МИНИСТЕРСТВО ВЫСШЕГО И СРЕДНЕГО СПЕЦИАЛЬНОГО ОБРАЗОВАНИЯ РЕСПУБЛИКИ УЗБЕКИСТАН

ТАШКЕНСТКИЙ АВТОМОБИЛЬНО-ДОРОЖНЫЙ ИНСТИТУТ

КАФЕДРА «ВЫСШАЯ МАТЕМАТИКА»

ТЕКСТЫ ЛЕКЦИЙ

по курсу

МАТЕМАТИЧЕСКОЕ ПРОГРАММИРОВАНИЕ

Ташкент 2007

М.У.Гафуров, Р.Х.Кенджаев, Ф.М.Закиров, Я.К.Лятиев
Математическое программирование. 2007 г. — 114 стр.

В основу текстов лекций положен семестровый курс математического программирования, читаемый авторами в течение ряда лет в Ташкенстком автомобильно-дорожном институте. В сборнике основные понятия и факты вводятся для задач линейного, выпуклого, квадратического и динамического программирования. Приведенные теоретические материалы проиллюстрированы большим числом примеров прикладного содержания.

В сборнике лекций содержатся следующие разделы: математические и экономические модели, постановка задач линейного, квадратического, выпуклого, динамического программирования и методы их решения, а также элементы теории игр.

Настоящий сборник рассчитан для студентов экономических специальностей высших экономических и технических учебных заведений, а также для всех заинтересованных в решении практических задач методами математического программирования.

Рецензенты: **Х.Жумаев** — доцент кафедры «Высшая математика» Ташкентского государственного экономического университета, кандидат физико-математических наук,

Х.Валиджанов — доцент кафедры «Высшая математика» Ташкенсткого автомобильно-дорожного института, кандидат физико-математических наук

Утверждено на заседании Научно-методического совета естественных и инженерных наук ТАДИ (протокол № 5 от 20 декабря 2006 года)

ОГЛАВЛЕНИЕ

Тема № 1

Введение в дисциплину «Математическое программирование». Понятие о математической модели. Модели экономических задач

План :

1. Предмет, составные части и краткая история развития дисциплины «Математическое программирование».
2. Понятие о математических моделях экономических задач.
3. Модель задачи использования сырья.
4. Модель задачи потребительской корзины.

Развитие современного общества характеризуется повышением технического уровня, усложнением организационной структуры производства, углублением общественного разделения труда, предъявлением высоких требований к методам планирования и хозяйственного руководства.

Вследствие того, что в настоящее время предприятия, научно-исследовательские институты, отрасли промышленности, производственные объединения и т.п. являются сложными комплексами, объединяющими большой коллектив персонала и оснащенными сложным оборудованием, для управления предприятиями требуется работа целого аппарата. Выполнение такой задачи, в свою очередь, обуславливает широкое использование точных математических методов и современных компьютерных технологий.

По велению времени в середине XX века возникла наука, называемая «Исследование операций». Причина наименования этой науки таким образом в том, что она является научной теорией, дающей основание управленческому аппарату делать правильные выводы путем анализа каждого его движения (операции) по достижению целей, стоящих перед предприятием. По мнению русского математика-экономиста Л.В.Канторовича, исследование операций является отраслью науки, обучающей путям правильного управления производственными процессами.

Одним из основных частей математического аппарата, применяемого при исследовании операций, является математическое программирование.

Одним из основных условий дальнейшего развития экономики является осуществление в ней численного анализа, основанного на математических методах и новых компьютерных технологиях, и принятие на этой основе экономических решений. Наукой, обучающей методам, пригодным в осуществлении таких задач, является математическое программирование.

Математическое программирование так же, как и математика, имеет давнюю историю, а ее неклассические отрасли сформировались в 30-40 годах XX века. Оно, являясь одним из направлений математики, в основном помогает находить самое лучшее, целесообразное (оптимальное) решение экономических задач, имеющих многовариантное решение.

Составными частями математического программирования являются линейное, нелинейное и динамическое программирование. Впервые постановку задачи линейного программирования в виде предложения по составлению оптимального плана перевозок, позволяющего минимизировать суммарный километраж, дал советский экономист А.Н.Толстой в 1930 году. В 1939 году Л.В.Канторович начал систематическое исследование задач линейного программирования и разработку общих методов их решения.

В 1941 году Хичкок поставил транспортную задачу. Основной метод решения задач линейного программирования — симплексный метод — был опубликован в 1949 году Д.Данцигом. В том же году Л.В.Канторович совместно с М.К.Гавуриным разработал метод потенциалов, который применяется при решении транспортных задач.

