Для нахождения минимума функции найдем её градиент по формуле 1.1:

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

Градиент функции будет равен {1,-1}, а антиградиент функции будет равен {-1,1}. Изобразим эти вектора на графике.

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

$$\begin{cases} 3x_1 + 2x_2 = 14 \\ x_2 = 0 \end{cases} \begin{cases} 3x_1 = 14 \\ x_2 = 0 \end{cases} \begin{cases} x_1 = \dfrac{14}{3} \\ x_2 = 0 \end{cases}$$

Изобразим эти вектора и точку максимума на графике (Рисунок 4.4).

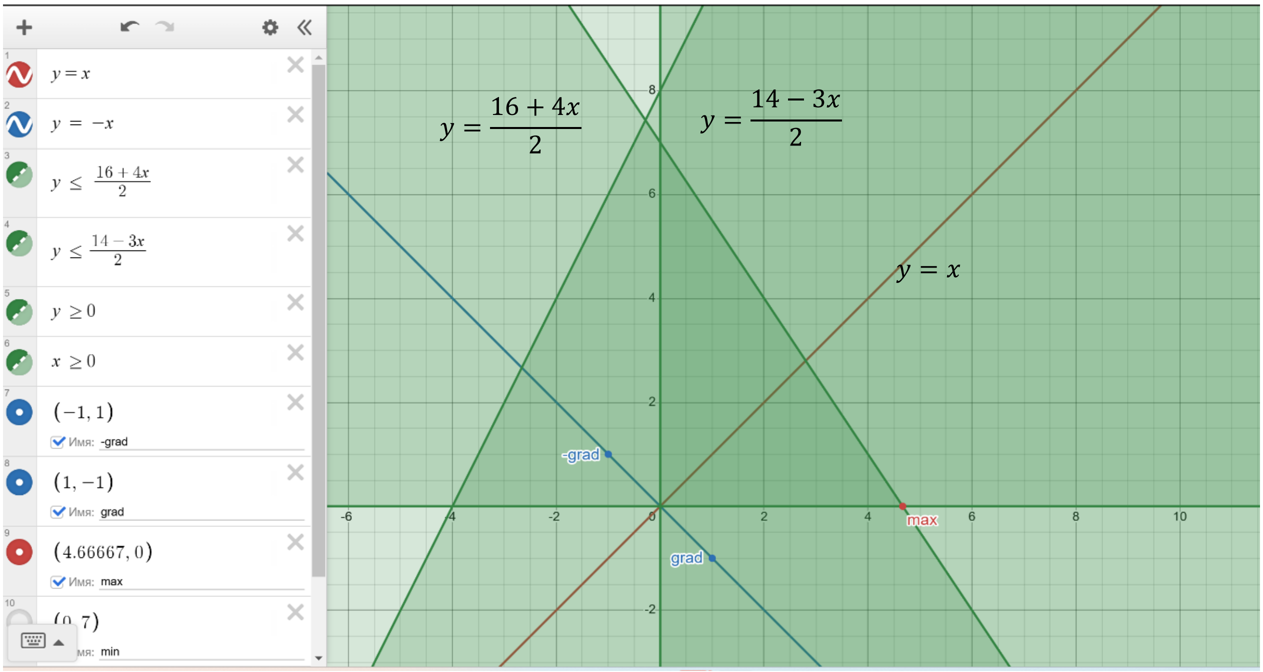


Рисунок 4.4 – Точка максимума функции

Найдем значение функции в точке максимума.

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

$$\begin{cases} -4x_1 + 2x_2 \le 16 \\ 3x_1 + 2x_2 \le 14 \\ x_1, x_2 \ge 0 \end{cases} \begin{cases} -4 \cdot \frac{14}{3} \le 16 \\ 3 \cdot \frac{14}{3} \le 14 \\ \frac{14}{3}, 0 \ge 0 \end{cases} \begin{cases} -\frac{56}{3} \le 16 \\ 14 \le 14 \\ \frac{14}{3}, 0 \ge 0 \end{cases}$$

Получим значение равное F(x)max $= \frac{14}{3}$.

4.7 Минимум функции

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

max(2,8;4,8)

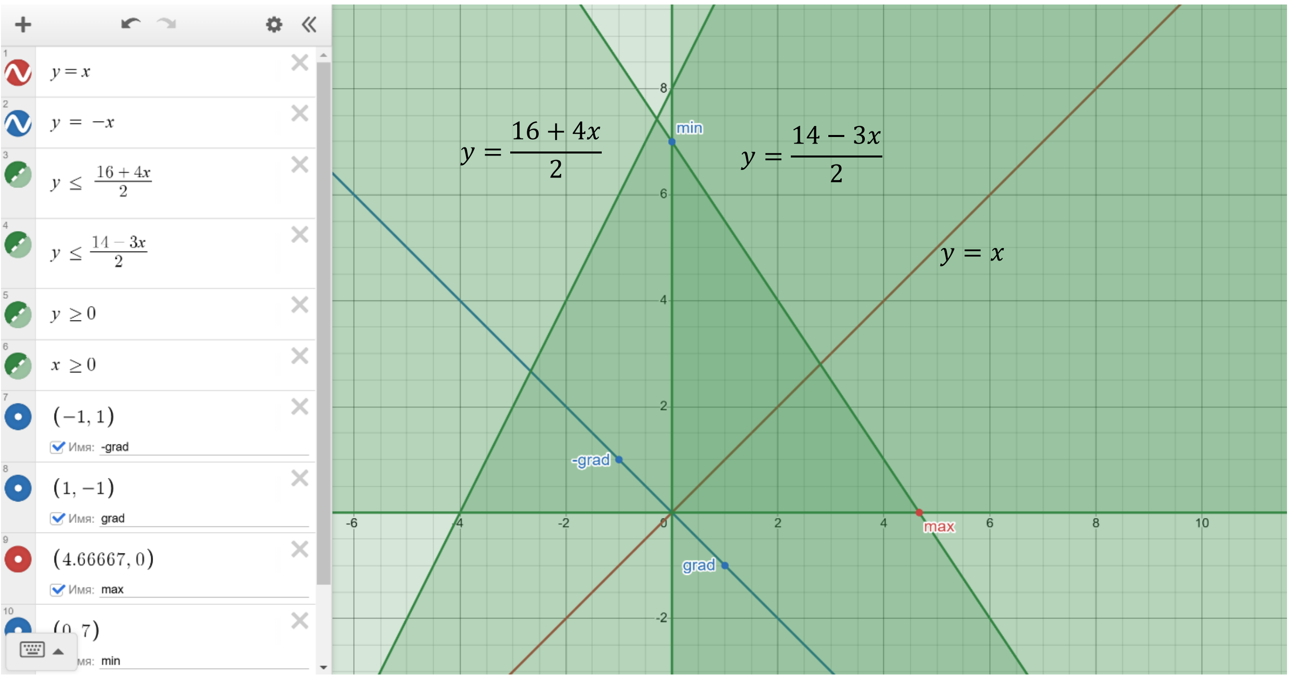


Рисунок 4.5 – Точка минимума функции

Найдем координаты точки минимума:

$$\begin{cases} x_1 = 0 \\ 3x_1 + 2x_2 = 14 \end{cases} \begin{cases} x_1 = 0 \\ 2x_2 = 14 \end{cases} \begin{cases} x_1 = 0 \\ x_2 = 7 \end{cases}$$

В результате получим точку с координатами $(\frac{8}{3}, 0)$. Найдем значение функции в этой точке.

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

$$\begin{cases} -4x_1 + 2x_2 \le 16 \\ 3x_1 + 2x_2 \le 14 \\ x_1, x_2 \ge 0 \end{cases} \begin{cases} 2 \cdot 7 \le 16 \\ 2 \cdot 7 \le 14 \\ 0, 7 \ge 0 \end{cases} \begin{cases} 14 \le 16 \\ 14 \le 14 \\ 0, 7 \ge 0 \end{cases}$$

Получим результат F(x)min = -7.

Ответ:

F(x)max = $\frac{14}{3}$.

F(x)min = -7.

4.8 Вывод по графическому методу

В ходе данной работы был изучен графический метод решения задачи линейного программирования. Из его преимуществ можно выделить простоту реализации и большую наглядность. Однако, при количестве переменных больше двух метод становится трудным для графической интерпретации.

