МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Тульский государственный университет»

Институт прикладной математики и компьютерных наук
Кафедра «Прикладная математика и информатика»

Утверждено на заседании кафедры
«Прикладная математика и информатика»
24 января 2023 г., протокол № 5

И.о. заведующего кафедрой

_____Н.В. Ларин

МЕТОДИЧЕСКИЕ УКАЗАНИЯ
по выполнению практических (семинарских) занятий
по дисциплине (модулю)
«Дополнительные главы исследования операций (модуля)»

основной профессиональной образовательной программы
высшего образования – программы магистратуры

по направлению подготовки
01.04.02 Прикладная математика и информатика

с направленностью (профилем)
Перспективные методы искусственного интеллекта
в сетях передачи и обработки данных

Форма обучения: очная

Идентификационный номер образовательной программы: 010402-03-23

Тула 2023 год

Разработчик методических указаний

Смирнов О.И., доцент каф. ПМиИ, к.ф.-м.н., доцент _____

(ФИО, должность, ученая степень, ученое звание) *(подпись)*

На практических занятиях по курсу «Дополнительные главы исследования операций» обсуждаются вопросы, закрепляющие и расширяющие материал, изложенный на лекциях. Рассматриваются некоторые расчетные задачи по решению практических задач из соответствующих разделов учебного курса.

Решение типовых задач курса базируется на учебных пособиях:

1. Воробьёв С.А. **Теория игр и исследование операций.**: Учеб. пособие. Тула, Изд–во ТулГУ. 2012. – 100 с.

2. Воробьёв С.А. **Модели и методы исследования операций**.: Учеб. пособие. Тула, Изд–во ТулГУ. 2007. – 148 с.

В эти пособиях приведены решения практически всех основных изучаемых вопросов. В конце каждой главы предлагаются индивидуальные задачи для каждого студента.

Дополнительно в качестве самостоятельных (аттестационных) предлагаются типовые задания с индивидуальными исходными данными.

Эти задания включают в себя следующие темы:

1. Построение моделей, приводящих к задачам линейного программирования
2. Построение моделей, приводящих к транспортным задачам
3. Решение задач целочисленного программирования методом отсечений
4. Решение задач целочисленного программирования методом ветвей и границ
5. Решение задач нелинейного программирования методом допустимых направлений
6. Построение Марковских сетевых моделей
7. Построение системы дифференциальных уравнений Колмогорова. Понятие предельных вероятностей состояний
8. Построение модели и определение основных характеристик СМО
9. Нахождение оптимальных решений матричных игр
10. Поиск точек равновесия по Нэшу, максиминных стратегий и стратегий угроз
11. Поиск оптимума по Парето
12. Нахождение решения арбитра и соответствующих ему оптимальных наборов частот

СОДЕРЖАНИЕ

Практическое занятие № 1.
«Элементы выпуклого анализа. Евклидово пространство. Выпуклые множества. Выпуклые функции»

Вот несколько типичных задач, приводящихся к задачам линейного программирования:

⇒ Есть предприятие, которое может выпускать продукцию нескольких различных типов. Для производства единицы каждого из них требуется определённое количество сырья и прочих ресурсов, запасы которых ограничены. Известна рыночная стоимость этих товаров и ограничения на возможный объём продажи. Требуется так составить план производства, чтобы получить максимально возможную прибыль.

⇒ Известно количество белков, жиров, углеводов и прочих веществ, необходимых человеку в течении суток. Дан список продуктов, для каждого из которых указано содержание всех этих веществ. Требуется составить суточный рацион минимальной массы (или стоимости).

⇒ Есть группа предприятий, размещенных в разных местах. Им требуется сырье, источники которого (склады) тоже находятся в разных местах. Есть система транспортного сообщения (тарифы на перевозку тоже разные). Требуется таким образом разместить заказы на сырьё, чтобы обеспечить заводы сырьем при минимуме расходов.

Эти задачи необходимо записать в виде математических моделей. Рассмотрим этот процесс на примере первой из них. Введём n переменных $x_1,x_2,...,x_n$ – количество продукции каждого вида, что в совокупности образует план выпуска. Очевидно, эти переменные удовлетворяют условиям $x_1 \geq 0$, $x_2 \geq 0$, ..., $x_n \geq 0$ – невозможно выпустить отрицательное количество продукции. Пусть имеется m ограничений по ресурсам. Обозначим максимальное доступное количество каждого из них как $b_1,b_2,...,b_m$. Введём коэффициенты расхода этих ресурсов. a_{ij} показывает, какое количество ресурса вида i пойдёт на изготовление единицы продукции вида j. Для определения наилучшего плана используем величины $c_1,c_2,...,c_n$ – прибыль, получаемую от реализации единицы продукции каждого вида. Общая прибыль при реализации конкретного плана составит величину $L=c_1x_1+c_2x_2+...+c_nx_n$. Очевидно, что оптимальным решением задачи будет набор переменных, доставляющих максимальное значение этой функции. Кроме того, этот набор должен удовлетворять ограничениям

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + ... + a_{1n}x_n \leq b_1 \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + ... + a_{2n}x_n \leq b_2 \\ ..... \\ a_{m1}x_1 + a_{m2}x_2 + a_{m3}x_3 + ... + a_{mn}x_n \leq b_m \end{cases}$$

т.е. суммарный расход любого ресурса не должен превосходить его имеющихся запасов.

Для удобства методы, применяемые для решения какого-либо класса задач исследования операций, обычно разрабатываются для задачи некоторого стан-

дартного вида, к которой в свою очередь можно свести любую задачу данного класса. Соответственно и для линейного программирования формулируется основная задача, сокращённо ОЗЛП. Её главная особенность заключается в том, что все ограничения записываются в виде равенств. Позже будет показано, что и задачи с ограничениями–неравенствами могут быть сведены к ОЗЛП.
Итак, имеется ряд переменных $x_1, x_2, \ldots, x_n$. Требуется найти такие неотрицательные значения этих переменных, которые удовлетворяли бы системе линейных уравнений

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \ldots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + \ldots + a_{2n}x_n = b_2 \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ a_{m1}x_1 + a_{m2}x_2 + a_{m3}x_3 + \ldots + a_{mn}x_n = b_n \end{cases} \quad (1)$$

и, кроме того, обращали бы в минимум линейную функцию

$$L = c_1x_1 + c_2x_2 + \ldots + c_nx_n. \quad (2)$$

Условимся считать ***допустимым решением*** ОЗЛП любую совокупность переменных

$$x_1 \geq 0,\ x_2 \geq 0,\ \ldots,\ x_n \geq 0. \quad (3)$$

удовлетворяющую уравнениям (1). ***Оптимальным*** решением будем называть то из допустимых решений, при котором линейная функция (2) обращается в минимум.
Заметим, что далеко не все авторы приводят ОЗЛП именно в таком виде. Существуют различные взгляды на то, как оно должно выглядеть. Например, часто “стандартным” считают поиск максимума функции (2), бывают и иные отличия. Но, зная методы решения для любой из “стандартных форм” задачи, не представляет труда найти решение для любой другой формы.
ОЗЛП не обязательно имеет решение. Возможно, уравнения в системе (1) противоречат друг другу, или же они имеют решение только вне области неотрицательных значений переменных, что противоречит условию (3), или же решения существуют, но среди них нет оптимального. Последний случай возможен, если область допустимых решений (ОДР) не замкнута в сторону уменьшения целевой функции – в результате чего для любого решения задачи можно найти другое решение с ещё меньшим значением этой функции, т.е. решения, лучшего всех остальных, не может быть в принципе.

Индивидуальные задания по теме № 1. «Элементы выпуклого анализа. Евклидово пространство. Выпуклые множества. Выпуклые функции»

Каждому студенту выдаются условия задачи: матрица расходов сырья на выпуск продукции, вектор запасов сырья и показатели дохода от его продажи.
Требуется:
1 – составить задачу выпуклого программирования по определению оптимального плана выпуска продукции;

2 – привести эту задачу к стандартному виду.

Практическое занятие № 2.
«Сильно выпуклые множества. Проекция точки на множество»

Как уже говорилось ранее, любая задача линейного программирования может быть приведена к стандартному виду и решена симплекс-методом. Однако не каждую задачу имеет смысл решать всегда именно таким способом. Некоторые задачи имеют специальный вид, позволяющий решить их более быстродействующими методами.

В частности, к таким относится транспортная задача. Она описана в начале второй главы – третий по порядку пример. Это безусловно задача линейного программирования, но она имеет достаточно специфический вид:

В m пунктах отправления имеются запасы некоторого груза в количестве $a_1, a_2, ..., a_m$ единиц соответственно. Этот груз требуется доставить в n пунктов назначения, в каждый из пунктов – соответственно $b_1, b_2, ..., b_n$ единиц груза. $a_1, a_2, ..., a_m$ называются запасами, а $b_1, b_2, ..., b_n$ – заявками. В классической постановке задачи сумма заявок равна сумме всех запасов

$$\sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_j \quad (1)$$

Если это равенство не выполняется, то применяются методы приведения задач к стандартной форме. Они рассмотрены ниже.