А в 1951 году опубликована работа Куна и Таккера, в которой приведены необходимые и достаточные условия оптимальности для решения нелинейных задач. Кроме этого, большой вклад в развитие математического программирования внесли также Б.Эгервари, В.С.Немчинов, А.Л.Лурье, Д.Б.Юдин, Р.Беллман, Л.Форд, С.Гасс и другие.

Предметом математического программирования являются математические модели, описывающие экономические процессы, связанные с предприятием, фирмой, рынком, производственным объединением, отраслями народного хозяйства, всем народным хозяйством, а также методы оптимального решения связанных с ними задач.

Математические модели используются в экономике с давних времен. Например, одной из первых моделей, примененных в экономике, была предложенная в 1758 году Ф.Кене модель воспроизводства.

Математической моделью экономической задачи называется описание с помощью математических формул основных условий и целей этой задачи.

Прежде чем решить любую экономическую задачу с применением методов математического программирования, следует построить ее математическую модель; другими словами, нужно выразить через математические формулы граничные условия и цели данной экономической задачи. Для построения математической модели любой задачи следует:

1) изучив экономический смысл задачи, определить ее основные условия и цель;
2) обозначить неизвестные в задаче;
3) выразить через алгебраические уравнения или неравенства условия задачи;
4) выразить через функцию цель задачи.

В общем виде математическая модель задачи математического программирования имеет следующий вид:

найти экстремум функции $f(x_1, x_2, \ldots, x_n)$, удовлетворяющей условиям

$$g_i(x_1, x_2, \ldots, x_n) \le b_i \qquad (i = 1, 2, \ldots, m).$$

Здесь: f, g_i — заданные функции, b_i — произвольные положительные числа.

Если все функции f и g_i являются линейными, то данная задача является задачей линейного программирования.

Если какая-нибудь из функций f и g_i нелинейная, то данная модель выражает задачу нелинейного программирования.

Если же функции f и g_i зависят от времени и решение задачи рассматривается как многоэтапный процесс, то данная модель выражает задачу динамического программирования.

Теперь в качестве примера приведем процесс построения математической модели нескольких простейших экономических задач.

Предположим, для изготовления двух видов продукции P_1 и P_2 используется три вида сырья: S_1, S_2 и S_3. Запасы сырья, количество единиц сырья, затрачиваемых на изготовление единицы продукции, а также величина прибыли, получаемая от реализации единицы продукции, приведены в таблице 1.1.

Таблица 1.1

Вид сырья	Запас сырья	Количество единиц сырья, идущих на изготовление единицы продукции	
		P_1	P_2
S_1	20	2	5
S_2	40	8	5
S_3	30	5	6
Прибыль от единицы продукции, сум		500	400

Необходимо составить такой план выпуска продукции, чтобы при ее реализации получить максимальную прибыль.

Обозначим через x_1 количество единиц продукции P_1, а через x_2 — количество единиц продукции P_2. Тогда, учитывая количество единиц сырья, затрачиваемых на изготовление единицы продукции, а также запасы сырья, получим систему ограничений

$$\begin{cases} 2x_1 + 5x_2 \le 20 \\ 8x_1 + 5x_2 \le 40, \\ 5x_1 + 6x_2 \le 30 \end{cases} \tag{1.1}$$

которая показывает, что количество сырья, расходуемое на изготовление продукции, не может превысить имеющихся запасов.

Если продукция P_1 не выпускается, то $x_1 = 0$; в противном случае $x_1 > 0$. То же самое получаем и для продукции P_2. Таким образом, на неизвестные x_1 и x_2 должно быть наложено ограничение неотрицательности: $x_1 \geq 0$, $x_2 \geq 0$.

Конечную цель решаемой задачи — получение максимальной прибыли при реализации продукции — выразим как функцию двух переменных x_1 и x_2. Реализация x_1 единиц продукции P_1 и x_2 единиц продукции P_2 дает соответственно $500\,x_1$ и $400\,x_2$ сум прибыли, суммарная прибыль равна $Z = 500\,x_1 + 400\,x_2$ сум.

Значит, необходимо найти такие неотрицательные значения x_1 и x_2, при которых функция Z достигает максимума, т.е. найти максимальное значение линейной функции $Z = 500\,x_1 + 400\,x_2$ при ограничениях

$$\begin{cases} 2x_1 + 5x_2 \leq 20 \\ 8x_1 + 5x_2 \leq 40, \; x_1 \geq 0, \; x_2 \geq 0. \\ 5x_1 + 6x_2 \leq 30 \end{cases} \tag{1.2}$$

Построенная линейная функция называется *целевой функцией* и совместно с системой ограничений образует математическую модель рассматриваемой экономической задачи.

Теперь рассмотрим следующую задачу. При откорме каждое животное ежедневно должно получить не менее 9 единиц питательного вещества S_1, не менее 8 единиц питательного вещества S_2 и не менее 12 единиц питательного вещества S_3. Для составления рациона используется два вида корма. Содержание количества единиц питательных веществ в 1 кг каждого вида корма и стоимость 1 кг корма приведены в таблице 1.2.