5 СИМПЛЕКСНЫЙ МЕТОД

Далее рассматривается симплексный метод для решения задачи линейного программирования. В симплексном методе для нахождения максимума (минимума) рассматривается многогранник условий. Если рассматриваемая вершина не соответствует максимуму (минимуму), то переходят к соседней, увеличивая значение функции цели при решении задачи на максимум (уменьшая для задачи на минимум). Так как число вершин многогранника ограничено, то за конечное число шагов точно можно найти оптимальное значение или определить, что задача неразрешима. Таким образом, симплексный метод сводится к целенаправленному перебору опорных решений ЗЛП.

5.1 Постановка задачи

Задание 11. Решить прямую ЗЛП с помощью симплексного метода и обратную с помощью теорем двойственности. Определить интервалы устойчивости.

Задача. Для изготовления четырех видов продукции A, B, C и D используются три вида ресурсов I, II, III. Дальнейшее условие задачи в таблице П.8.

Таблица 5.1. Затраты ресурсов

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

Найти программу максимального выпуска продукции из имеющихся ресурсов.

5.2 Математическая модель задачи

Пусть x1 – тип продукции A, x2 –тип продукции B, x3 –тип продукции C, x4 – тип продукции D. Прибыль от продажи всей продукции составит 7,5x1 + 3x2 + 6x3 + 12x4, прибыль требуется максимизировать.

Ограничения задачи:

$$\begin{cases} 2x1 \ + \ x2 \ + \ 0{,}5x3 + 4x4 \ \le \ 3400 \\ x1 \ + \ 5x2 \ + \ 3x3 \ \le 1200 \\ 3x1 \ \ + \ 6x3 + x4 \ \le \ 3000 \end{cases}$$

Таким образом, переходим к задаче линейного программирования:

$$f(x) = 7{,}5x1 \ + \ 3x2 \ + \ 6x3 \ + \ 12x4 \to max$$

$$\begin{cases} 2x1 \ + \ x2 \ + \ 0{,}5x3 + 4x4 \ \le \ 3400 \\ x1 \ + \ 5x2 \ + \ 3x3 \ \le 1200 \\ 3x1 \ \ + \ 6x3 + x4 \ \le \ 3000 \\ xi \ \ge \ 0, \qquad 1 \ \le \ i \ \le 4 \end{cases}$$

5.3 Ручной расчет метода

Приведем задачу к канонической форме. Для этого в левые части ограничений вводим дополнительные переменные x5 ≥ 0, x6 ≥ 0, x7 ≥ 0. Эти переменные выбираются так, чтобы они обращали неравенства в равенства.

$$\begin{cases} 2x1 \ + \ x2 \ + \ 0{,}5x3 + 4x4 + x5 \ \le \ 3400 \\ x1 \ + \ 5x2 \ + \ 3x3 + x6 \le 1200 \\ 3x1 \ \ + \ 6x3 + x4 + x7 \le \ 3000 \\ xi \ \ge \ 0, \qquad 1 \ \le \ i \ \le 7 \end{cases}$$

$$f(x) = 7{,}5x1 \ + \ 3x2 \ + \ 6x3 \ + \ 12x4 + 0x5 + 0x6 + 0x7$$

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

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

$$A1 = \begin{pmatrix} 2 \\ 1 \\ 3 \end{pmatrix}, A2 = \begin{pmatrix} 1 \\ 5 \\ 0 \end{pmatrix}, A3 = \begin{pmatrix} 0{,}5 \\ 3 \\ 6 \end{pmatrix}, A4 = \begin{pmatrix} 4 \\ 0 \\ 1 \end{pmatrix}, A5 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, A6 = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, A7 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix},$$

$$A0 = \begin{pmatrix} 3400 \\ 1200 \\ 3000 \end{pmatrix}$$

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, x3, x4.* Разложение позволяет найти первое базисное допустимое решение.

Для этого свободные переменные *x1, x2, x3, x4* приравниваем нулю. В результате получим разложение

$$\overline{A5x5} + \overline{A6x6} + \overline{A7x7} = \overline{A0},$$

Которому соответствует первоначальный опорный план

$$x^{(0)} = (x1, x2, x3, x4, x5, x6) = (0, 0, 0, 0, 3400, 1200, 3000),$$

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

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

$$\overline{C_B} = (c5, c6, c7)^T = (0{,}0{,}0)^T.$$

В левый столбец Таблицы 1.2 запишем переменные x5, $x6$, $x7$ образующие базис, в верхней строке – небазисные переменные $x1$, $x2$, $x3$, $x4$. В строке cj запишем коэффициенты целевой функции, соответствующие небазисным переменным c1 = 15/2, c2 = 3, c3 = 6, c4 = 12. В столбце CB запишем коэффициенты целевой функции, соответствующие базисным переменным Столбец, определяемый переменной $x1$, состоит из коэффициентов вектора $A1$. Аналогично, столбцы, определяемые переменными $x2$, x3, x4, состоят из коэффициентов векторов $A2$, $A3$, $A4$, соответственно. Крайний правый столбец заполняется элементами столбца $A0$, в нем же в результате вычислений получаем оптимальный план.

Таблица 5.2 – Начальная симплекс-таблица задачи о максимальном доходе

	c_j	15/2	3	6	12	
$\overline{C_B}$		X1	X2	X3	X4	$\overline{A_0}$
0	X5	2	1	1/2	4	3400
0	X6	1	5	3	0	1200
0	X7	3	0	6	1	3000
	f					
		Δ_1	Δ_2	Δ_3	Δ_4	Q

Заполнение f-строки (Таблица 1.3). Найдем относительные оценки Δ1, Δ2, Δ3, Δ4 и значение целевой функции Q.

$$\Delta_1 = (\overline{C_B} * \overline{A_1}) - c_1 = 0 * 2 + 0 * 1 + 0 * 3 - 15/2 = -15/2;$$

$$\Delta_2 = (\overline{C_B} * \overline{A_2}) - c_2 = 0 * 1 + 0 * 5 + 0 * 0 \ - 3 \ = -3;$$

$$\Delta_3 = (\overline{C_B} * \overline{A_3}) - c_3 = -6;$$

$$\Delta_4 = (\overline{C_B} * \overline{A_4}) - c_4 = -12;$$

$$\mathrm{Q} = (\overline{C_B} * \overline{A_0}) = 0 * 3400 + 0 * 1200 + 0 * 3000 = 0.$$

Таблица 5.3 – Заполнение f-строки

	c_j	15/2	3	6	12		
$\overline{C_B}$		X1	X2	X3	X4	$\overline{A_0}$	
0	X5	2	1	1/2	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	-15/2	-3	-6	-12	0	
		Δ_1	Δ_2	Δ_3	Δ_4	Q	

Для оптимальности опорного решения в задаче на максимум требуется выполнение неотрицательности всех относительных оценок Δi ≥ 0. Так как оценки Δ1= −15/2, Δ2= -3, Δ3 = -6, Δ4 = -12 в f-строке отрицательны, то это свидетельствуют о возможности улучшения полученного решения. Наибольшая по модулю отрицателья оценка Δ4= −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 строки с переменными $x4$. Исключим из базиса переменную $x4$. В Таблице 5.3 разрешающий столбец и разрешающая строка выделены. Разрешающим элементом является число $a54 = 4$.

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

Таблица 5.4 – Новая симплекс-таблица

	c_j	15/2	3	6	12	
$\overline{C_B}$		X1	X2	X3	X4	$\overline{A_0}$
0	X5				1/4	
0	X6					
0	X7					
	f					
		Δ_1	Δ_2	Δ_3	Δ_4	Q

В Таблице 5.4 переменные $x5$ и $x4$ меняются местами вместе с коэффициентами cj. Разрешающий элемент заменяется на обратный. В Таблице 1.5 элементы разрешающей строки делятся на разрешающий элемент. Элементы разрешающего столбца делятся на разрешающий элемент и меняют знак.