Также дана стоимость перевозки единицы товара с каждого склада каждому потребителю (физический смысл данной величины может быть не стоимостью, а расстоянием или временем перевозки, это для рассмотрения общих методов решения задачи непринципиально). Она задана матрицей

$$\begin{Bmatrix} c_{11} & c_{12} & ..... & c_{1n} \\ c_{21} & c_{22} & ..... & c_{2n} \\ ..... & & & \\ c_{m1} & c_{m2} & ..... & c_{mn} \end{Bmatrix}$$

Необходимо найти такой план перевозок, чтобы выполнить все заявки и при этом минимизировать общую (суммарную) стоимость всех перевозок.

Решение задачи представляет собой набор переменных x_{ij} – количество груза, отправляемого из i–ого пункта отправления в j–ый пункт назначения – всего $m \times n$ переменных, и все они *неотрицательны*.

На переменные x_{ij} наложены следующие условия:

Суммарное количество груза, выходящего из каждого пункта отправления, равно запасу груза в этом пункте. Всего получаем m ограничений–равенств:

$$\sum_{i=1}^{n} x_{ki} = a_k, \quad k = 1,...,m \quad (2)$$

Суммарное количество груза, приходящего на каждый пункт назначения, равно заявленному этим пунктом количеству. Всего получаем n ограничений–равенств:

$$\sum_{j=1}^{m} x_{jl} = b_l, \quad l = 1,\ldots,n \quad (3)$$

Целевая функция, которую надо минимизировать – стоимость всех перевозок

$$L = \sum_{i=1}^{n}\sum_{j=1}^{m} c_{ji}x_{ji} \Rightarrow \min \quad (4)$$

Это типичная задача линейного программирования, поскольку все ограничения и целевая функция линейны.

Но у данной задачи есть ряд особенностей. Главное из них – это то, что коэффициенты при всех переменных в условиях (2) и (3) равны **1** (единице). Кроме того, одно из уравнений в этой системе не является независимым и может быть пропущено. Всего получается свободных переменных $(n-1)(m-1)$. Именно такое количество значений x_{ij} должно обращаться в нуль в оптимальном плане (или больше в случае вырожденного плана, но не меньше). Эти переменные будем называть перевозками, и по прежнему мы используем понятия *допустимого* плана, *опорного* и *оптимального*.

Для поиска оптимального плана не требуется применять симплекс-метод, все манипуляции выполняем непосредственно в таблице, называемой транспортной таблицей. В неё записаны все условия задачи: пункты отправления и назначения, запасы и заявки, стоимость перевозок.

	B_1	B_2	.....	B_n	Запасы a_i
A_1	c_{11}	c_{12}	.....	c_{1n}	a_1
A_2	c_{21}	c_{22}	.....	c_{1n}	a_2
.....	.....	.....	.....	.....	.....
A_m	c_{m1}	c_{m1}	.....	c_{mn}	a_i
Заявки b_j	b_1	b_2	.....	b_n	$\sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_j$

Стоимость перевозки единицы товара помещаем в каждой клетке вверху справа. Те ячейки, в которые помещаем отличные от нуля значения перевозок, называем базисными, а остальные – свободными. При этом сумма этих элементов плана перевозок x_{ij} в каждой строке равна a_i, а в каждом столбце – b_j. Ячейку, содержащую x_{ij}, будем обозначать координатами (i,j). Заметим, что в отличие от ОЗЛП, решение ТЗ *всегда существует*.

В ячейках основной части транспортной таблицы используются все 4 угла – в каждый из них помещается какое-нибудь значение, используемое при нахождении оптимального плана перевозок.

Нахождение опорного плана. Самый простой способ построения опорного плана – “метод северо-западного угла”. Этот “научный термин” по сути означа-

ет, что двигаться мы будем слева направо и сверху вниз. Действительно, если представить основную часть транспортной таблицы в виде географической карты, то ячейка (1,1) будет её северо-западным углом, а ячейка (***m***,***n***) – её юго-восточным углом. Начинаем с ячейки 1,1 и удовлетворяем потребность первого потребителя за счёт первого склада. Если на складе ещё что-то после этого осталось, оно идёт на покрытие заявки второго потребителя, т.е. направо (ячейка 1,2), если же заявка полностью не удовлетворена, то остаток потребности удовлетворяется за счёт следующего склада, т.е. движение вниз к ячейке 2,1 – и так далее.

Важно заметить, что двигаемся мы на каждом шаге строго горизонтально или вертикально, никаких диагональных перемещений быть не должно. Если у нас одновременно заканчивается и запас на данном складе и полностью удовлетворяется заявка данного потребителя, то мы сначала делаем шаг вправо, к следующему потребителю, удовлетворяем его заявку с данного склада (перевозки равны нулю), и лишь после этого делаем шаг вниз, к следующему складу. Тем самым мы обеспечиваем постоянно одно и то же число базисных ячеек – пусть даже некоторые из них, вырожденные, содержат нулевые значения. Мы их будем принципиально отличать от оставшихся ячеек, называемых свободными и по определению всегда содержащих нулевые значения перевозок, так что в них ноль даже и не пишется.

Рассмотрим этот метод на примере – составим оптимальный план для таблицы:

	B_1	**B_2**	**B_3**	**B_4**	**B_5**	Запасы a_i
A_1	10 **18**	8 **27**	5 **3**	6	9	48
A_2	6	7	8 **30**	6	5	30
A_3	8	7	10 **9**	8 **12**	7 **6**	27
A_4	7	5	4	6	8 **20**	20
Заявки b_j	18	27	42	12	26	125

Мы сразу получили план перевозок, удовлетворяющий условиям (2) и (3). Он является опорным, т.к. базисных (не равных 0) переменных ровно $r = n+m-1 = 8$. Остальные “пустые” клетки соответствуют нулевым перевозкам, т.е. свободным переменным. Таких у нас получилось $(n-1)(m-1) = 12$.

Этот план не является оптимальным. Действительно, его стоимость получается 1039. Если, к примеру, перенести 18 единиц из ячейки 1,1 в 1,2 – а для того, чтобы сохранить баланс, те же 18 единиц перенести из ячейки 2,3 в 1,3. – то получится план, имеющий стоимость уже 913.

Число ненулевых переменных может быть меньше, чем $r=n+m-1$, т.е. в данном случае меньше 8, т.е. часть базисных переменных также равна 0. Это *вырожденный* план. Он возникает, когда одновременно заканчивается и запас на очередном складе, и заявка от очередного потребителя, и следующая заполненная ячейка будет располагаться не ниже (“южнее”) и не правее (“восточнее”), а “юго-восточнее” последней заполненной. Но мы предполагаем, что сначала сделан шаг направо, заполнена клетка (величина перевозки равна нулю) и после этого сделан шаг вниз. В дальнейшем ***всегда*** будем считать, что в плане перевозок $r=n+m-1$ базисных клеток, даже если в каких-то из них и помещаются “нулевые” значения перевозок. Это удобно для построения алгоритма.

Методы приведения к стандартному виду очевидно зависят от того, какое отличие было у приводимой задачи.

Транспортная задача с избытком запасов

Рассмотрим ситуацию, когда поданных заявок меньше, чем запасов на складах. Равенство (1) не выполняется.

$$\sum_{i=1}^{m} a_i > \sum_{j=1}^{n} b_j$$

В этом случае вводим ещё один (фиктивный) пункт назначения, “заявка” которого равна остатку груза, невостребованному потребителями:

$$b_{n+1} = \sum_{i=1}^{m} a_i - \sum_{j=1}^{n} b_j$$

Реально *перевозки* из этого ПН означают, что такое количество груза останется на данном складе. Стоимость перевозок для данного ПН в общем случае будет равна нулю ($c_{i,n+1}=0,\ i=1,...,m$), хотя можно ввести цену и отличную от нуля: стоимость или убытки от хранения единицы груза. Последнее имеет смысл, если эта стоимость на разных складах (пунктах отправления) существенно различна.

Транспортная задача с избытком заявок (с недостатком запасов)

Это задача прямо противоположна ранее рассмотренной. Здесь запасов не хватает на всех потребителей, то есть

$$\sum_{i=1}^{m} a_i < \sum_{j=1}^{n} b_j .$$

Есть несколько способов решения такой задачи.