Таблица 1.2

Питательные вещества	Количество единиц питательных веществ в 1 кг корма	
	корм I	корм II
S_1	3	1
S_2	1	2
S_3	1	6
Стоимость 1 кг корма, сум	400	600

Необходимо составить дневной рацион нужной питательности, причем затраты на него должны быть минимальными.

Для составления математической модели обозначим через x_1 и x_2 соответственно количество килограммов корма I и II в дневном рационе. Если в качестве целевой функции взять линейную функцию $Z = 400\,x_1 + 600\,x_2$, означающую общую стоимость дневного рациона, то для решения поставленной задачи нужно найти минимальное значение линейной функции $Z = 400\,x_1 + 600\,x_2$ при ограничениях

$$\begin{cases} 3x_1 + x_2 \geq 9 \\ x_1 + 2x_2 \geq 8 \\ x_1 + 6x_2 \geq 12 \end{cases}, \; x_1 \geq 0, \; x_2 \geq 0. \tag{1.3}$$

В общем случае решение системы линейных неравенств с n неизвестными приходиться сталкиваться с большими трудностями, поэтому эта система неравенств сводится к системе уравнений путем ввода дополнительных переменных.

Вопросы для повторения и контроля :

1. Что является предметом математического программирования и каковы его составные части?
2. Что вы знаете о краткой истории развития математического программирования?
3. Что такое математическая модель экономической задачи и из каких этапов состоит ее построение?
4. Что вы знаете о математической модели задачи математического программирования в общем случае?
5. Что вы знаете о задаче использования сырья?
6. Что вы знаете о задаче потребительской корзины?

Опорные слова :

Математическое программирование, линейное программирование, нелинейное программирование, динамическое программирование, математическая модель экономической задачи, целевая функция.

Тема № 2

Формулировка и выражение в различных формах задачи линейного программирования

План :

1. Формулировка задачи линейного программирования.
2. Преобразование линейного неравенства с n неизвестными в линейное уравнение с $n + 1$ неизвестными.
3. Равносильные замены в задаче линейного программирования.
4. Канонический вид и выражение в различных формах задачи линейного программирования.

В общем виде задача линейного программирования выражается следующим образом:

Даны линейная функция

$$Z = c_1x_1 + c_2x_2 + \ldots + c_nx_n \tag{2.1}$$

и линейные ограничения

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n \le (\ge)\, b_1 \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n \le (\ge)\, b_2 \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n \le (\ge)\, b_m \end{cases}, \tag{2.2}$$

$$x_1 \ge 0,\ x_2 \ge 0,\ \ldots,\ x_n \ge 0. \tag{2.3}$$

Найти такие значения неизвестных, удовлетворяющих условиям (2.2) и (2.3), которые доставляют линейной функции (2.1) максимальное (минимальное) значение.

Условия (2.2) и (2.3) задачи называются ее *граничными условиями*, а линейная функция (2.1) — *целью* или *целевой функцией* задачи. Так как все граничные условия и целевая функция этой задачи являются линейными, задача (2.1) – (2.3) называется *задачей линейного программирования*.

В конкретных задачах условие (2.2) может быть системой уравнений, системой неравенств вида «≤» или «≥», или же смешанной системой. Однако, можно показать, что задачу вида (2.1) – (2.3) можно привести к виду

$$Z = c_1x_1 + c_2x_2 + \ldots + c_nx_n, \tag{2.4}$$

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n = b_2 \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n = b_m \end{cases}, \qquad (2.5)$$

$$x_1 \ge 0,\ x_2 \ge 0,\ \ldots,\ x_n \ge 0. \qquad (2.6)$$

Действительно, рассмотрим линейное неравенство с n неизвестными

$$a_1x_1 + a_2x_2 + \ldots + a_nx_n \le b. \qquad (2.7)$$

Для преобразования этого неравенства в уравнение прибавим к его меньшой стороне неизвестное число $x_{n+1} \ge 0$. В результате получим линейное уравнение с $n + 1$ неизвестными

$$a_1x_1 + a_2x_2 + \ldots + a_nx_n + x_{n+1} = b. \qquad (2.8)$$

Переменная $x_{n+1} \ge 0$, с помощью которой неравенство (2.7) преобразуется в уравнение (2.8), называется *дополнительной переменной*. В следующей теореме показано, что неравенство (2.7) и уравнение (2.8) имеют одинаковые решения.