Таблица 5.5 – Симплекс преобразования

	c_j	15/2	3	6	0	
$\overline{C_B}$		X1	X2	X3	X5	$\overline{A_0}$
12	X4	1/2	1/4	1/8	1/4	850
0	X6				0	
0	X7				-1/4	
	f				-12	
		Δ_1	Δ_2	Δ_3	Δ_4	Q

Остальные элементы рассчитываются по «правилу прямоугольника». Сформируем полную таблицу со всеми элементами (Таблица 5.6).

$$\mathrm{a}_{21} = \frac{(1 * 4) - (2 * 0)}{4} = 1;\ \mathrm{a}_{22} = \frac{(5 * 4) - (1 * 0)}{4} = 5;$$

$$a_{23} = \frac{(3*4)-(1/2*0)}{4} = 3;\ a_{25} = \frac{(1200*4)-(3400*0)}{4} = 1200;$$

$$a_{33} = \frac{(6*4)-(1/2*1)}{4} = 47/8;\ a_{35} = \frac{(3000*4)-(3400*1)}{4} = 2150;$$

$$\Delta_1 = \frac{(-15/2*4)-(-2\ *\ -12)}{4} = -3/2;\ \Delta_2 = \frac{(-3*4)-(1\ *\ -12)}{4} = 0;$$

$$\Delta_3 = \frac{(-6*4)-\left(-\frac{1}{2}*-12\right)}{4} = -9/2;\ Q = \frac{(0*4)-(-12*3400)}{4} = 10200;$$

Базисное решение, которое дает последняя таблица

$$x^{(1)} = (x1, x2, x3, x4, x5, x6, x7) = (0,0,0,850,0,1200,2150),$$

$$f\left(x^{(1)}\right) = (\overline{C_B} * \overline{A_0}) = 12*850+\ +0\ *\ 1200\ +\ 0\ *\ 2150\ =\ 10200.$$

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

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

Таблица 5.7 – Итерация 2

	c_j	0	0	0	0	
$\overline{C_B}$		X1	X5	X6	X7	$\overline{A_0}$
12	X4	0.46	0.25	-0.05	0	799.17
3	X2	-0.05	0.02	0.2	-0.1	19.92
6	X3	0.42	-0.04	0.01	0.17	366.81
	f	0.4	2.81	0.04	0.75	11850.62
		Δ_1	Δ_2	Δ_3	Δ_4	Q

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

Подставляем базисное решение, которое дает последняя таблица

$$x^{(4)} = (x1, x2, x3, x4, x5, x6, x7) = (0, 19.92, 366.81, 799.17, 0, 0, 0),$$

Где n – количество итераций

$$f\left(x^{(3)}\right) = (\overline{C_B} * \overline{A_0})$$
$$= 7{,}5 * 0 + 3 * 19.92 + 6 * 366.81 + 12 * 799.17 + 0 * 0 + \ 0 * 0 + 0 * 0 = 11850.62.$$

Таким образом, фабрика должна выпускать в течении недели x2 = 19.92 шт. типа продукции B, x3 = 366.81 шт. типа продукции C, x4 = 799.17 шт. типа продукции D. Тогда фабрика получит максимальный доход от продажи 11850.62 [тыс. ден.ед].

5.4 Консольный результат работы

Результат работы программы показан ниже (Рисунки 5.1, 5.2).

```
Начальная симплекс-таблица
      Св    Базис        X1        X2        A0
      cj                260       300
     0.0       x3     16.00     12.00   1200.00
     0.0       x4      0.20      0.40     30.00
     0.0       x5      6.00      5.00    600.00
     0.0       x6      3.00      4.00    300.00
       F            -260.00   -300.00      0.00
   Delta            -260.00   -300.00

Итерация 0
      Св    Базис        X1        X4        A0
     0.0       x3     10.00    -30.00    300.00
   300.0       x2      0.50      2.50     75.00
     0.0       x5      3.50    -12.50    225.00
     0.0       x6      1.00    -10.00      0.00
       F            -110.00    750.00  22500.00
   Delta            -110.00    750.00
0.4
```

Рисунок 5.1 - Вывод программы часть 1

```
Итерация 1
        Св    Базис       X4        X6        A0
       0.0       x3    70.00    -10.00    300.00
     300.0       x2     7.50     -0.50     75.00
       0.0       x5    22.50     -3.50    225.00
     260.0       x1   -10.00      1.00      0.00
         F           -350.00    110.00  22500.00
     Delta           -350.00    110.00
1.0

Итерация 2
        Св    Базис       X3        X6        A0
       0.0       x4     0.01     -0.14      4.29
     300.0       x2    -0.11      0.57     42.86
       0.0       x5    -0.32     -0.29    128.57
     260.0       x1     0.14     -0.43     42.86
         F              5.00     60.00  24000.00
     Delta              5.00     60.00
70.0

Оптимальное решение:
x1 = 42.86
x2 = 42.86
Максимальный выпуск продукции: 24000.00
```

Рисунок 5.2 - Вывод программы часть 2

5.5 Вывод по симплексному методу

В ходе данной работы было найдено оптимальное решение прямой задачи линейного программирования. Для этого ЗЛП приводится к канонической форме, формируется начальная симплекс-таблиц. Далее на ее основе составлялись новые таблицы с помощью повторяющихся расчетов итераций до тех пор, пока не было получено оптимальное решение.

Из преимуществ симплексного метода можно выделить гарантированное решение за конечное число шагов, что позволяет эффективно решать различные виды задач даже больших объемов. Однако, для метода задачи должны быть представлены в каноническом виде, а сам алгоритм требуют более углубленных знаний математики.

6 ДВОЙСТВЕННАЯ ЗАДАЧА

Рассмотрим двойственный метод для решения задачи линейного программирования. Для каждой ЗЛП можно составить другую линейную задачу, называемую двойственной. Каждая из задач двойственной пары фактически является самостоятельной задачей и может быть решена не зависимо от другой. Связь задач заключается в том, что решение одной из них может быть получено непосредственно из решения другой. Взаимная симметрия прямой и двойственной задач определяет существование определенного соответствия между их оптимальными решениями. Эти соответствия устанавливают теоремы двойственности, которые рассматриваются в этой работе.

6.1 Постановка задачи

Задание 11. Решить прямую ЗЛП с помощью симплексного метода и обратную с помощью теорем двойственности. Определить интервалы устойчивости.

Задача. Ткань трех артикулов производится на ткацких станках двух типов с различной производительностью.

Таблица 6.1. Затраты ресурсов

Виды ресурсов	Объем ресурсов	Производительность и норма труда		
		1	2	3
Станки I вида	30	20	10	25
Станки II вида	45	8	20	10
Пряжа	30	120	180	210
Красители	1	10	5	8
Цена		15	15	20

Для изготовления ткани используется пряжа и красители. В таблице П.15. указаны мощности станков (тыс. станко-час), ресурсы пряжи и красителей (тыс. кг), производительность станков по каждому виду ткани (м/ч), нормы расхода пряжи и краски (кг на 1000 м) и цена (у. е.) 1 м ткани. Определить оптимальный ассортимент, максимизирующий прибыль, если себестоимость 1 м ткани составляет соответственно 3, 5 и 15 у.е.

6.2 Математическая модель исходной задачи

Пусть:

- x1 – объем производства ткани артикула 1, тыс. м.
- x2 – объем производства ткани артикула 2, тыс. м.
- x3 – объем производства ткани артикула 3, тыс. м. Рассчитаем прибыль на 1 тыс. м каждого артикула:
- Прибыль от ткани 1: (Цена 15 - Себестоимость 3) * 1000 = 12 * 1000 = 12000 у.е. -> коэффициент 12 при x1 (в тыс. у.е.).
- Прибыль от ткани 2: (Цена 15 - Себестоимость 5) * 1000 = 10 * 1000 = 10000 у.е. -> коэффициент 10 при x2 (в тыс. у.е.).
- Прибыль от ткани 3: (Цена 20 - Себестоимость 15 * 1000 = 5 * 1000 = 5000 у.е. -> коэффициент 5 при x3 (в тыс. у. е.).

$$f(x) = 12x1 + 10x2 + 5x3 \rightarrow max$$
$$\begin{cases} 20x1 + 10x2 + 25x3 \leq 30 \\ 8x1 + 20\,x2 + 10\,x3 \leq 45 \\ 120x1 + 180x2 + 210x3 \leq 30 \\ 10x1 + 5x2 + 8x3 \leq 1 \\ xi \geq 0, i = 1, 2, 3 \end{cases}$$

Векторный вид:

$$\text{A}1 = \begin{pmatrix} 20 \\ 8 \\ 120 \\ 10 \end{pmatrix}, \text{A}2 = \begin{pmatrix} 10 \\ 20 \\ 180 \\ 5 \end{pmatrix}, \text{A}3 = \begin{pmatrix} 25 \\ 10 \\ 210 \\ 8 \end{pmatrix}, \text{A}4 = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \text{A}5 = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}, \text{A}6 = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix},$$
$$\text{A}7 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

В ходе решения прямой задачи было определено, что максимальный доход от продажи составляет $f_{max} = 1800$ у.е, оптимальный план $\overline{\text{x}^*} = (x1, x2, x3, s1, s2, s3, s4) = (0.025, 0.15, 0, 28, 41.8, 0, 0)$.