Первый – вводится “фиктивный” $m+1$–ый склад, запасы на котором равны

$$a_{m+1} = \sum_{j=1}^{n} b_j - \sum_{i=1}^{m} a_i$$

и стоимость перевозок с этого склада любому потребителю равна нулю. На самом деле цифра перевозок с этого ПО какому-то потребителю означает, что именно такое количество груза из заявленного *не будет доставлена* данному

потребителю. Этот метод применим тогда, когда не важно, каким потребителям и в каком количестве недогрузить товар. Можно решить эту задачу и с учётом степени важности получателей груза. В этом случае вводятся различные стоимости перевозки единицы груза к каждому из потребителей, имеющие смысл штрафа за недопоставку единицы заказанного товара этому потребителю.

Второй способ – “исправляем” заявки, умножив каждую на коэффициент

$$k = \sum_{i=1}^{m} a_i \Big/ \sum_{j=1}^{n} b_j .$$

Тем самым мы в равной пропорции недогрузили потребителей. Возможны иные варианты, но все они сводятся к уменьшению заявок до выполнения равенства (1) и решения задач с “нормальным” балансом.

Индивидуальные задания по теме № 2.
«Сильно выпуклые множества. Проекция точки на множество»

Каждому студенту выдаются условия задачи: матрица стоимостей транспортировки, вектора заказов и заявок. Все задачи не сбалансированы.
Требуется:
1 – составить транспортную задачу линейного программирования по определению оптимального плана выпуска продукции;
2 – привести эту задачу к стандартному виду.

Практическое занятие № 3.
«Теорема Фаркаша»

В данном разделе рассматриваются методы, которые также могут быть отнесены к методам дискретной оптимизации. Характерной особенностью этих методов является наличие дополнительного требования о том, что все либо только часть переменных в оптимальном решении должны обладать свойством целочисленности, или как вариант – быть кратными некоторому заданному числу. Иногда встречаются и иные ограничения. Данные методы тесно связаны с задачами линейного программирования, однако дополнительные требования делают задачу нелинейной. Поэтому стандартные методы линейного программирования для решения этих задач непригодны, хотя в ряде случаев используются в качестве “вспомогательных” в составе более сложных алгоритмов решения целочисленных задач.

В качестве примера целочисленной задачи можно привести стандартную задачу составления оптимального по прибыли плана выпуска продукции предприятием в условиях ограниченных ресурсов – если вспомнить, что любой вид продукции мы можем выпустить лишь в целом количестве экземпляров. При этом округление до ближайшего целого числа далеко не всегда даёт наилучшее из целочисленных решений, а может вообще оказаться вне области допустимых решений – тогда как наилучшее из целочисленных решений может оказаться весьма далеко от нецелочисленного оптимума, найденного методами линейного программирования.

С другой стороны, стандартная транспортная задача не содержит в себе требования целочисленности, однако, если все граничные условия целочисленны, то полученный стандартными методами оптимальный план перевозок также оказывается целочисленным, несмотря на то, что для достижения этого не прилагалось никаких специальных усилий.

К транспортной задаче может быть приложено дополнительное требование не целочисленности, а дискретности, то есть кратности одному или одному из нескольких целых чисел. Представим себе транспортную сеть, связывающую склады и потребителей. Перевозки по этой сети осуществляет транспортная фирма, имеющая грузовики одного или нескольких номиналов грузоподъёмности. В таком случае стоимость перевозки от данного склада данному потребителю должно задаваться не для единицы груза, а для одного рейса грузовика (или для рейса по отдельности каждого номинала грузоподъёмности). Нам придётся по каждому маршруту отправлять количество груза, кратное грузоподъёмности. Возможен вариант отправки не полностью загруженного автомобиля, но стоимость рейса (аренды) будет точно такой же, как у полностью загруженного.

Есть ещё ряд целочисленных или дискретных транспортных задач, называемых задачей о рюкзаке (ранце), задачей о бомбардировщике и т.п. В наиболее сложных случаях “контейнер” может быть разбит на отсеки определённой ёмкости,

предназначенные для перевозки разных видов груза. Например, “бензовоз” с отсеками для перевозки горючего с различным октановым числом.
Можно встретить и несколько иной тип задач, также называемый в некоторых учебниках “задачей о рюкзаке”. Это задача, внешне сходная с задачей линейного программирования, имеющая линейную целевую функцию и линейные ограничения вида – найти максимум функции

$$\max(\min)Z = \sum_{j=1}^{n} c_j x_j$$

с линейными ограничениями

$$\sum_{j=1}^{n} a_{ij} x_i = b_i, \quad i = 1,\ldots,m$$

при этом на все переменные наложено стандартное для задач линейного программированмя условие неотрицательности

$$x_j \geq 0, \quad j = 1,\ldots,n$$

и, кроме того, на все переменные наложено дополнительное условие

$$x_j \in \{0;1\}, \quad j = 1,\ldots,n$$

таким образом, любая переменная может принимать только одно из двух значений – 0 или 1, т.е. мы как бы решаем, “класть в рюкзак” данный предмет или нет, но положить можем только одну штуку любого вида.
Теперь рассмотрим некоторую классификацию. Можно выделить:

1. задачи линейного программирования с дополнительным требованием целочисленности, причём они разделяются на полностью целочисленные и частично целочисленные (требование целочисленности касается не всех переменных, а только части);
2. транспортные задачи формально к целочисленным не относятся, но если исходные данные целочисленны, то получим и целочисленный оптимальный план;
3. Задачи с неделимостью (с кратностью некоторому числу, обычно целому);
4. Комбинаторные задачи (например, о коммивояжере) используются для решения задач, в которых число возможных решений заведомо конечно. Сюда же относятся задачи с булевыми переменными, т.е. принимающими только значения 0 и 1. Например, какое-то действие или назначение может либо состояться, либо нет. Хотя иногда комбинаторные и булевы задачи относят к разным группам и применяют методы решения отдельно – только комбинаторный или только булевый.

Любая классификация в некоторой мере условна, т.е. какую-либо задачу и метод можно без большой натяжки отнести к нескольким разным группам. Так, симплекс-метод ищет оптимальное решение не на всей ОДР – многограннике с бесконечным числом точек, а на конечном множестве вершин этого многогранника. Поэтому его иногда относят к комбинаторным методам и используют как составную часть комбинаторных алгоритмов.

Для решения подобных задач используются методы: отсечения, ветвей и границ, приближённые и ряд других. Далее рассмотрим некоторые наиболее употребительные из целочисленных и дискретных моделей и методов.

Индивидуальные задания по теме № 3.
«Теорема Фаркаша»

Рассматривается задача, полученная студентом при изучении темы №1. Её требуется дополнить условием целочисленности. Поскольку план может составлять лишь целое количество выпуска любого вида продукции.

Практическое занятие № 4.
«Задачи математического программирования.»

Бывают практические ситуации, когда вместо одного показателя W требуется оценивать эффективность операции по целой группе показателей $W_1, W_2, ..., W_k$. Некоторые из них надо увеличить, некоторые уменьшить. Решения обычно несовместимы, то есть если максимизируем один критерий, другой может сильно ухудшиться и т.п. Количественный анализ может здесь помочь выбросить явно нерациональные варианты. Эти ситуации также называются задачами многокритериальной оптимизации, также они рассматриваются в задачах выбора и принятия решений.

Рассмотрим геометрическую интерпретацию: пусть требуется составить суточный рацион, содержащий все необходимые для человека питательные вещества, из заданного набора продуктов. Пусть эта задача решена. Очевидно, что наборов продуктов, отвечающих этому требованию, будет несколько. По каким критериям выбрать из них наилучший? Для примера возьмём два критерия: стоимость рациона S и его вес m. Оба эти критерия желательно минимизировать. Но, как видим, экстремумы этих критериев не совпадают, самый лёгкий из них не является самым дешёвым. Таким образом окончательный выбор сделает человек, и этот выбор будет компромиссом между этими двумя критериями. Возможно, на выбор повлияют и другие соображения. Тем не менее очевидно, что при этом в любом случае будет выбран один из четырёх рационов, расположенных слева внизу. Они различаются и массой, и стоимостью, но по обоим этим критериям превосходят остальные 15, которые мы отбросим как заведомо невыгодные, поскольку для каждого из них при том же весе существуют более дешёвые, а при той же стоимости – более лёгкие.

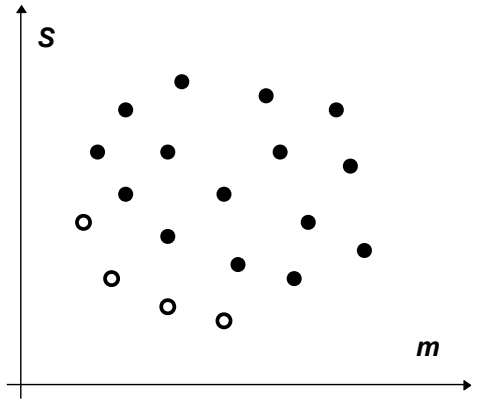

Рисунок 1.1. Операция с двумя показателями

Полученное подмножество решений называется **поверхностью Парето**. Члены этого множества удовлетворяют следующему условию: невозможно улучшить значение одного из показателей без того, чтобы при этом не ухудшилось значение хотя бы какого-то из оставшихся показателей.