Теорема 2.1. *Каждому решению* $\overline{\mathbf{X}} = (\beta_1, \beta_2, \ldots, \beta_n)$ *неравенства (2.7) соответствует единственное решение* $\overline{\mathbf{Y}} = (\beta_1, \beta_2, \ldots, \beta_n, \beta_{n+1})$ *уравнения (2.8), и, наоборот, каждому решению* $\overline{\mathbf{Y}}$ *уравнения (2.8) соответствует единственное решение* $\overline{\mathbf{X}}$ *неравенства (2.7).*

Доказательство. Пусть $\overline{\mathbf{X}}$ — решение неравенства (2.7). Тогда имеет место соотношение

$$a_1\beta_1 + a_2\beta_2 + \ldots + a_n\beta_n \le b.$$

Перенесем левую часть этого неравенства в правую и обозначим образовавшееся выражение в правой части через β_{n+1}:

$$0 \le b - (a_1\beta_1 + a_2\beta_2 + \ldots + a_n\beta_n) = \beta_{n+1}.$$

Отсюда следует, что решение $\overline{\mathbf{Y}} = (\beta_1, \beta_2, \ldots, \beta_n, \beta_{n+1})$ действительно является решением уравнения (2.8):

$$a_1\beta_1 + a_2\beta_2 + \ldots + a_n\beta_n + \beta_{n+1} =$$

$$= a_1\beta_1 + a_2\beta_2 + \ldots + a_n\beta_n + b - (a_1\beta_1 + a_2\beta_2 + \ldots + a_n\beta_n) = b.$$

Наоборот, пусть $\overline{\mathbf{Y}}$ удовлетворяет уравнению (2.8), т.е. пусть имеют место соотношения

$$a_1\beta_1 + a_2\beta_2 + \ldots + a_n\beta_n + \beta_{n+1} = b, \quad \beta_{n+1} \ge 0.$$

Тогда, отбрасывая в левой части этого равенства число $\beta_{n+1} \ge 0$, получаем неравенство

$$a_1\beta_1 + a_2\beta_2 + \ldots + a_n\beta_n \le b.$$

Отсюда следует, что $\overline{\mathbf{X}} = (\beta_1, \beta_2, \ldots, \beta_n)$ — решение неравенства (2.7).

Теперь рассмотрим линейное неравенство с n неизвестными

$$a_1x_1 + a_2x_2 + \ldots + a_nx_n \ge b. \qquad (2.9)$$

Для преобразования этого неравенства в уравнение отнимем от его большой стороны дополнительную переменную $x_{n+1} \ge 0$. В результате получим линейное уравнение с n + 1 неизвестными

$$a_1x_1 + a_2x_2 + \ldots + a_nx_n - x_{n+1} = b. \qquad (2.10)$$

Таким образом, любые неравенства в системе ограничений задачи линейного программирования можно преобразовать в уравнения. То есть, если система ограничений задачи линейного программирования содержит неравенства, то, вводя в каждое из них свою неотрицательную дополнительную переменную, ее можно преобразовать в систему уравнений. При этом в линейную функцию каждая дополнительная переменная входит с коэффициентом, равным нулю.

Например, пусть задача линейного программирования задана в виде

$$Z = c_1x_1 + c_2x_2 + \ldots + c_nx_n, \qquad (2.11)$$

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n \le b_1 \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n \le b_2 \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n \le b_m \end{cases}, \qquad (2.12)$$

$$x_1 \ge 0,\ x_2 \ge 0,\ \ldots,\ x_n \ge 0. \qquad (2.13)$$

Вышеупомянутым способом эту задачу можно привести к схожему с задачей типа (2.4) – (2.6) виду

$$Z = c_1x_1 + c_2x_2 + \ldots + c_nx_n + 0\cdot(x_{n+1} + \ldots + x_{n+m}), \qquad (2.14)$$

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n + x_{n+1} = b_1 \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n + x_{n+2} = b_2 \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n + x_{n+m} = b_m \end{cases}, \qquad (2.15)$$

$$x_1 \geq 0,\ x_2 \geq 0,\ \ldots,\ x_n \geq 0,\ x_{n+1} \geq 0,\ \ldots,\ x_{n+m} \geq 0. \qquad (2.16)$$

Теперь, пусть задача линейного программирования задана в другом виде:

$$Z = c_1x_1 + c_2x_2 + \ldots + c_nx_n, \qquad (2.17)$$

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n \geq b_1 \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n \geq b_2 \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n \geq b_m \end{cases}, \qquad (2.18)$$

$$x_1 \geq 0,\ x_2 \geq 0,\ \ldots,\ x_n \geq 0. \qquad (2.19)$$

В этом случае данную задачу также можно привести к схожему с задачей типа (2.4) – (2.6) виду

$$Z = c_1x_1 + c_2x_2 + \ldots + c_nx_n + 0 \cdot (x_{n+1} + \ldots + x_{n+m}), \qquad (2.20)$$