6.3 Соответствующая исходной двойственная задача

Найдем соответствующую двойственную задачу. Введем вектор двойственных переменных размерности три y = $(y1,\ y2,\ y3,\ y4)^{\text{T}}$. Соответствующие векторы и матрица ограничений имеет вид

$$\overline{c} = (12, 10, 5), \overline{b} = (30, 45, 30, 1), A = \begin{pmatrix} 20 & 10 & 25 \\ 8 & 20 & 10 \\ 120 & 180 & 210 \\ 10 & 5 & 8 \end{pmatrix}, A^T = \begin{pmatrix} 20 & 8 & 120 & 10 \\ 10 & 20 & 180 & 5 \\ 25 & 10 & 210 & 8 \end{pmatrix}.$$

Запишем двойственную задачу. Найти минимум функции.

$$g(\overline{y}) = (\overline{b}, \overline{y}) = 30y_1 + 45y_2 + 30y_3 + 1y_4 \rightarrow min$$

При ограничениях:

$$\begin{pmatrix} 20 & 10 & 25 \\ 8 & 20 & 10 \\ 120 & 180 & 210 \\ 10 & 5 & 8 \end{pmatrix} * \begin{pmatrix} y1 \\ y2 \\ y3 \\ y4 \end{pmatrix} \geq \begin{pmatrix} 12 \\ 10 \\ 5 \end{pmatrix}$$

$$\begin{cases} 20y1 + 8y2 + 120y3 + 10y4 \geq 12, \\ 10y1 + 20\,y2 + 180y3 + 5y4 \geq 10, \\ 25y1 + 10y2 + 210y3 + 8y4 \geq 5, \\ yi \geq 0, \qquad 1 \leq i \leq 4. \end{cases}$$

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 максимальная прибыль составляет f_max = 1.8 тыс. ден.ед., оптимальный план x* = (x1,x2,x3,x4,x5,x6,x7) = (0.025, 0.15, 0, 28, 41.8, 0, 0).

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 двойственная задача также разрешима. Ее решение может быть найдено из формулы:

$$\overline{x^*} = \overline{C_B} \cdot D^{-1},$$

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 (s1), x5 (s2), x2, x1. Соответствующие этим переменным векторы A4, A5, A2, A1 используются для формирования столбцов матрицы D.

$$A4 = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}, A5 = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}, A2 = \begin{pmatrix} 10 \\ 20 \\ 180 \\ 5 \end{pmatrix}, A1 = \begin{pmatrix} 20 \\ 8 \\ 120 \\ 10 \end{pmatrix}$$

Тогда,

$$D = (\overline{A_4}, \overline{A_5}, \overline{A_2}, \overline{A_1}) = \begin{pmatrix} 1 & 0 & 10 & 20 \\ 0 & 1 & 20 & 8 \\ 0 & 0 & 180 & 120 \\ 0 & 0 & 5 & 10 \end{pmatrix}$$

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

$$\left(\begin{array}{cccc|cccc} 1 & 0 & 10 & 20 & 1 & 0 & 0 & 0 \\ 0 & 1 & 20 & 8 & 0 & 1 & 0 & 0 \\ 0 & 0 & 180 & 120 & 0 & 0 & 1 & 0 \\ 0 & 0 & 5 & 10 & 0 & 0 & 0 & 1 \end{array}\right)$$

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

Запишем обратную матрицу.

$$D^{-1} = (y_1^*, y_2^*, y_4^*, y_5^*) = \begin{pmatrix} 1 & 0 & 0 & -2 \\ 0 & 1 & -2/15 & 0.8 \\ 0 & 0 & 1/120 & -0.1 \\ 0 & 0 & -1/140 & 0.15 \end{pmatrix}$$

Базисными переменными в симплекс-таблице являются $\bar{\bar{C}}_{\bar{B}} = (0, 0, 10, 12)$, тогда

$$\overline{y^*} = (y_1^*, y_2^*, y_4^*, y_5^*) = \overline{C_B} \cdot D^{-1} =$$

$$= (0*1+0*0+10*0+12*0; 0*0+0*1+10*0+12*0; 0*0$$
$$+0*(-2/15)+10*(1/120)+12*(-1/240)); 0*(-2)+0$$
$$*(0.8)+10*(-0.1)+12*(0.15) =$$
$$= (0, 0, 1/30, 0.8).$$
$$= (5;\ 0;\ 0;\ 60).$$

При этом минимальное значение целевой функции двойственной задачи

$$g_{min} = g\overline{(y^*)} = \left(\bar{b}, \overline{y^*}\right) = 30*0\ +\ 45*0\ +\ 30*(1/30)\ +\ 1*0.8$$
$$=\ 0\ +\ 0\ +\ 1\ +\ 0.8\ =\ 1.8 \text{ тыс. ден. ед.}$$

совпадает с максимальным значением f_{max} = 1.8 [тыс. ден.ед.] прямой задачи, что является результатом взаимодвойственности. Таким образом:

$$maxf(\bar{x}) = ming(\bar{y}) = 1.8 \text{ [тыс. ден. ед.]}$$

6.5 Вторая теорема двойственности

Для того, чтобы планы $\overline{x^*} = (x_1^*, x_2^*, \ldots, x_n^*)$ и $\overline{y^*} = (y_1^*, y_2^*, \ldots, y_m^*)$ ЗЛП двойственной пары были оптимальными, необходимо и достаточно, чтобы эти планы удовлетворяли условиям дополняющей нежесткости.

$$\left\{x_j^*\left(\sum_{i=1}^{m} a_{ij} y_i^* - c_j\right) = 0, j = \overline{1, n}.\right.$$

$$\left\{y_i^*\left(\sum_{i=1}^{n} a_{ij} x_j^* - b_i\right) = 0, i = \overline{1, m}.\right.$$

Итак, имеем оптимальное решение прямой задачи: объем производства ткани артикула 1 – x1* = 0.025 тыс. м; объем производства ткани артикула 2 – x2* = 0.15 тыс. м; объем производства ткани артикула 3 – x3* = 0 тыс. м; максимальная прибыль от продажи f_max = 1.8 [тыс. ден.ед. / неделю]. Рассмотрим выполнение неравенств прямой задачи при подстановке x1*, x2*, x3* в систему ограничений (Таблица 6.2).

$$\begin{cases} 20y1 + 8y2 + 120y3 + 10y4 = 12, \\ 10y1 + 20\ y2 + 180y3 + 5y4 \geq 10 \\ \text{y1} =\ 0 \\ \text{y2} =\ 0 \end{cases}.$$

Решим данную систему уравнений:

Подставляем y1 = 0 и y2 = 0 в третье и четвертое уравнения:

1. 20(0) + 8(0) + 120y3 + 10y4 = 12 => 120y3 + 10y4 = 12
2. 10(0) + 20(0) + 180y3 + 5y4 = 10 => 180y3 + 5y4 = 10

Из уравнения (1) 10y4 = 12 - 120y3 => y4 = (12 - 120y3) / 10 = 1.2 - 12y3.

Подставим это в уравнение (2):

180y3 + 5(1.2 - 12y3) = 10

180y3 + 6 - 60y3 = 10

120y3 = 10 - 6

120y3 = 4

y3 = 4 / 120 = 1/30

Теперь найдем y4:

y4 = 1.2 - 12(1/30) = 1.2 - 12/30 = 1.2 - 2/5 = 1.2 - 0.4 = 0.8

Итак, решение системы: y1 = 0, y2 = 0, y3 = 1/30, y4 = 0.8. Следовательно, y* = (0, 0, 1/30, 0.8).