Рассмотрим ещё один пример. Пусть есть 2 показателя – вероятность выполнения задачи P и стоимость S. Нас интересует максимум первого критерия при минимуме второго, но они явно не совпадают. Пусть есть набор критериев. Изобразим их на плоскости, в осях координат W и P. Очевидно, что предпо-

честь надо выделенные решения. Те, что лежат слева от них, имеют при той же стоимости более низкую вероятность выполнения задачи. Остаётся проанализировать эти 6 вариантов вместо 30. Нижний – самый дешевый, но с небольшой вероятностью. Верхний – более вероятный, но самый дорогой. Тут уже будет разбираться принимающий решения человек.

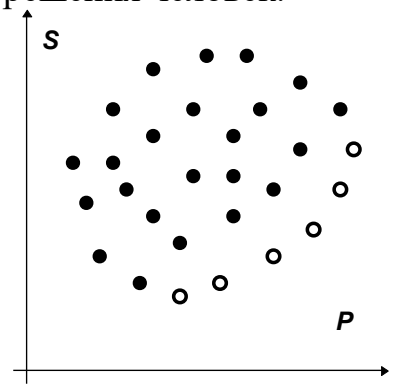

Рисунок 1.2. Варианты операций ко второму примеру

Разумеется, критериев может быть и больше двух. Принцип удаления заведомо неоптимальных решений при этом не изменится, но при этом не будет таких наглядных геометрических интерпретаций. Математических методов для этого существует достаточно много, они известны как “теория выбора” и “теория принятия решений”. Коротко рассмотрим наиболее часто применяемые методы окончательного выбора. При этом напомним, что достаточно часто задачи, принадлежащие к одной и той же группе, можно решать различными методами. Мы это увидим по мере изучения соответствующих разделов.

Построение обобщённых критериев

После того, как множество вариантов сокращено, надо выбрать из них один. Это можно сделать двумя способами – либо принимает решение человек, руководствуясь интуицией и рядом дополнительных соображений, не включённых в модель, либо строится *обобщённый критерий*, который включает в себя все частные критерии. Рассмотрим этот случай подробнее.

Пусть имеется $\boldsymbol{k}$ критериев $\boldsymbol{W_1, W_2, ..., W_k}$, часть из которых требуется максимизировать, а часть – минимизировать. Пусть максимизировать надо первые $\boldsymbol{m}$ критериев, а остальные минимизировать (иначе критерии можно просто перенумеровать. В качестве обобщённого критерия используем дробь

$$U = \frac{W_1 \times \ldots \times W_m}{W_{m+1} \times \ldots \times W_k}$$

Здесь в числителе те показатели, которые нужно увеличить, а в знаменателе – те, которые нужно уменьшить. Недостаток такого подхода – разнородные показатели могут взаимно компенсироваться, например, по обобщённому критерию будет выбран вариант с малой вероятностью осуществления, но зато очень дешёвый.

Существует и *другой способ* построения обобщённых показателей, более гибкий и позволяющий учесть больше особенностей задачи. Так, при выборе рациона кроме рассмотренных критериев – массы и стоимости – можно использовать и ряд других, например, длительность хранения, время и трудоёмкость

приготовления, стойкость к транспортировке и погодным условиям и ряд других. Очевидно, эти критерии имеют неодинаковое значение, но их сравнительная значимость сильно зависит от ряда факторов. Например, если рационы на достаточно большой срок требуется нести на себе, то наибольшую важность имеют вес и стойкость к транспортировке. Если их используют не перемещаясь – на первый план выходят другие критерии.

В таких случаях обобщённый показатель строится как линейная комбинация: $U=\alpha_1 W_1+\alpha_2 W_2+...+\alpha_k W_k$. Здесь коэффициенты $\alpha_1,\alpha_2,...,\alpha_k$ положительны при тех показателях, которые надо увеличить, и отрицательны при тех, которые надо уменьшить. Степень важности показателя определяется абсолютной величиной соответствующего ему коэффициента. Недостаток у этого способа – тот же, что и у ранее рассмотренного.

В ряде практических случаев задачу многокритериального выбора сводят к задаче с одним показателем. Выделяется главный показатель, по которому и проводится оптимизация, при этом остальные показатели превращают в ограничения, которые входят в комплекс заданных условий.

Метод последовательных уступок

Это способ построения компромиссного решения для многокритериальной задачи. Предполагается, что показатели $W_1, W_2, W_3,...,$ расположены в убывающей последовательности по значимости. Сначала идет самый основной показатель W_1, затем $W_2, W_3,...$ по степени убывания значимости. Предположим, что все эти критерии надо максимизировать. Если это не так, то соответствующие критерии умножаем на –1.

На первом шаге решается задача с одним критерием эффективности W_1, остальные показатели пока не рассматриваются. Находится решение, которое максимизирует главный показатель W_1. Теперь, когда оно найдено, назначается величина *уступки* ΔW_1, то есть величина, на которую мы согласны отклониться от максимума в показателе W_1, чтобы максимизировать показатель W_2. Вводим это ограничение в качестве ещё одного условия операции.

На следующем шаге с добавленным ограничением решаем задачу оптимизации для находим решение, обеспечивающее максимум второго показателя, затем определяется величина *уступки* ΔW_2 в показателе W_2, чтобы за счёт неё максимизировать показатель W_3. Процесс продолжается до тех пор, пока не будут рассмотрены все критерии.

Напомним, что все эти способы остаются не до конца формализованными, то есть окончательный выбор все равно связан с волевым решением человека. Данные методы предоставляют лишь возможность оценить преимущества и недостатки каждого варианта выбора.

Метод идеальной точки

Этот метод выбора окончательного решения из всего множества поверхности Парето, он также называется методом точки утопии. Метод применим, если в пространстве критериев $W_1, W_2, W_3,...,W_n$ все «оси координат» можно считать равноценными, то есть в этом пространстве можно определить (ввести) метри-

ку, т.е. понятие **расстояния**. Таким образом, мы не только получаем для каждого исхода операции чётко определённую точку в пространстве признаков, но ещё и задаём меру близости этих исходов.

В произвольном случае мы должны для каждого критерия выбрать соответствующий масштаб таким образом, чтобы одинаковое расстояние по любой из осей соответствовало бы равной степени значимости улучшения (или ухудшения) общей ситуации. Применительно к теории игр ситуация несколько упрощается. Здесь каждый из критериев соответствует одному и тому же понятию – выигрышу, просто относятся эти выигрыши к разным игрокам. Таким образом метрика в пространстве критериев определяется без каких-либо дополнительных действий.

Теперь рассмотрим сам метод. Мы знаем, что «Утопией» называли государство, имевшее множество достоинств и только один недостаток: в реальности оно не существовало. Также и «идеальная точка», она же «точка утопии» -- является идеальным, но недостижимым для данной системы операций исходом.

Пусть имеется многокритериальная задача с набором критериев W_1, W_2, W_3,...,W_n, причём на пространстве этих показателей определена метрика. Не ограничивая общности вопроса, будем считать, что все эти показатели следует максимизировать. Найдём по отдельности максимумы каждого из этих показателей, не обращая внимания на остальные: W_1 **max,** W_2 **max** и т.д. Очевидно, что эти значения несовместны.

Точку с координатами **(**W_1 **max,** W_2 **max,** W_3 **max,...,**W_n **max)** будем называть идеальной точкой и обозначать ***Ut***. Очевидно, что решением задачи такая точка не является, поскольку не принадлежит области допустимых решений – как и к поверхности Парето. Однако она позволяет легко сделать выбор из достижимых вариантов: окончательным решением будет та точка поверхности Парето, которая ближе всего к идеальной точке.