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n - x_{n+1} = b_1 \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n - x_{n+2} = b_2 \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n - x_{n+m} = b_m \end{cases}, \qquad (2.21)$$

$$x_1 \geq 0,\ x_2 \geq 0,\ \ldots,\ x_n \geq 0,\ x_{n+1} \geq 0,\ \ldots,\ x_{n+m} \geq 0. \qquad (2.22)$$

Следует также заметить, что правые части неравенств и уравнений любой системы ограничений можно считать неотрицательными, т.е. $b_i \geq 0 \quad (i = 1,2,\ldots,m)$. Если для какого-нибудь k-го неравенства или уравнения $b_k < 0$, то это неравенство или уравнение умножается на –1, причем знак неравенства меняется на противоположный.

Если для какой-либо переменной x_j не задано условие неотрицательности ($x_j \geq 0$), т.е. на знак этой переменной не наложено никакого ограничения, то эту переменную можно представить в виде разности двух неотрицательных переменных:

$$x_j = x_k - x_l, \qquad x_k \geq 0,\ x_l \geq 0.$$

Затем эту разность нужно подставить вместо переменной x_j во всех ограничениях и в целевой функции.

С другой стороны, если для какой-либо переменной x_j вместо условия $x_j \geq 0$ задано условие $x_j \geq b_j$, то от этой переменной к другой неотрицательной переменной x_k переходят с помощью формул

$$x_j = b_j + x_k \text{ и } x_k = x_j - b_j \geq 0.$$

Для преобразования проблемы нахождения максимального значения целевой функции задачи линейного программирования в проблему нахождения минимального значения этой функции целевую функцию следует взять с противоположным знаком и наоборот.

Вид (2.4) – (2.6) называется *каноническим видом задачи линейного программирования*. Задачу (2.4) – (2.6) в векторной форме можно выразить в виде

$$Z = \mathbf{CX}, \tag{2.23}$$

$$\mathbf{A}_1 x_1 + \mathbf{A}_2 x_2 + \ldots + \mathbf{A}_n x_n = \mathbf{A}_0, \tag{2.24}$$

$$\mathbf{X} \geq 0, \tag{2.25}$$

здесь $\mathbf{X} = (x_1, x_2, \ldots, x_n)$; $\mathbf{C} = (c_1, c_2, \ldots, c_n)$; $\mathbf{CX}$ — скалярное произведение;

$$\mathbf{A}_1 = \begin{pmatrix} a_{11} \\ a_{21} \\ \vdots \\ a_{m1} \end{pmatrix}, \ \mathbf{A}_2 = \begin{pmatrix} a_{12} \\ a_{22} \\ \vdots \\ a_{m2} \end{pmatrix}, \ \ldots, \ \mathbf{A}_n = \begin{pmatrix} a_{1n} \\ a_{2n} \\ \vdots \\ a_{mn} \end{pmatrix}, \ \mathbf{A}_0 = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{pmatrix}.$$

Найти такой неизвестный вектор, удовлетворяющий условиям (2.24) и (2.25), который доставляет линейной функции (2.23) максимальное (минимальное) значение.

Матричная форма записи задачи (2.4) – (2.6) выражается в виде

$$Z = \mathbf{CX}, \tag{2.26}$$

$$\mathbf{AX} = \mathbf{A}_0, \tag{2.27}$$

$$\mathbf{X} \geq 0 \tag{2.28}$$

здесь $\mathbf{C} = (c_1, c_2, \ldots, c_n)$ — матрица-строка; $\mathbf{A} = (a_{ij})$ — матрица системы;

$$\mathbf{X} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}, \quad \mathbf{A}_0 = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{pmatrix}$$ — матрицы-столбцы.

Задачу (2.4) – (2.6) можно выразить также с помощью знаков суммирования:

$$Z = \sum_{j=1}^{n} c_j x_j \,, \tag{2.29}$$

$$\sum_{j=1}^{n} a_{ij} x_j = b_i, \quad (i = 1,2,\ldots,m), \tag{2.30}$$

$$x_j \geq 0, \quad (j = 1,2,\ldots,n). \tag{2.31}$$

Вопросы для повторения и контроля :

1. Как выражается в общем виде задача линейного программирования и какие ее составные части вы знаете?
2. Как преобразовывается линейное неравенство с n неизвестными в линейное уравнение и что вы знаете о теореме 2.1?
3. Что такое канонический вид задачи линейного программирования и каким образом задача линейного программирования с системой ограничений, состоящей из неравенств разных типов, приводится к каноническому виду?
4. Как можно выразить задачу линейного программирования в векторной и матричной формах, а также с помощью знаков суммирования?