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

$$g(\bar{\bar{y}}^*) = (\bar{b}, \bar{\bar{y}}^*) = 30y1* + 45y2* + 30y3* + 1y4*$$
$$= 30(0) + 45(0) + 30(1/30) + 1(0.8) = 0 + 0 + 1 + 0.8$$
$$= 1.8 \text{ тыс. ден. ед.}$$

$$min\ g(\bar{y}) = 1.8[\text{тыс. ден. ед.}]$$

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

Таблица 6.2 – Выполнение неравенств прямой задачи

Ограничение	Расчет	Вывод
20x1 + 10x2 + 25x3 <= 30	20(0.025) + 10(0.15) + 25(0) = 0.5 + 1.5 = 2 <30	Первое ограничение прямой задачи выполняется как строгое неравенство, остается избыток ресурса Станков I вида. Значит, этот ресурс не является дефицитным и его оценка в оптимальном плане равна нулю (y1 = 0).
8x1 + 20x2 + 10x3 <= 45	8(0.025) + 20(0.15) + 10(0) = 0.2 + 3.0 = 3.2<45	Второе ограничение прямой задачи выполняется как строгое неравенство, остается избыток ресурса Станков II вида. Значит, этот ресурс не является дефицитным и его оценка в оптимальном плане равна нулю (y2 = 0).
120x1 + 180x2 + 210x3 <= 30	120(0.025) + 180(0.15) + 210(0) = 3 + 27 = 30 = 30	Третье ограничение прямой задачи выполняется как равенство. Это означает, что ресурс Пряжа полностью используется в оптимальном плане, является дефицитным и его оценка согласно второй теоремы двойственности отлична от нуля (y3 ≠ 0).
10x1 + 5x2 + 8x3 <= 1	10(0.025) + 5(0.15) + 8(0) = 0.25 + 0.75 = 1 = 1	Четвертое ограничение прямой задачи выполняется как равенство. Это означает что ресурс Красители полностью используется в оптимальном плане, является дефицитным и его оценка согласно второй теоремы двойственности отлична от нуля (y4 ≠ 0).
x1 ≥ 0	0.025> 0	Первое ограничение в двойственной задаче будет равенством 20y1 + 8y2 + 120y3 + 10y4 = 12, т.е. производство ткани артикула 1 экономически оправдано до предела.
x2 ≥ 0	0.15> 0	Второеограничение в двойственной задаче будет равенством 10y1 + 20y2 + 180y3 + 5y4 = 10, т.е. производство ткани артикула 2 экономически оправдано до предела.

x3 ≥ 0	0 = 0	Ткань артикула 3 не производится (x3*=0). Соответствующее третье ограничение двойственной задачи 25y1 + 10y2 + 210y3 + 8y4> = 5 выполняется (здесь 13.4> = 5), что означает, что данный артикул не является предельно рентабельным. Известные условия y1=0 и y2=0 (из-за избытка ресурсов Станков I и II) влияют на оценку этой рентабельности.

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 Z_{max}.

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

$$D^{-1} = (y_1^*, y_2^*, y_4^*, y_5^*) = \begin{pmatrix} 1 & 0 & 0 & -2 \\ 0 & 1 & -2/15 & 0.8 \\ 0 & 0 & 1/120 & -0.1 \\ 0 & 0 & -1/140 & 0.15 \end{pmatrix}.$$

Индексы базисных переменных оптимального плана:

$$\overline{A_0^*} = (x_1^*, x_2^*, x_4^*, x_5^*) = \begin{pmatrix} 28 \\ 41.8 \\ 0.15 \\ 0.025 \end{pmatrix}$$

Свободные члены неравенств (ограничений) прямой задачи:

$$\overline{\mathrm{A}_0} = (b_1, b_2, b_3, b_4) = \begin{pmatrix} 30 \\ 45 \\ 30 \\ 1 \end{pmatrix}$$

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

Ресурс 1 (Станки I вида). Найдем нижнюю границу. В первом столбце обратной матрицы один положительный элемент (1), ему соответствует индекс базисной переменной оптимального плана (28).

$$\Delta b_1^{H} = min\{28/1\} = 28$$

Найдем верхнюю границу. В обратной матрице стоит < 0 и таких элементов нет, следовательно.

$$\Delta b_1^{B} = +\infty$$

Таким образом, получаем $\Delta b_1 \in [-28, +\infty)$.

Тогда первый ресурс может изменяться в интервале:

$$(b_1 - \Delta b_1^{H}, b_1 + \Delta b_1^{B}) = [30 - 28, 30 + \infty) = [2, +\infty)$$

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

Ресурс 2 (Станки II вида). Рассматриваем второй столбец обратной матрицы, (0, 1, 0, 0).

Найдем нижнюю границу.

$$\Delta b_2^{H} = min\{41.8 \ / \ 1\} = 41.8$$

Найдем верхнюю границу.

$$\Delta b_2^{B} = +\infty$$

Получаем $\Delta b_2 \in [-41.8, +\infty)$.

Тогда второй ресурс может изменяться в интервале:

$$(b_2 - \Delta b_2^H, b_2 + \Delta b_2^B) = (45 - 41.8, +\infty\ ) = (3.2; +\infty)$$

Ресурс 3 (Пряжа). Рассматриваем третий столбец обратной матрицы, в котором один положительный элемент (1/120). Данному элементу соответствует индекс соответствующего базисного переменного оптимального плана – 0.15.

Находим нижнюю границу.

$$\Delta b_3^H = min\{0.15\ /\ (1/120)\} = 18$$

Верхняя граница: $\Delta b_3^В = 6$, так как 0.025 / (1/240) = 0.025 * 240 = 6. Тогда, получаем что $\Delta b_3 \in [-18, 6]$.

Получаем интервал устойчивости оценок по отношению к третьему ограничению:

$$(b_3 - \Delta b_3^H, b_3 + \Delta b_3^B) = (30 - 18;\ 30 + 6\ ) = (12;\ 36)$$

Ресурс 4 (Красители). Рассматриваем четвертый столбец обратной матрицы, в котором один положительный элемент (0.8). Данному элементу соответствует индекс соответствующего базисного переменного оптимал

Находим нижнюю границу.

$$\Delta b_3^H = min\{52.25, 1/6\} = 1/6$$

Верхняя граница: $\Delta b_3^В = 1.5$,

Получаем интервал устойчивости оценок по отношению к третьему ограничению:

$$(b_4 - \Delta b_4^H, b_4 + \Delta b_4^B) = \left(1 - \frac{1}{6}; 1 + 1.5\ \right) = (0.833;\ 2.5)$$

Далее оценим влияние изменения объема ресурсов на величину максимальной прибыли. Как известно, это дефицитные ресурсы $y_3^* = 0.033$ и y*4 = 0.8/ Введем верхние границы $\Delta b_3^В$ и $\Delta b_4^В$ в формулу:

$$\Delta G^{i}_{max} \approx y^{*}_{i} \times \Delta b_{i}$$

$$\Delta G_{max_1} = y_3 \times \Delta b^{B}_{1} = 0.033 \times 6 = \frac{6}{30} = 0.2 \text{ тыс. ед.}$$

$$\Delta G_{max_2} = y_4 \times \Delta b^{B}_{4} = 0.8 \times 15 = 1.2 \text{ тыс. ед.}$$

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

$$\Delta Gmax = \Delta Gmax3 + \Delta Gmax4 = 0.2 + 1.2 = 1.4 \text{ тыс. ден. ед.}$$

Следовательно, оптимальное значение целевой функции при максимальном изменении ресурсов:

$$Gmax \approx 1.8 + 1.4 = 3.2 \text{ тыс. ден. ед.}$$

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

6.7 Консольный результат программы

Результат работы программы показан ниже (Рисунки 6.1, 6.2).

```
Первая теорема двойственности
Оптимальное решение прямой задачи:
x1 = 42.86, x2 = 42.86
Максимальное значение целевой функции: 24000
Оптимальное решение двойственной задачи:
y1 = 5, y2 = 0, y3 = 0, y4 = 60
Минимальное значение целевой функции: 24000
Проверка первой теоремы: f_max == g_min
24000 == 24000

Вторая теорема двойственности
Проверка условий дополняющей нежесткости:
Ограничение 1: 1200.00 <= 1200
  Ресурс дефицитный, y1 = 5.00
Ограничение 2: 25.71 <= 30
  Ресурс недефицитный, y2 = 0
Ограничение 3: 471.43 <= 600
  Ресурс недефицитный, y3 = 0
Ограничение 4: 300.00 <= 300
  Ресурс дефицитный, y4 = 60.00
```

Рисунок 6.1 – Первая и вторая теоремы двойственности

```
Третья теорема двойственности
Ресурс 1 (1200):
  Нижняя граница: 300.00
  Верхняя граница: 400.00
  Допустимый интервал: (900.00, 1600.00)
Ресурс 2 (30):
  Нижняя граница: 4.29
  Верхняя граница: inf
  Допустимый интервал: (25.71, inf)
Ресурс 3 (600):
  Нижняя граница: 128.57
  Верхняя граница: inf
  Допустимый интервал: (471.43, inf)
Ресурс 4 (300):
  Нижняя граница: 75.00
  Верхняя граница: 450.00
  Допустимый интервал: (225.00, 750.00)
```

Рисунок 6.2 – Третья теорема двойственности

6.8 Вывод по двойственному методу

В ходе данной работы было найдено оптимальное решение двойственной задачи линейного программирования. Для этого на основе прямой задачи строится двойственная. Каждому ограничению исходной задачи соответствует неизвестное в двойственной задаче. А целевая функция двойственной задачи оптимизируется противоположно целевой функции прямой задачи. Далее устанавливаются соответствия двойственной и прямой задач с помощью теорем двойственности.