Рассмотрим двумерную задачу (рис. 1.3.):

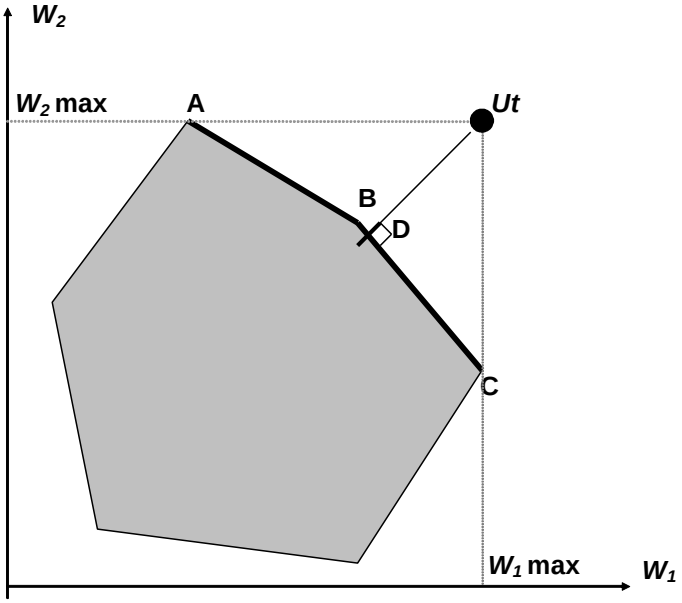

Рисунок 1.3. Точка утопии ***Ut*** и ближайшая к ней точка на поверхности Парето

Требуется максимизировать критерии W_1 и W_2. Геометрическая фигура на рисунке – область допустимых решений. Мы видим, что максимумам по первому и второму критериям W_1 **max** и W_2 **max** соответствуют различные точки -- **A** и **C**. Поверхность Парето состоит из двух отрезков – **AB** и **BC**. Точка утопии ***Ut*** не принадлежит ОДР. Ищем ближайшую к ней точку поверхности Парето. Как видим, это точка **D** на отрезке **BC**. Она и будет окончательным решением.

Индивидуальные задания по теме № 4.
«Задачи математического программирования»

Рассматривается множество решений на плоскости. Для него строится поверхность Парето и находится решение методом идеальной точки.

Практическое занятие № 5.
«Экстремальные свойства. Необходимые условия существования локальных минимумов. Экстремальные свойства на выпуклых множествах»

Пусть имеем стандартную задачу линейного программирования - поиск максимума функции

$$\max(\min)Z = \sum_{j=1}^{n} c_j x_j \quad (1)$$

с ограничениями (линейными)

$$\sum_{j=1}^{n} a_{ij} x_i = b_i, \quad i = 1,\ldots,m \quad (2)$$

при этом на все переменные наложено стандартное для задач линейного программированмя условие неотрицательности

$$x_j \geq 0, \quad j = 1,\ldots,n \quad (3)$$

и, кроме того, на все переменные наложено дополнительное условие

x_j – целые, $j=1,\ldots,n$ (4)

Для большинства “целочисленных” методов требуется, чтобы все элементы в системе уравнений (2) были целочисленны. Этого нетрудно добиться умножением соответствующих уравнений на подходящее целое число, так что в принципе любую систему линейных уравнений можно привести к соответствующему виду.

Решение этой задачи находится следующим образом:

1. Сначала решаем обычную задачу линейного программирования для условий (1)–(3), игнорируя требование целочисленности (4) – например, стандартным симплекс-методом.
2. Получив оптимальное решение (если оно существует), проверяем его на целочисленность (4). Возможно, полученное решение “само” оказалось целочисленным, и нам не требуется прибегать к каким-либо дополнительным мерам. Если для всех переменных оно выполняется, то получено оптимальное решение задачи. Если задача (1)–(3) не имеет решения, то его не имеет и задача (1)–(4).
3. Если хоть одна из переменных в полученном оптимальном решении – не целое число, то строится дополнительное ограничение, которое отсекает часть области, в которой содержится оптимальное решение задачи (1)–(3) и не содержится ни одного допустимого значения задачи (1)–(4), т.е. ни одного целого «вектора».
4. Возвращаемся к задаче (1)–(3) и решаем её с учётом появившегося дополнительного ограничения. Если опять получено дробное решение, то опять вводим дополнительное ограничение и так до тех пор, пока не будет получено целочисленное решение.

Это так называемый алгоритм Гомори. Он позволяет за конечное число шагов прийти к оптимальному целочисленному решению, если оно существует.

Основа этого метода – составление дополнительного ограничения, так называемого “правильного отсечения”. Оно должно:
а – быть линейным;
б – отсекать от ОДР найденное оптимальное нецелочисленное решение задачи;
в – не затрагивать ни одной из целых “точек” исходной задачи.
Пусть задача без учёта условия целочисленности решена. Обозначим ***B*** множество индексов переменных, соответствующих *базисным* переменным оптимального решения. Соответственно обозначим ***R*** множество индексов ***j***, которые соответствуют *свободным* переменным. После каждой итерации система ограничений, каждое из которых соответствует строке таблицы симплекс-метода имеет вид

$$x_i = b_i - \sum_{j \in R} a_{ij} x_j, \qquad i \in B \quad (5)$$

В случае, если выполняется условие оптимальности, находим оптимальное решение, в котором все свободные переменные равны нулю, а все базисные равны соответствующим свободным членам $x_j^* = 0, \quad j \in R; \quad x_i^* = \beta_i, \quad i \in B$. Если полученное решение целочисленно, то задача решена. Рассмотрим ситуацию, когда это не так, т.е. хотя бы один элемент решения принимает нецелочисленное значение.
Пусть какие-то b_i – нецелые. Например, b_{i0}.
Рассмотрим i_0 – ое равенство системы (5). При этом мы помним, что для данной системы выполнено условие оптимальности.

$$x_{i_0} = b_{i_0} - \sum_{j \in R} a_{i_0 j} x_j \quad (6)$$

Обозначим целую и дробную части числа ***a*** как **[*a*]** и **{*a*}**.
Напомним свойства целой и дробной части. Целая часть может быть какой угодно, но дробная всегда неотрицательна. Например **[*7,3*]=7** и **{7,3}=0,3**. Но **[-7,3]=– 8** и **{-7,3}=0,7**. **– 8 + 0,7 = – 7,3**.
Очевидно, что ***a*=[*a*]+{*a*}**. Слагаемые в уравнении (6) разделим на целые и дробные части и группируем. Получим

$$x_{i_0} = \left(\left[b_{i_0} \right] - \sum_{j \in R} \left[a_{i_0 j} \right] x_j \right) + \left(\left\{ b_{i_0} \right\} - \sum_{j \in R} \left\{ a_{i_0 j} \right\} x_j \right) \quad (7)$$

Выражение в левых круглых скобках – целое, а для того, чтобы x_{i0} было целым, необходимо, чтобы была целой и величина

$$L_{i_0} = \left\{ b_{i_0} \right\} - \sum_{j \in R} \left\{ a_{i_0 j} \right\} x_j$$

также была целой. А поскольку x_{i0} – координата *допустимого*, т.е. целочисленного решения задачи (1)–(4), то L_{i_0} – всегда целое число. Очевидно, что $L_{i_0} \le 0$. В самом деле, величина $\sum_{j \in R} \left\{ a_{i_0 j} \right\} x_j$ не может быть отрицательной, $0 \le \left\{ b_{i_0} \right\} \le 1$. Так как L_{i_0} – целое, то из предположения $L_{i_0} > 0$ следует $\left\{ b_{i_0} \right\} > 1$, чего не может быть, т.к. это дробная часть числа. Итак, любое допустимое решение задачи (1)–(4) должно удовлетворять неравенству

$$\{b_i\} - \sum_{j \in R} \{a_{ij}\} x_j \le 0$$

Это соотношение и определяет правильное отсечение Гомори (его называют также *первым* правильным отсечением Гомори).

Признаком отсутствия целочисленного решения служит появление в таблице хотя бы одной строки с дробными свободными членами и целыми остальными коэффициентами, *так как в этом случае соответствующее уравнение не имеет решения в целых числах.*

Теоретически всё получается достаточно надёжно, но вот при практической реализации алгоритмов, реализующих данную схему метода, возникает ряд проблем.

Первая проблема состоит в том, что какова бы ни была точность вычислений, в любом случае количество разрядов под запись числа “с плавающей точкой” ограничено, вычисления выполняются с некоторой конечной точностью, поэтому дробное число может восприниматься как целое и наоборот. Способ борьбы с таким недостатком – представление коэффициентов в таблице симплекс-метода в виде простых дробей, т.е. каждое число записывается в виде двух целых чисел, отдельно числителя и знаменателя.

Вторая проблема состоит в том, что после решения нецелочисленной задачи мы можем получить не одно, а несколько нецелочисленных значений в столбце свободных членов. Однако на каждом шаге алгоритма мы должны ввести только одно повое отсечение, т.е. выбрать из всех строк таблицы с нецелым свободным членом только одну строку. Как лучше сделать такой выбор? На практике оказывается, что от такого выбора существенно зависит скорость сходимости алгоритма. Гомори предложил два способа выбора строки таблицы для построения отсечения.

Первый способ. Если оптимальный план содержит несколько дробных компонент, то дополнительное ограничение рекомендуется записывать для компоненты с наибольшей дробной частью (т.е. для строки с наибольшей дробной частью свободного члена).

$$\max\{b_i\}$$

Второй способ – выбирается та строка, в которой достигнут максимум дробной части отношения свободного члена к сумме всех коэффициентов данной строки.

$$\max\left\{ b_i \Big/ \sum_{j \in R} a_{ij} \right\}$$

Второй способ обеспечивает более быструю сходимость.