Опорные слова :

Задача линейного программирования, граничные условия задачи линейного программирования, целевая функция задачи линейного программирования, линейное неравенство с n неизвестными, линейное уравнение с n неизвестными, дополнительная переменная, канонический вид задачи линейного программирования, векторная форма записи задачи линейного программирования, матричная форма записи задачи линейного программирования, выражение задачи линейного программирования с помощью знаков суммирования.

Тема № 3

Решения задачи линейного программирования и их свойства. Геометрическая интерпретация и графический метод решения задачи линейного программирования

План :

1. Допустимое решение задачи линейного программирования. Опорное решение. Оптимальное решение.
2. Выпуклые множества.
3. Свойства решений задачи линейного программирования.
4. Геометрическая интерпретация и графический метод решения задачи линейного программирования.

Допустимым решением или *планом* данной задачи (2.4) – (2.6) называется вектор $\mathbf{X}=(x_1, x_2, \ldots, x_n)$, удовлетворяющий ее условиям (2.5) и (2.6).

Решение $\mathbf{X}=(x_1, x_2, \ldots, x_n)$ называется *опорным решением* или *планом*, если векторы $\mathbf{A}_j$ $(j=1,2,\ldots,n)$, входящие в разложение (2.24) с положительными коэффициентами x_j, являются линейно независимыми.

Если опорное решение $\mathbf{X}=(x_1, x_2, \ldots, x_n)$ содержит m положительных компонент, то это решение называется *невырожденным опорным решением* или *планом*, в противном случае —*вырожденным опорным решением* или *планом*.

Оптимальным решением или *планом* задачи называется опорное решение $\mathbf{X}=(x_1, x_2, \ldots, x_n)$, доставляющее максимальное (минимальное) значение линейной функции (2.4).

Выпуклой линейной комбинацией векторов $\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_m$ называется вектор

$$\mathbf{A}=\lambda_1\mathbf{A}_1+\lambda_2\mathbf{A}_2+\ldots+\lambda_m\mathbf{A}_m,$$

для которого выполняются условия

$$\lambda_i \geq 0 \quad (i=1,2,\ldots,m), \qquad \sum_{i=1}^{m}\lambda_i=1.$$

В n-мерном пространстве каждому вектору $\mathbf{A}=(a_1,a_2,\ldots,a_n)$ соответствует точка с координатами $(a_1,a_2,\ldots,a_n)$. Поэтому в дальнейшем вектор $\mathbf{A}=(a_1,a_2,\ldots,a_n)$ будем рассматривать как точку в n-мерном пространстве.

Например, *выпуклой линейной комбинацией точек* $\mathbf{A}_1$ и $\mathbf{A}_2$ называется точка $\mathbf{A}$, для которой выполняются условия

$$\lambda_1 \geq 0,\quad \lambda_2 \geq 0,\quad \lambda_1+\lambda_2=1, \tag{3.1}$$

$$\mathbf{A}=\lambda_1\mathbf{A}_1+\lambda_2\mathbf{A}_2. \tag{3.2}$$

Из условия (3.1) вытекает, что и λ_1, и λ_2 лежат между нулем и единицей. При $\lambda_1=1$ и $\lambda_2=0$ точка $\mathbf{A}$ совпадает с точкой $\mathbf{A}_1$, а при $\lambda_1=0$ и $\lambda_2=1$ — с точкой $\mathbf{A}_2$. Значит, если λ_1 (или λ_2) пробегает все значения от 0 до 1, то точка $\mathbf{A}$ описывает отрезок $\overline{\mathbf{A}_1\mathbf{A}_2}$.

Точки $\mathbf{A}_1$ и $\mathbf{A}_2$ называются *угловыми* или *крайними точками отрезка* $\overline{\mathbf{A}_1\mathbf{A}_2}$. Из определения линейной выпуклой комбинации точек очевидно, что угловая точка не может быть представлена как выпуклая линейная комбинация двух других точек отрезка.

Множество C n-мерного пространства называется *выпуклым множеством*, если оно вместе с любыми двумя своими точками $\mathbf{A}_1$ и $\mathbf{A}_2$ содержит и точку $\mathbf{A}$ (т.е. $\mathbf{A}_1 \in C,\ \mathbf{A}_2 \in C \Rightarrow \mathbf{A} \in C$), являющуюся произвольной выпуклой линейной комбинацией этих точек, для которой выполняются условия (3.1) и (3.2).

Геометрический смысл этого определения состоит в том, что множеству вместе с его двумя произвольными точками полностью принадлежит и прямолинейный отрезок, их соединяющий. Примерами выпуклых множеств служат прямолинейный отрезок, прямая, полуплоскость, круг, шар, куб, полупространство и др.