Из преимуществ двойственного метода можно выделить быстрое получение решения, также метод дает понимание о чувствительности решения к изменениям. Однако, для метода задачи должны быть представлены в каноническом виде, а сам алгоритм требуют более углубленных знаний математики

ЗАКЛЮЧЕНИЕ

Таким образом, примененные методы принятия решений достаточно эффективны для решения задач оптимизации с несколькими критериями, а также задач, нацеленных на оптимальное распределение ресурсов относительно выбранных критериев. Для задачи с большим количеством критериев и альтернатив – выбор лучшего VPN-сервиса использовались методы, учитывающие веса критериев (Метод Парето, Электра II, МАИ). В задачах линейного программирования применяются методы, требующие более математического подхода. Например, графический метод для задач с двумя переменными, а для более сложных задач — симплексный метод и его модификации. Тем не менее, каждый из рассмотренных способов решения имеет свои ограничения и области наилучшего применения.

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. Ростовцев, В. С. Теория принятия решений : учебное пособие / В. С. Ростовцев. — 2-е изд., перераб. и доп. — Киров : ВятГУ, 2021. — 192 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/201932 (дата обращения: 01.04.2025).

5. Постников, В. М. Методы принятия решений в системах организационного управления : учебное пособие / В. М. Постников, В. М. Черненький. — Москва : МГТУ им. Баумана, 2014. — 205 с. — ISBN 978-5-7038-3946-1. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/10662 (дата обращения: 25.04.2025)

6. Юдин Д.Б. Вычислительные методы теории принятия решений – 2-е изд. – М.: КРАСАНД, 2010. – 320с.

7. Единственность решения задачи линейного программирования: Избранные лекции по экстремальным задачам/ Под редакцией В.Н. Малозёмова – Том Часть 1 – С.: ООО "Издательство ВВМ", 2017. – 35- 37с.

8. Борзунов С.В., Кургалин С.Д. Языки программирования. Python: решение сложных задач: учебное пособие — Воронеж: , 2021. — 192 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/319391 (дата обращения: 01.04.2025).

9. Довгучиц, С. И. Системы поддержки принятия решений. Теория принятия решений : учебное пособие / С. И. Довгучиц, И. О. Паршин. — Москва : РТУ МИРЭА, 2023. — 112 с. — ISBN 978-5-7339-2013-9. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/398141 (дата обращения: 15.04.2025)

10. Пулькин, И. С. Теория принятия решений в условиях информационных конфликтов : учебное пособие / И. С. Пулькин. — Москва : РТУ МИРЭА, 2023. — 65 с. — ISBN 978-5-7339-1848-8. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/382409 (дата обращения: 15.04.2025).

11. Болотский, А. В. Математическое программирование и теория игр : учебное пособие для вузов / А. В. Болотский. — 5-е изд., стер. — Санкт-Петербург : Лань, 2024. — 116 с. — ISBN 978-5-507-50227-1. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/414734 (дата обращения: 15.04.2025).

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 - Реализация Парето.

```
import csv
import pandas as pd
from tabulate import tabulate

# Функция для вывода таблицы
def print_table(data, headers):
    df = pd.DataFrame(data)
    print(tabulate(df, headers = 'keys', tablefmt = 'pretty', showindex =
False))


# Функция для сравнения альтернатив по отношению Парето-доминирования
def compare_alternatives(a, b):
    better = False
    worse = False

    for key in a:
        if key.endswith('(+)'):
            if float(a[key]) > float(b[key]):
                better = True
            elif float(a[key]) < float(b[key]):
                worse = True
        elif key.endswith('(-)'):
            if float(a[key]) < float(b[key]):
                better = True
            elif float(a[key]) > float(b[key]):
                worse = True
    if better and not worse:
        return 1
    elif worse and not better:
        return -1
    return 0


# Функция для создания оптимального множества Парето
def create_Pareto_set(data):
    pareto_front = []
    dominated_alternatives = set()

    for i in range(len(data)):
        dominates_any = False  # Доминирует ли текущая альтернатива хотя бы одну
другую
        dominated = False  # Доминируется ли текущая альтернатива

        for j in range(len(data)):
            if i == j:
                continue
            comparison = compare_alternatives(data[i], data[j])
            if comparison == 1:
                dominates_any = True
            elif comparison == -1:
                dominated = True
                break
```

Продолжение листинга А

```
        if not dominated and dominates_any:
            pareto_front.append(data[i])
        elif dominated:
            dominated_alternatives.add(data[i]['№'])
    return pareto_front

# Метод указания верхних/нижних границ критериев
def bounds_optimization(data, bounds):
    winners = []
    for i in data:
        flag = True
        for j in bounds:
            key, value = list(j.items())[0]
            if key.endswith('(+)'):
                if float(i[key]) < value:
                    flag = False
                    break
            elif key.endswith('(-)'):
                if float(i[key]) > value:
                    flag = False
                    break
        if flag:
            winners.append(i)
    return winners


# Метод субоптимизации
def suboptimization(data, branches, main_criteria):

    filtered_data = bounds_optimization(data, bounds)
    if not filtered_data:
        return []

    # Находим минимальное значение для главного критерия
    if main_criteria.endswith('(-)'):
        min_value = min(float(item[main_criteria]) for item in filtered_data)
        return [item for item in filtered_data if float(item[main_criteria]) ==
min_value]
    elif main_criteria.endswith('(+)'):
        max_value = max(float(item[main_criteria]) for item in filtered_data)
        return [item for item in filtered_data if float(item[main_criteria]) ==
max_value]

# Лексикографический метод
def lex_optimization(data, priority):
    def sorting_key(item):
        return tuple(float(item[key]) if key.endswith('(+)') else -
float(item[key]) for key in priority)
    sorted_data = sorted(data, key=sorting_key, reverse=True)

 return [sorted_data[0]]

# Чтение данных из CSV-файла
with open('tpr_1.csv', encoding='utf-8') as file:
    data = [d for d in csv.DictReader(file)]

    headers = ["№", "Варианты решений", "Средний чек (руб.) (-)", "Рейтинг на
Яндекс.Картах (+)",
               "Удаленность от метро (м) (-)", "Среднее время пути (мин) (-)",
"Количество отзывов (+)"]
```

Продолжение листинга А

```
    print("Исходная таблица с альтернативами и критериями:".center(201))
    print_table(data, headers)

    print("Оптимальное множество Парето:".center(201))
    pareto_set = create_Pareto_set(data)
    print_table(pareto_set, headers)

    print("Метод указания верхних/нижних границ:".center(201))
    bounds = [{"Средний чек (руб.) (-)": 700}, {"Количество отзывов (+)": 2000}]
    filtered_data = bounds_optimization(pareto_set, bounds)
    print_table(filtered_data, headers)

    print("Субоптимизация:".center(201))
    bounds = [{"Рейтинг на Яндекс.Картах (+)": 4.7}, {"Среднее время пути (мин)
(-)": 23}]
    main_criteria = "Средний чек (руб.) (-)"
    subopt_result = suboptimization(pareto_set, bounds, main_criteria)
    print_table(subopt_result, headers)

    print("Лексикографическая оптимизация:".center(201))
    priority = ("Средний чек (руб.) (-)", "Рейтинг на Яндекс.Картах
(+)","Среднее время пути (мин) (-)","Количество отзывов (+)",
                "Удаленность от метро (м) (-)", )
    lex_result = lex_optimization(pareto_set, priority)
    print_table(lex_result, headers)
```

Приложение Б

Код реализации метода Электра II на языке Python.