Заметим также, что требование целочисленности в данном алгоритме распространяется на все переменные – как переменные исходной задачи, так и вновь добавляемые в процессе решения.

Пример. Рассмотрим применение этого метода на практическом примере, рассмотренном в разделе 2.4. – только дополним его требованием целочисленности. В итоге получим задачу:

Найти максимум целевой функции

$$z = 70x_1 + 30x_2 \to max$$

при ограничениях

$$\begin{cases} 10x_1 + 4x_2 \le 40 \\ 6x_1 + 4x_2 \le 36 \end{cases}$$
$$x_j \ge 0, \quad j = 1,2$$
$$x_j - целые, \quad j = 1,2$$

Видно, что задача имеет нестандартный вид – целевая функция стремится к максимуму, ограничения записаны в виде неравенств. Приведём задачу к стандартному виду. Сначала преобразуем целевую функцию так, чтобы она стремилась к минимуму

$$L = -z = -(70x_1 + 30x_2) \to min$$

Теперь преобразуем ограничения из неравенств в равенства, для чего к правой части каждого из неравенств добавим по дополнительному неотрицательному переменному

$$\begin{cases} 10x_1 + 4x_2 + x_3 = 40 \\ 6x_1 + 4x_2 + x_4 = 36 \end{cases}$$
$$x_j \ge 0, \quad j = 1,\ldots,4$$

Теперь надо решить эту задачу симплекс-методом без учёта требования целочисленности. Эта задача была решена в разделе 2.4. В итоге была получена таблица вида

Таблица 5.1

Базисные переменные	Свободный член	Свободные переменные	
		x_3	x_4
L	**–295**	**–25/4**	**–5/4**
x_1	**1**	**1/4**	**–1/4**
x_2	**15/2**	**–3/8**	**5/8**

Найденное оптимальное решение имеет вид:
Свободные переменные равны нулю $x_3=x_4=0$, базисные переменные равны соответствующим свободным членам $x_1=1$, $x_2=7{,}5$. Целевая функция $L = –295$.
Вспомним теперь требование целочисленности и проверим полученное решение. Требование не выполняется, так как переменная $x_2=7{,}5$ нецелочисленна. Следовательно, мы должны построить дополнительное отсечение. Выберем в таблице какую-нибудь строку с нецелочисленным свободным членом. Она там одна, самая нижняя (выделена), со свободным членом 15/2. Соответственно по рассмотренному правилу строим правильное отсечение Гомори

$$\frac{1}{2} - \left(\frac{5}{8}x_3 + \frac{5}{8}x_4\right) \le 0 \text{ или } \frac{5}{8}x_3 + \frac{5}{8}x_4 \ge \frac{1}{2}$$

Заметим, что хотя в строке при переменных x_3 и x_4 различные коэффициенты -- –3/8 и 5/8, но в отсечении получаем одинаковые коэффициенты 5/8. Это связано со свойством дробной части отрицательного числа – целая и дробная часть числа –3/8 соответственно равны

$$\left[-\frac{3}{8}\right] = -1 \quad \left\{-\frac{3}{8}\right\} = \frac{5}{8}$$

Полученное дополнительное ограничение записано в виде неравенства. Необходимо преобразовать его в стандартное ограничение - равенство. Для этого введём дополнительную неотрицательную переменную x_5. Получим

$$\frac{5}{8}x_3 + \frac{5}{8}x_4 - x_5 = \frac{1}{2}.$$

Выразим отсюда новую переменную , которую будем использовать в качестве базисной

$$x_5 = -\frac{1}{2} - \left(-\frac{5}{8}x_3 - \frac{5}{8}x_4\right)$$

Заметим, что эта переменная выражена через те же свободные переменные x_3 и x_4, что и остальные переменные в таблице 5.1. Таким образом нам не требуется возвращаться к самому началу решения, достаточно лишь добавить новую строку к таблице 5.1. В результате получим

Таблица 5.2

Базисные переменные	Свободный член	Свободные переменные	
		x_3	x_4
L	**–295**	**–25/4**	**–5/4**
x_1	**1**	**1/4**	**–1/4**
x_2	**15/2**	**–3/8**	**5/8**
x_5	**–1/2**	**–5/8**	**–5/8**

Рассмотрим полученную таблицу. Сразу заметим, что столбец свободных членов содержит отрицательный элемент –1/2, то есть это решение не является допустимым. Этого и следовало ожидать, поскольку дополнительное ограничение отсекает от ОДР найденное на предыдущем шаге оптимальное нецелочисленное решение.

Опять решим задачу, не обращая внимания на условие целочисленности. Выберем в таблице разрешающий элемент. Для этого рассмотрим строку с отрицательным свободным членом –1/2 и найдём в ней отрицательные коэффициенты. Их целых два, оба –5/8. Рассмотрим оба столбца, содержащих эти коэффициенты, в качестве кандидатов в разрешающие столбцы.

В каждом из столбцов найдём коэффициенты, имеющие одинаковый знак со своими свободными членами. Таких всего четыре (выделены в таблице 5.2). В строке x_1 это коэффициент 1/4, в строке x_2 это коэффициент 5/8, и наконец в

строке x_5 это оба коэффициента –5/8. Найдём теперь отношение соответствующих свободных членов к каждому из этих коэффициентов. В первой строке получим 1/(1/4)=4, во второй (15/2)/(5/8)=12, в третьей для обоих коэффициентов (–1/2)/(–5/8)=4/5, то есть наименьшее отношение получено для обоих коэффициентов последней строки. Поскольку оба эти коэффициента "равноправны", то произвольно выберем, например, второй из них – соответственно получим разрешающую строку и разрешающий столбец.

Таблица 5.3

Базисные переменные	Свободный член	Свободные переменные	
		x_3	$x_4 \to x_5$
L	**–295**	**–25/4**	**–5/4**
x_1	**1**	**1/4**	**–1/4**
x_2	**15/2**	**–3/8**	**5/8**
$x_5 \to x_4$	**–1/2**	**–5/8**	**–5/8**

Переменную x_4 будем перемещать в базисные, а x_5 наоборот – в свободные. Пересчитаем коэффициенты по известному нам (рассмотренному в разделе 2.4.) алгоритму, в результате получим

Таблица 5.4

Базисные переменные	Свободный член	Свободные переменные	
		x_3	$x_4 \to x_5$
L	**–295**	**–25/4**	**–5/4**
x_1	**1**	**1/4**	**–1/4**
x_2	**15/2**	**–3/8**	**5/8**
$x_5 \to x_4$	**–1/2**	**–5/8**	**–5/8**

Переменную x_4 будем перемещать в базисные, а x_5 наоборот – в свободные. Пересчитаем коэффициенты по известному нам (рассмотренному в разделе 2.4.) алгоритму, в результате получим

Таблица 5.5

Базисные переменные	Свободный член	Свободные переменные	
		x_3	x_5
L	**–294**	**–5**	**–2**
x_1	**6/5**	**1/2**	**–2/5**
x_2	**7**	**–1**	**5/8**
x_4	**4/5**	**1**	**–8/5**

Проанализируем результат. В столбце свободных членов (не считая целевой функции) все числа неотрицательны, значит решение осталось допустимым (опорным). В строке целевой функции (не считая свободного члена) все коэффициенты отрицательны, значит решение оптимально. Поскольку в этой строке нет нулевых элементов, делаем вывод – найденное оптимальное решение единственно.

Запишем это оптимальное решение. Свободные переменные равны нулю $x_3=x_5=0$, базисные переменные равны соответствующим свободным членам $x_1=6/5$, $x_2=7$, $x_4=4/5$. Целевая функция $L = -294$.

Опять проанализируем решение с точки зрения целочисленности. Имеем целых два нецелочисленных переменных, $x_1=6/5$ и $x_4=4/5$. Надо выбрать из этих двух строк одну, по которой провести отсечение. В первом случае дробная часть свободного члена равна 1/5, во втором 4/5, то есть больше. Выберем эту строку (в таблице 5.5. выделена). Построим отсечение. Условие будет иметь вид

$$\frac{4}{5}-\left(0x_3+\frac{2}{5}x_5\right)\le 0$$

При переменной x_3 в строке коэффициент 1, т.е. целое число, значит в неравенство x_3 входит с нулевым коэффициентом. Преобразуем ограничение-неравенство в равенство, для чего вводим дополнительно неотрицательную переменную x_6. В результате получим

$$\frac{4}{5}-\left(0x_3+\frac{2}{5}x_5\right)+x_6=0$$

или в преобразованном виде

$$x_6=-\frac{4}{5}-\left(0x_3-\frac{2}{5}x_5\right)$$

Добавляем к последней таблице ещё одну строку. Получаем

Таблица 5.6

Базисные переменные	Свободный член	Свободные переменные	
		x_3	x_5
L	**–294**	**–5**	**–2**
x_1	**6/5**	**1/2**	**–2/5**
x_2	**7**	**–1**	**5/8**
x_4	**4/5**	**1**	**–8/5**
x_6	**–4/5**	**0**	**–2/5**

Эту задачу опять надо решить симплекс-методом без учёта требования целочисленности. Рассмотрим полученную таблицу. Заметим, что столбец свободных членов содержит отрицательный элемент –4/5, то есть это решение не является допустимым, как и после добавления предыдущего отсечения.
Так же, как и в предыдущем случае, выберем разрешающий элемент и соответственно разрешающую строку и разрешающий столбец.