Угловыми или *крайними точками выпуклого множества* называются точки, не являющиеся выпуклой линейной комбинацией никаких двух различных точек этого множества. Например, угловыми точками треугольника являются его вершины, угловыми точками круга — точки окружности, которая его ограничивает. Таким образом, выпуклое множество может иметь конечное и бесконечное число угловых точек. Прямая, плоскость, полуплоскость, пространство и полупространство угловых точек не имеют.

Теперь приведем свойства решений задачи линейного программирования.

Теорема 3.1. *Множество всех решений задачи линейного программирования является выпуклым.*

Доказательство. Покажем, что если $\mathbf{X}_1$ и $\mathbf{X}_2$ являются решениями задачи линейного программирования (2.4) – (2.6), то их любая выпуклая линейная комбинация $\mathbf{X}$, для которой выполняются условия (3.1) и (3.2), также является решением задачи (2.4) – (2.6). Для этого воспользуемся выражением в виде (2.26) – (2.28) задачи (2.4) – (2.6).

Так как $\mathbf{X}_1$ и $\mathbf{X}_2$ — решения задачи, то выполняются соотношения

$$\mathbf{AX}_1 = \mathbf{A}_0,\ \mathbf{X}_1 \geq 0,\ \mathbf{AX}_2 = \mathbf{A}_0,\ \mathbf{X}_2 \geq 0.$$

Отсюда, учитывая соотношения (3.1) и (3.2), имеем:

$$\mathbf{AX} = \mathbf{A}(\lambda_1\mathbf{X}_1 + \lambda_2\mathbf{X}_2) = \lambda_1\mathbf{AX}_1 + \lambda_2\mathbf{AX}_2 = \lambda_1\mathbf{A}_0 + \lambda_2\mathbf{A}_0 =$$
$$= (\lambda_1 + \lambda_2)\mathbf{A}_0 = \mathbf{A}_0.$$

Значит, $\mathbf{X}$ удовлетворяет систему (2.27). С другой стороны, так как $\mathbf{X}_1 \geq 0$, $\mathbf{X}_2 \geq 0$, $\lambda_1 \geq 0$ и $\lambda_2 \geq 0$, то и $\mathbf{X} \geq 0$, т.е. $\mathbf{X}$ удовлетворяет и условию (2.28).

Таким образом, $\mathbf{X}$ — решение задачи линейного программирования.

Теорема 3.2. *Линейная функция задачи линейного программирования достигает своего экстремального значения в угловой точке выпуклого множества всех решений этой задачи.*

Если линейная функция принимает экстремальное значение более чем в одной угловой точке этого выпуклого множества, то она достигает того же значения в любой точке, являющейся выпуклой линейной комбинацией этих точек.

Теорема 3.3. *Если вектора* $\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_k$ $(k \leq n)$ *в разложении (2.24) линейно независимы и для них выполняется соотношение*

$$\mathbf{A}_1 x_1 + \mathbf{A}_2 x_2 + \ldots + \mathbf{A}_k x_k = \mathbf{A}_0,$$

где $x_j \geq 0$ $(j = 1,2,\ldots,k)$, *то точка* $\mathbf{X} = (x_1, x_2, \ldots, x_k, 0, \ldots, 0)$ *является угловой точкой выпуклого множества всех решений задачи линейного программирования.*

Теорема 3.4. *Если* $\mathbf{X} = (x_1, x_2, \ldots, x_n)$ — *угловая точка выпуклого множества всех решений задачи линейного программирования, то векторы в разложении (2.24), соответствующие положительным* x_j, *являются линейно независимыми.*

Эти теоремы приводятся без доказательства. Из них вытекают следующие следствия.

Следствие 3.1. *Каждой угловой точке выпуклого множества всех решений задачи линейного программирования соответствует* $k \le m$ *линейно независимых векторов системы* $\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_n$.

Следствие 3.2. *Каждому опорному решению задачи линейного программирования соответствует угловая точка выпуклого множества всех решений этой задачи.*

Следствие 3.3. *Для решения задачи линейного программирования необходимо исследовать только угловые точки выпуклого множества всех решений этой задачи, т.е. только опорные решения.*

Теперь остановимся на геометрической интерпретации и графическом методе решения задачи линейного программирования. Пусть задача линейного программирования задана в виде (2.11) – (2.13). Если эта задача имеет, по крайней мере, одно решение, то система неравенств (2.12) будет совместной, в противном случае — несовместной.