Листинг Б - Реализация Электра II.

```
import numpy as np
from tabulate import tabulate

def create_pref_matrix(alternatives, weight, directions):
    n = len(alternatives)
    P = np.zeros((n, n))
    N = np.zeros((n, n))

    for i in range(n):
        for j in range(n):
            if i == j:
                continue
            P_ij = 0
            N_ij = 0
            for k in range(len(weight)):
                if directions[k] == '+':
                    if alternatives[j][k] > alternatives[i][k]:
                        P_ij += weight[k]
                    elif alternatives[j][k] < alternatives[i][k]:
                        N_ij += weight[k]
                else:
                    if alternatives[j][k] < alternatives[i][k]:
                        P_ij += weight[k]
                    elif alternatives[j][k] > alternatives[i][k]:
                        N_ij += weight[k]
            if N_ij == 0:
                D_ij = float ('inf')
            else:
                D_ij =  P_ij / N_ij

            if D_ij > 1:
                P[i][j] = np.round(D_ij, 2) #Округляем до 2 цифр после запятой
            else:
                P[i][j] = 0
    return P

#Создание матрицы с порогом
def create_porog(matrix, porog):
    return np.where(matrix >= porog, matrix, 0)

def find_best_alt(matrix):
    n = matrix.shape[0]
    output = np.sum(matrix > 0, axis=1)
    input = np.sum(matrix > 0, axis=0)

    best_alt = []
    for i in range(n):
        best_alt.append((i + 1, output[i], input[i]))

    # Сортировка по количеству выходов и входов
    best_alt.sort(key=lambda x: (-x[1], -x[2]))

    return best_alt
```

Продолжение листинга Б

```
# Данные
alternatives = [
    [8, 12, 6, 4, 9],
    [12, 8, 12, 8, 9],
    [12, 4, 6, 8, 9],
    [12, 4, 6, 8, 6],
    [4, 8, 6, 4, 6],
    [4, 12, 6, 12, 12],
    [12, 12, 6, 8, 3],
    [4, 4, 6, 8, 9],
    [8, 4, 6, 12, 6],
    [4, 4, 6, 8, 9]
]
weight = [5, 4, 2, 4, 1]
directions = ['-', '+', '-', '-', '+']

# Создание матрицы предпочтений
pref_matrix = create_pref_matrix(alternatives, weight, directions)
print("Матрица предпочтений:")
print(tabulate(pref_matrix, headers=range(1, len(alternatives) + 1),
showindex=range(1, len(alternatives) + 1), tablefmt="pretty"))

# Создание матрицы предпочтений с порогом
porog = 4
porog_matrix = create_porog(pref_matrix, porog)
print("\nМатрица предпочтений с порогом C =", porog)
print(tabulate(porog_matrix, headers=range(1, len(alternatives) + 1),
showindex=range(1, len(alternatives) + 1), tablefmt="pretty"))

# Определение лучших альтернатив
best_alternatives = find_best_alt(porog_matrix)
print("\nЛучшие альтернативы:")
for alt in best_alternatives:
    print(f"Альтернатива {alt[0]}: Выходы = {alt[1]}, Входы = {alt[2]}")
```

Приложение В

Код реализации метода МАИ на языке Python.

Листинг В - Реализация МАИ.

```
import numpy as np
from tabulate import tabulate

SIZE = 5
CI = 1.12

class MAI:
    def __init__(self):
        self.criteria = [f"K{i + 1}" for i in range(SIZE)]
        self.alternatives = [f"A{i + 1}" for i in range(SIZE)]
        self.cr_comparison = None
        self.alt_comparison = {}
        self.results = {
            'cr_weights': None,
            'alt_weights': {},
            'priorities': None,
            'sogl': {}
        }

    def set_cr_comparison(self, comp_matrix):
        self.cr_comparison = np.array(comp_matrix)
        weights = self.calc_weights(self.cr_comparison)
        os = self.check_sogl(self.cr_comparison, weights)

        print("\nМатрица попарного сравнения критериев:")
        self.print_comp_matrix(self.cr_comparison, self.criteria, weights)
        print(f"Отношение согласованности: {os:.3f}")

    def set_alt_comparison(self, criteria, comp_matrix):
        self.alt_comparison[criteria] = np.array(comp_matrix)
        weights = self.calc_weights(self.alt_comparison[criteria])
        os = self.check_sogl(self.alt_comparison[criteria], weights)

        print(f"\nМатрица сравнения альтернатив по критерию {criteria}:")
        self.print_comp_matrix(self.alt_comparison[criteria], self.alternatives,
weights)
        print(f"Отношение согласованности: {os:.3f}")

    def print_comp_matrix(self, matrix, items, weights):
        table = []
        for item, row, weight in zip(items, matrix, weights):
            table.append([item] + [f"{x:.2f}" for x in row] + [f"{weight:.3f}"])

        print(tabulate(table, headers=[""] + items + ["W"], tablefmt="plain"))

    def calc_weights(self, comp_matrix):
        geometric_mean = np.array([(np.prod(row) ** (1 / SIZE)) for row in
comp_matrix])
        norm_weights = geometric_mean / geometric_mean.sum()
        return norm_weights
```

Продолжение листинга В

```
    def check_sogl(self, comp_matrix, weights):
        weighted_sum = np.sum(comp_matrix * weights, axis=1)
        lambda_max = np.mean(weighted_sum / weights)
        i_sogl = (lambda_max - SIZE) / (SIZE - 1)
        o_sogl = i_sogl / CI
        return o_sogl

    def analyze(self):
        # Анализ критериев
        cr_weights = self.calc_weights(self.cr_comparison)
        os = self.check_sogl(self.cr_comparison, cr_weights)
        self.results['cr_weights'] = dict(zip(self.criteria, cr_weights))

        # Анализ альтернатив по каждому критерию
        for criterion in self.criteria:
            matrix = self.alt_comparison[criterion]
            weights = self.calc_weights(matrix)
            os = self.check_sogl(matrix, weights)
            self.results['alt_weights'][criterion] = dict(zip(self.alternatives,
weights))
            self.results['sogl'][criterion] = os

        # Расчет приоритетов (W)
        priorities = {
            alt: sum(w * self.results['alt_weights'][crit][alt]
                     for w, crit in zip(cr_weights, self.criteria))
            for alt in self.alternatives
        }

        print("\nПриоритеты:")
        for rank, (alt, p) in enumerate(sorted(priorities.items(), key=lambda x:
-x[1]), 1):
            print(f"{rank}. {alt}: {p:.3f}")

        best_alt = max(priorities.items(), key=lambda x: x[1])
        print(f"\nЛучшая альтернатива: {best_alt[0]} (приоритет:
{best_alt[1]:.3f})")

        return self.results


def main():
    mai = MAI()
    # Матрица сравнения критериев
    criteria_matrix = [
        [1, 1, 3, 6, 8],
        [1, 1, 2, 5, 6],
        [1 / 3, 1 / 2, 1, 3, 5],
        [1 / 6, 1 / 5, 1 / 3, 1, 2],
        [1 / 8, 1 / 6, 1 / 5, 1 / 2, 1]
    ]
    mai.set_cr_comparison(criteria_matrix)
    # Матрицы сравнения альтернатив
    mai.set_alt_comparison("K1", [
        [1, 5, 6, 9, 2],
        [1/5, 1, 3, 4, 1 / 7],
        [1/6, 1 / 3, 1, 3, 1 / 8],
        [1 / 9, 1 / 4, 1 / 3, 1, 1 / 7],
        [1 / 2, 7, 8, 7, 1]
    ])
```

Продолжение листинга В

```
    mai.set_alt_comparison("K2", [
        [1, 2, 4, 4, 2],
        [1/2, 1, 2, 2, 1],
        [1/4, 1/2, 1, 1, 2],
        [1/4, 1/2, 1, 1, 1/2],
        [1/2, 1, 1/2, 2, 1]
    ])

    mai.set_alt_comparison("K3", [
        [1, 7, 2, 3, 3],
        [1/7, 1, 1/7, 1/6, 1/6],
        [1/2, 7, 1, 2, 2],
        [1/3, 6, 1 / 2, 1, 1],
        [1/3, 6, 1 / 2, 1, 1]
    ])

    mai.set_alt_comparison("K4", [
        [1, 5, 4, 5, 2],
        [1/5, 1, 1, 1, 1/2],
        [1/4, 1, 1, 1, 1/2],
        [1/5, 1, 1, 1, 1/2],
        [1/2, 2, 2, 2, 1]
    ])

    mai.set_alt_comparison("K5", [
        [1, 3, 3, 5, 2],
        [1/3, 1, 2, 1/2, 1],
        [1/3, 1/2, 1, 2, 1],
        [1/5, 2, 1/2, 1, 1/2],
        [1/2, 1, 1, 2, 1]
    ])

    results = mai.analyze()


if __name__ == "__main__":
    main()
```

Приложение Г

Код реализации Симплексного метода на языке Python.