Таблица 5.7

Базисные переменные	Свободный член	Свободные переменные	
		x_3	$x_5 \rightarrow x_6$
L	**–294**	**–5**	**–2**
x_1	**6/5**	**1/2**	**–2/5**
x_2	**7**	**–1**	**5/8**
x_4	**4/5**	**1**	**–8/5**
$x_6 \rightarrow x_5$	**–4/5**	**0**	**–2/5**

Переменную x_5 будем перемещать в базисные, а x_6 наоборот – в свободные. Пересчитаем коэффициенты по табличному алгоритму преобразования коэффициентов, в результате получим

Таблица 5.8

Базисные переменные	Свободный член	Свободные переменные	
		x_3	x_6
L	**–290**	**–5**	**–5**
x_1	**2**	**1/2**	**–1**
x_2	**5**	**–1**	**5/2**
x_4	**4**	**1**	**–4**
x_5	**2**	**0**	**–5/2**

Проанализируем результат. В столбце свободных членов (не считая целевой функции) все числа неотрицательны, значит решение осталось допустимым (опорным). В строке целевой функции (не считая свободного члена) все коэффициенты отрицательны, значит решение оптимально. Поскольку в этой строке нет нулевых элементов, делаем вывод – найденное оптимальное решение единственно.

Запишем это оптимальное решение. Свободные переменные равны нулю $x_3=x_6=0$, базисные переменные равны соответствующим свободным членам $x_1=2$, $x_2=5$, $x_4=4$, $x_2=2$. Целевая функция $L = -290$.

Рассмотрим полученное решение с учётом условия целочисленности. Все свободные члены целочисленны, так что мы наконец получили целочисленное решение исходной задачи. С учётом того, что исходная задача содержала всего два переменных и преобразованную нами целевую функцию, можно записать найденное решение в виде:

$x_1=2$, $x_2=5$, целевая функция $Z = 290$.

Значение целевой функции на 5 единиц меньше, чем у оптимального нецелочисленного решения – это та цена, которую пришлось уплатить за выполнение условия целочисленности. Полученная точка (2; 5) достаточно далеко отстоит от точки оптимального целочисленного решения (1; 7,5), так что получить решение простым округлением до ближайшего целого числа явно не получилось бы.

Заметим также, что в данном примере второе отсечение мы строили по строке для переменной x_5, которая в исходную задачу не входила, а была добавлена при преобразовании условия первого отсечения к стандартному виду. Тем не менее к этой переменной также предъявляются требования целочисленности.

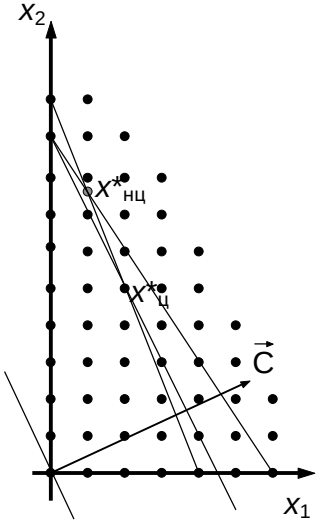

Рисунок 5.1. Геометрическая интерпретация рассмотренной задачи и её решения методом отсечений

На рисунке дана геометрическая интерпретация решения задачи. Показаны точка $x^*_{нц}$ нецелочисленного оптимального решения, линия, отсекающая её, и точка $x^*_{ц}$ оптимального целочисленного решения. Видно, что эта точка не могла быть получена округлением $x^*_{нц}$ до ближайшего целого значения.

Индивидуальные задания по теме № 5.
«Экстремальные свойства. Необходимые условия существования локальных минимумов. Экстремальные свойства на выпуклых множествах»

Задача, составленная в разделе 3, решается методом отсечений. Полученное решение сравнивается с оптимальным решением нецелочисленной задачи.

Практическое занятие № 6.
«Достаточные условия оптимальности. Теория множителей Лагранжа, теорема Куна-Таккера»

Пусть имеется стандартная задача целочисленного линейного программирования – требуется найти экстремум (для определённости пусть будет максимум) целевой функции

$$\max \quad Z = \sum_{j=1}^{n} c_j x_j$$

с линейными ограничениями

$$\sum_{j=1}^{n} a_{ij} x_i = b_i, \quad i = 1,\ldots, m$$

при этом на все переменные наложено условие неотрицательности

$$x_j \geq 0, \quad j = 1,\ldots, n$$

и условие целочисленности

$$\boldsymbol{x_j} \text{ – целые, } \boldsymbol{j=1,\ldots,n}$$

Будем решать эту задачу с использованием общей схемы метода ветвей и границ. В основу метода положен тот уже замеченный нами при рассмотрении методов отсечений факт, что целевая функция для целочисленного решения принимала “худшее” значение, чем для нецелочисленного – это позволяет находить верхние границы на подмножествах. Как и при решении задач методами отсечений, сначала решаем задачу линейного программирования, игнорируя требование целочисленности, каким-нибудь стандартным методом решения – например, симплекс-методом.

Если в результате мы получили полностью целочисленное решение, то дальнейших шагов не требуется. Если задача не имеет оптимального решения, то и целочисленная задача также не имеет оптимального решения. Если же хотя бы одна из компонент плана имеет отличную от нуля дробную часть, то будем искать целочисленное решение.

Пусть в найденном оптимальном нецелочисленном плане X^* имеется некоторое нецелое число x^*_i, для которого $\{x^*_i\}>0$. Поскольку и само значение $x^*_i>0$, то очевидно, что этот нецелочисленный элемент решения заключён в промежутке между двумя целыми числами $K_i<x^*_i<K_i+1$, где $K_i=[x^*_i]$ – целая часть числа x^*_i Очевидно, что в целочисленном решении данной задачи эта переменная примет либо значение K_i и меньше, либо значение K_i+1 и больше. Других целочисленных значений просто не существует. *Разветвим* наше множество решений. Добавим к системе ограничений в одном случае условие $x^*_i<K_i$, а во втором – условие $x^*_i >K_i+1$. В двух полученных задачах линейного программирования не будет ни одного общего целочисленного решения, при этом ни одного целочисленного решения из ОДР исходной задачи мы не потеряем, т.е. основное

условие разбиения на подмножества выполнено. При этом, правда, из ОДР исходной задачи будет удалена полоса $K_i<x^*_i<K_i+1$, но она не содержит ни одного целочисленного решения. С этой полосой будет удалено также найденное на предыдущем шаге оптимальное нецелочисленное решение, таким образом наблюдаем некоторое сходство с методами отсечений. Однако наряду с отсечением нецелочисленного решения мы делим оставшуюся ОДР на две части. Но какое-нибудь из двух подмножеств может оказаться пустым.

Теперь решим уже две полученные задачи линейного программирования каким-либо стандартным методом, не учитывая условия целочисленности. Проанализируем полученные решения. Всего возможны шесть вариантов исхода:

1. Обе получившиеся задачи не имеют оптимального решения. Вывод: исходная задача имела оптимальное нецелочисленное решение, но оптимального целочисленного решения она не имеет.
2. Одна из задач не имеет решения, другая имеет целочисленное оптимальное решение. Последнее и является решением исходной задачи.
3. Одна из задач не имеет решения, другая имеет оптимальное, но опять нецелочисленное решение. Выбираем какую-либо из нецелочисленных компонент этого решения и опять разбиваем задачу на две задачи, аналогично описанному ранее. Решаем их без учёта условия целочисленности и анализируем результат.
4. Обе задачи имеют решение, и оба решения целочисленны. В этом случае сравниваем найденные значения целевой функции и выбираем то, у которого оно больше.
5. Обе задачи имеют решение, и оба решения имеют нецелочисленные компоненты. Сравниваем целевые функции полученных решений. Выбираем задачу, для которой она больше – *с наибольшей вероятностью* оптимальное решение находится там. Разветвляем её на две задачи и решаем их.
6. Обе задачи имеют решение, одно из которых целочисленно, а другое имеет нецелочисленные компоненты. Сравниваем целевые функции полученных решений. Если больше целевая функция целочисленного решения, оно и является решением исходной задачи. Если же больше целевая функция нецелочисленного решения, то *возможно* искомое решение находится там. Выбираем одну из нецелочисленных компонент плана и производим разветвление этой задачи.