Листинг Г - Реализация Симплексного метода.

```
import numpy as np

def print_table(table, basis, iteration, cb=None):
    # Определяем небазисные переменные (все кроме A0)
    all_x = list(range(table.shape[1] - 1))
    non_basis = [x for x in all_x if x not in basis]

    print(
        f"\n{'Начальная симплекс-таблица' if iteration == -1 else f'Итерация
{iteration}' }")

    # Заголовки
    headers = ["Св", "Базис"] + [f"X{x+1}" for x in non_basis] + ["A0"]
    print(" ".join(f"{h:>8}" for h in headers))

    if iteration == -1:
        print(f"{'cj':>8} {'':>8} {'260':>8} {'300':>8} {'':>8}")

    # Строки с коэффициентами
    for i in range(len(basis)):
        row = [f"{cb[i]:.1f}" if cb is not None else "0.0",
               f"x{basis[i] + 1}"]
        row += [f"{table[i, x]:.2f}" for x in non_basis]
        row.append(f"{table[i, -1]:.2f}")
        print(" ".join(f"{item:>8}" for item in row))

    # Строки F и дельта
    f_row = ['F', ''] + [f"{table[-1, x]:.2f}" for x in non_basis] + [f"{table[-
1, -1]:.2f}"]
    delta_row = ['Delta', ''] + [f"{table[-1, x]:.2f}" for x in non_basis] +
['']

    print(" ".join(f"{item:>8}" for item in f_row))
    print(" ".join(f"{item:>8}" for item in delta_row))


def simplex_method(c, A, b, accuracy):
    num_c = len(c) # Количество переменных
    num_b = len(b) # Количество ограничений

    # Создаем начальную таблицу
    table = np.zeros((num_b + 1, num_c + num_b + 1))
    # Заполняем коэффициенты ограничений
    table[:-1, :num_c] = A
    # Добавляем единичную матрицу для доп.переменных
    table[:-1, num_c:num_c + num_b] = np.eye(num_b)
    # Заполняем ограничения
    table[:-1, -1] = b
    # Заполняем целевую функцию (минус для максимизации)
    table[-1, :num_c] = -np.array(c)
```

продолжение Листинга Г

```
    # Базисные переменные (x3 - x6)
    basis = list(range(num_c, num_c + num_b))
    cb = np.zeros(num_b)  # Коэффициенты базисных переменных

    iteration = -1
    print_table(table, basis, iteration, cb)

    for iteration in range(0, 5):

        # Проверяем оптимальность:
        # все дельта (последняя строка без последнего столбца)
        # должны быть >=0
        if all(table[-1, :-1] >= 0):
            break

        # Выбираем входящую переменную:
        # entering_x - индекс мин. дельты
        # если мин. дельта >=0, решение оптимально
        entering_x = np.argmin(table[-1, :-1])
        if table[-1, entering_x] >= 0:
            break

        # Выбираем выходящую переменную:
        # из каждой строки берем коэф. входящей переменной,
        # если он > 0, делим на него A0,
        # если он < 0 ставим inf
        ratios = []
        for row in table[:-1]:
            x = row[entering_x]
            if x > 0:
                ratios.append(row[-1] / x)
            else:
                ratios.append(float('inf'))

        # Находим мин.переменную из вычисленных,
        # если это inf, то у задачи нет решения
        leaving_x = np.argmin(ratios)
        if np.isinf(ratios[leaving_x]):
            print("Решение неограничено")
            return None

        # Обновляем базис:
        # меняем входящую на выходящую переменную и
        # обновляем коэф. целевой ф-ции для новой базисной переменной
        basis[leaving_x] = entering_x
        cb[leaving_x] = c[entering_x] if entering_x < num_c else 0

        # Поворот:
        # находим ведущий эл-т - pivot_x
        # делим строку на него (чтобы получить 1),
        # для остальных строк - обнуляем ведущий столбец
        pivot_x = table[leaving_x, entering_x]
        table[leaving_x] /= pivot_x
        for i in range(len(table)):
            if i != leaving_x:
                table[i] -= table[i, entering_x] * table[leaving_x]

        table = np.round(table, accuracy)
        print_table(table, basis, iteration, cb)
```

продолжение Листинга Г

```
    # Ищем решение: для каждой базисной переменной
    # вычисляем значение целевой ф-ции,
    # возвращаем решение и целевую ф-цию
    solution = np.zeros(num_c)
    for i, var in enumerate(basis):
        if var < num_c:
            solution[var] = table[i, -1]

    return solution, table[-1, -1]


# Параметры задачи
# Коэффициенты целевой функции
c = [260, 300]
# Матрица ограничений
A = [
    [16, 12],
    [0.2, 0.4],
    [6, 5],
    [3, 4]
]
# Ограничения
b = [1200, 30, 600, 300]
# Количество переменных
accuracy = 2

solution, optimal_value = simplex_method(c, A, b, accuracy)

if solution is not None:
    print("\nОптимальное решение:")
    print(f"x1 = {solution[0]:.2f}")
    print(f"x2 = {solution[1]:.2f}")
    print(f"Максимальный выпуск продукции: {optimal_value:.2f}")
```

Приложение Д

Код реализации Двойственной задачи на языке Pyhton.

Листинг Д - Реализация Двойственной задачи

```
import numpy as np

def inverse_matrix(D):
    D_inv = np.linalg.inv(D)
    return D_inv

def dual_method():
    c = np.array([260, 300])
    A = np.array([
        [16, 12],
        [0.2, 0.4],
        [6, 5],
        [3, 4]
    ])
    b = np.array([1200, 30, 600, 300])
    x_optimal = np.array([300/7, 300/7, 0, 30/7, 900/7, 0])
    D = np.array([
        [16, 12, 0, 0],
        [0.2, 0.4, 1, 0],
        [6, 5, 0, 1],
        [3, 4, 0, 0]
    ])


    print("Первая теорема двойственности")
    print("Оптимальное решение прямой задачи: ")
    print(f"x1 = {x_optimal[0]:.2f}, x2 = {x_optimal[1]:.2f}")
    f_max = np.dot(c, x_optimal[:2]) # Скалярное произведение
    print(f"Максимальное значение целевой функции: {f_max:.0f}")

    # Вычисление обратной матрицы
    #print("Обратная матрица:")
    D_inv = inverse_matrix(D)
    #print(D_inv)

    # Коэффициенты базисных переменных
    C_B = np.array([260, 300, 0, 0])
    # Решение двойственной задачи
    y_optimal = C_B @ D_inv # Матричное умножение

    print("Оптимальное решение двойственной задачи:")
    print(f"y1 = {y_optimal[0]:.0f}, y2 = {y_optimal[1]:.0f}, y3 =
{y_optimal[2]:.0f}, y4 = {y_optimal[3]:.0f}")
    g_min = np.dot(b, y_optimal) # Скалярное произведение
    print(f"Минимальное значение целевой функции: {g_min:.0f}")
    print("Проверка первой теоремы: f_max == g_min")
    print(f"{f_max:.0f} == {g_min:.0f}")


    print("\nВторая теорема двойственности")
    print("Проверка условий дополняющей нежесткости:")
    for i in range(len(b)):
        res = np.dot(A[i], x_optimal[:2])
        print(f"Ограничение {i + 1}: {res:.2f} <= {b[i]}")
```

продолжение Листинга Д

```
        if res - b[i] == 0:
            print(f"  Ресурс дефицитный, y{i + 1} = {y_optimal[i]:.2f}")
        else:
            print(f"  Ресурс недефицитный, y{i + 1} = 0")


    print("\nТретья теорема двойственности")

    for i in range(len(b)):
        column = D_inv[:, i]
        positive = column[column > 0]
        negative = column[column < 0]

        # Находим индексы базисных переменных опт. плана (x1, x2, x4, x5)
        basis_x = x_optimal[[0, 1, 3, 4]]

        if len(positive) > 0:
            delta_low = min(basis_x[column > 0] / positive)
        else:
            delta_low = -np.inf

        if len(negative) > 0:
            delta_high = max(abs(basis_x[column < 0] / negative))
        else:
            delta_high = np.inf

        print(f"Ресурс {i + 1} ({b[i]}):")
        print(f"  Нижняя граница: {delta_low:.2f}")
        print(f"  Верхняя граница: {delta_high:.2f}")
        print(f"  Допустимый интервал: ({b[i] - delta_low:.2f}, {b[i] +
delta_high:.2f})")
```