Заметим, что в случае вариантов 3, 5 и 6 после получения целочисленного решения нам необходимо будет выполнить процедуру обратного хода, то есть сравнить значение целевой функции найденного целочисленного решения с целевыми функциями ранее отброшенных задач. Возможно, придётся вернуться к какой-либо из этих задач и разветвлять уже её.

Индивидуальные задания по теме № 6. «Достаточные условия оптимальности. Теория множителей Лагранжа, теорема Куна-Таккера»

Задачу из предыдущей темы решаем методом ветвей и границ. Сравниваем результаты и трудоёмкость решения.

Практическое занятие № 7.
«Двойственность в выпуклом программировании»

Ведущее место среди прямых методов решения экстремальных задач занимает *градиентный метод* (точнее, семейство градиентных методов) поиска стационарных точек дифференцируемой функции. Напомним, что *стационарной* называется точка, в которой $\nabla f(x)=0$ и которая в соответствии с необходимым условием оптимальности является «подозрительной» на наличие локального экстремума. Таким образом, применяя градиентный метод, находят множество точек локальных максимумов (или минимумов), среди которых определяется максимум (или минимум) глобальный.

Идея данного метода основана на том, что градиент функции указывает направление ее *наиболее быстрого возрастания* в окрестности той точки, в которой он вычислен. Поэтому, если из некоторой текущей точки $x^{(1)}$ перемещаться в направлении вектора $\nabla f(x)=0$, то функция f будет возрастать, по крайней мере, в некоторой окрестности $x^{(1)}$. Следовательно, для точки $x^{(2)}=x^{(1)}+\lambda\nabla f\left(x^{(1)}\right),\ (\lambda>0)$, лежащей в такой окрестности, справедливо неравенство $f\left(x^{1}\right)\le f\left(^{(2)}\right)$. Продолжая этот процесс, мы постепенно будем приближаться к точке некоторого локального максимума (см. *рис. 2.1).*

Однако как только определяется направление движения, сразу же встает вопрос о том, как далеко следует двигаться в этом направлении или, другими словами, возникает проблема выбора шага λ, в рекуррентной формуле

$$x^{(q+1)}=x^{(q)}+\lambda\nabla f\left(x^{(q)}\right) \qquad (2.8)$$

задающей последовательность точек, стремящихся к точке максимума.

В зависимости от способа ее решения различают различные варианты градиентного метода. Остановимся на наиболее известных из них.

Метод наискорейшего спуска

Название метода можно было бы понимать буквально, если бы речь шла о минимизации целевой функции. Тем не менее, по традиции такое название используется и при решении задачи на максимум.

Пусть $f(x) = f(x_l,x_l,...x_n)$ — дифференцируемая функция, заданная на R^n, а $x^{(q)}=\left(x_1^{(q)},x_2^{(q)},...,x_n^{(q)}\right)$ — некоторая текущая точка. Оговоримся, что каких-либо общих рекомендаций, касающихся выбора исходной точки (или, как еще говорят, начального приближения) $x^{(0)}$, не существует, однако по возможности она должна находиться близко от искомого оптимального плана x^*. Как уже говорилось выше, если $x^{(q)}$ — нестационарная точка (т. е. $\left|\nabla f\left(x^{(q)}\right)\right|>0$), то при движении в направлении $\nabla f\left(x^{(q)}\right)$ функция $f(x)$ на некотором промежутке обязательно будет возрастать. Отсюда возникает естественная идея такого выбора шага, чтобы движение в указанном направлении продолжалось до тех пор, пока возрастание не прекратится. Для этого выразим зависимость значения $f(x)$ от шагового множителя $\lambda > 0$, полагая $x=x^{(q)}+\lambda\nabla f\left(x^{(q)}\right)$

$$f(x)=f\left(x^{(q)}+\lambda\nabla f\left(x^{(q)}\right)\right)=\varphi(\lambda) \qquad (2.9)$$

или, в координатной форме,

$$\varphi(\lambda)=f\left(x_1^{(q)}+\lambda\frac{\partial f\left(x^{(q)}\right)}{\partial x_1},...,x_n^{(q)}+\lambda\frac{\partial f\left(x^{(q)}\right)}{\partial x_n}\right) \qquad (2.10)$$

Чтобы добиться наибольшего из возможных значений f при движении по направлению $\nabla f\left(x^{(q)}\right)$, нужно выбрать такое значение $\tilde{\lambda}$, которое максимизирует функцию $\varphi(\lambda)$ $\left(\varphi\left(\tilde{\lambda}\right)=\max\limits_{\lambda>0}\varphi(\lambda)\right)$. Для вычисления $\tilde{\lambda}$, используется необходимое условие экстремума $d\varphi\left(\tilde{\lambda}\right)/d\lambda=0$. Заметим, что если для любого $\lambda>0$ $d\varphi\left(\tilde{\lambda}\right)/d\lambda>0$, то функция *f(x)* не ограничена сверху (т. е. не имеет максимума). В противном случае, на основе (2.10) получаем

$$\frac{d\varphi}{d\lambda}=\frac{\partial f(x)}{\partial x_1}\times\frac{dx_1}{d\lambda}+...+\frac{\partial f(x)}{\partial x_n}\times\frac{dx_n}{d\lambda} \qquad (2.11)$$

что, в свою очередь, дает

$$\frac{d\varphi}{d\lambda}=\frac{\partial f(x)}{\partial x_1}\times\frac{\partial f\left(^{(q)}\right)}{\partial x_1}+...+\frac{\partial f(x)}{\partial x_n}\times\frac{\partial f\left(x^{(q)}\right)}{\partial x_n}=\nabla f(x)\nabla f\left(x^{(q)}\right) \qquad (2.12)$$

Если считать, что следующая точка $x^{(q+1)}$ соответствует оптимальному значению $\lambda=\tilde{\lambda}$, то в ней должно выполняться условие $d\varphi\left(\tilde{\lambda}\right)/d\lambda=0$, и $\tilde{\lambda}$ следует находить из условия $\nabla f\left(x^{(q+1)}\right)\nabla f\left(x^{(q)}\right)=0$ или

$$\nabla f\left(x^{(q)}+\tilde{\lambda}\nabla f\left(x^{(q)}\right)\right)\nabla f\left(x^{(q)}\right)=0 \qquad (2.13)$$

Условие (2.13) означает равенство нулю скалярного произведения градиентов функции f точках $x^{(q+1)}$ и $x^{(q)}$ Геометрически оно может быть интерпретировано как перпендикулярность векторов градиентов функции f в указанных точках, что и показано на *рис. 2.2.* Продолжая геометрическую интерпретацию метода наискорейшего спуска, отметим, что в точке $x^{(q+1)}$ вектор $\nabla f\left(x^{(q+1)}\right)$, будучи градиентом, перпендикулярен линии уровня, проходящей через данную точку. Стало быть, вектор $\nabla f\left(x^{(q)}\right)$ является касательным к этой линии. Итак, движение в направлении градиента $\nabla f\left(x^{(q)}\right)$ следует продолжать до тех пор, пока он пересекает линии уровня оптимизируемой функции.

После того как точка $x^{(q+1)}$ найдена, она становится текущей для очередной итерации. На практике признаком достижения стационарной точки служит достаточно малое изменение координат точек, рассматриваемых на последовательных итерациях. Одновременно с этим координаты вектора $\nabla f\left(x^{(q)}\right)$ должны быть близки к нулю.

Метод дробления шага

Для нахождения шага λ, в методе наискорейшего спуска требуется решить уравнение (2.13), которое может оказаться достаточно сложным. Поэтому часто ограничиваются «подбором» такого значения λ, что $\varphi(\lambda)>\varphi(0)$. Для этого зада-

ются некоторым начальным значением λ_1 (например, $\lambda_1 = 1$) и проверяют условие $\varphi(\lambda_1) > \varphi(0)$. Если оно не выполняется, то полагают

$$\lambda_2 = \frac{1}{2}\lambda_1$$

и т. д. до тех пор, пока не удается найти подходящий шаг, с которым переходят к следующей точке $x^{(q+1)}$. Критерий завершения алгоритма, очевидно, будет таким же, как и в методе наискорейшего спуска.

Индивидуальные задания по теме № 7. «Двойственность в выпуклом программировании»

Студенту предлагается функция двух переменных, экстремум которой требуется найти. Это требуется проделать любым из изученных градиентных методов – на выбор студента. Сделанный выбор требуется обосновать.