Для наглядности задачу линейного программирования рассмотрим в двумерном пространстве, т.е. на плоскости, здесь количество неизвестных равно двум, т.е. $n = 2$. Рассмотрим на плоскости следующую совместную систему линейных неравенств

$$\begin{cases} a_{11}x_1 + a_{12}x_2 \le b_1 \\ a_{21}x_1 + a_{22}x_2 \le b_2 \\ \cdots\cdots\cdots\cdots\cdots\cdots \\ a_{m1}x_1 + a_{m2}x_2 \le b_m \end{cases}, \quad x_1 \ge 0,\ x_2 \ge 0.$$

Каждое неравенство этой системы в геометрическом смысле выражает полуплоскость с граничной прямой $a_{i1}x_1 + a_{i2}x_2 = b_i$ $(i = 1,2,\ldots,m)$. Условия неотрицательности выражают полуплоскости с граничными прямыми $x_1 = 0$ и $x_2 = 0$. Пересечение этих полуплоскостей является выпуклым множеством и называется *многоугольником решений* (рис. 3.1). Многоугольник решений может быть и неограниченной областью.

Если в системе ограничений (2.12) и (2.13) $n = 3$, то каждое неравенство системы в геометрическом смысле выражает полупространство с граничной плоскостью $a_{i1}x_1 + a_{i2}x_2 + a_{i3}x_3 = b_i$ $(i = 1,2,\ldots,m)$, а условия неотрицательности выражают полупространства с граничными плоскостями $x_1 = 0$, $x_2 = 0$ и $x_3 = 0$. Пересечение этих полупространств является выпуклым множеством и называется *многогранником решений*.

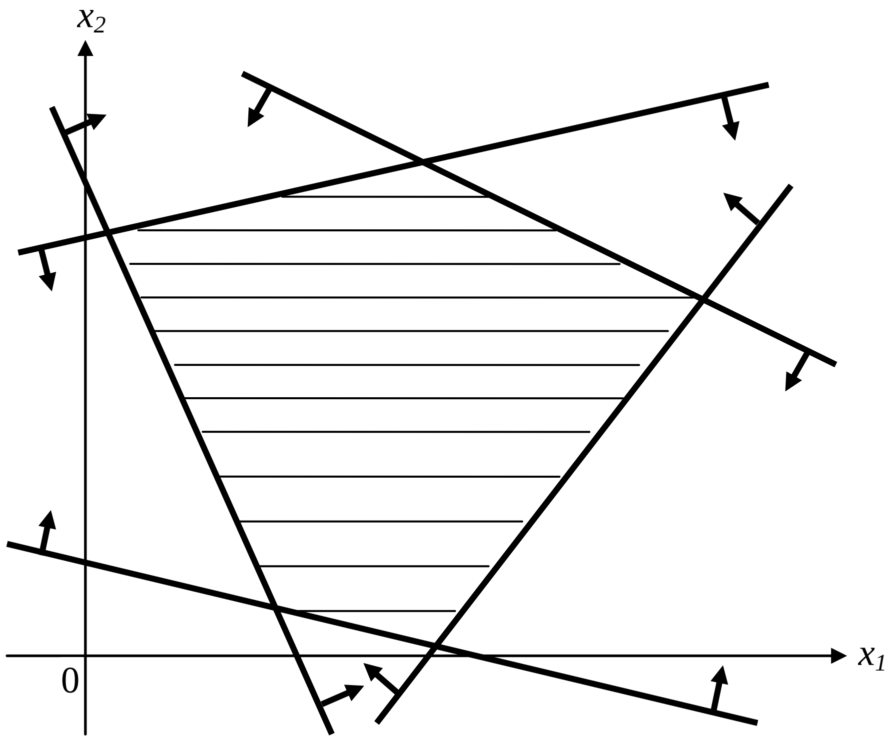

Рис. 3.1.

В общем случае, т.е. в n-мерном $(n > 3)$ пространстве каждое неравенство системы ограничений (2.12) и (2.13) в геометрическом смысле выражает полупространство с граничной гиперплоскостью $a_{i1}x_1 + a_{i2}x_2 + \ldots + a_{in}x_n = b_i$ $(i = 1,2,\ldots,m)$, а условия неотрицательности выражают полупространства с граничными гиперплоскостями $x_j = 0$ $(j = 1,2,\ldots,n)$. Пересечение этих полупространств также является выпуклым множеством и так же, как в трехмерном пространстве, называется *многогранником решений*.

Графический метод решения задачи линейного программирования применяется в основном при решении задач двумерного пространства и только некоторых задач трехмерного пространства. Задачу пространства размерности больше трех изобразить графически вообще невозможно.

Пусть задача линейного программирования задана в двумерном пространстве, т.е. ограничения содержат две переменные. Рассмотрим следующую задачу.

Найти максимальное (минимальное) значение функции

$$Z = c_1x_1 + c_2x_2 \tag{3.3}$$

при ограничениях

$$\begin{cases} a_{11}x_1 + a_{12}x_2 \le b_1 \\ a_{21}x_1 + a_{22}x_2 \le b_2 \\ \dots\dots\dots\dots\dots\dots \\ a_{m1}x_1 + a_{m2}x_2 \le b_m \end{cases}, \tag{3.4}$$

$$x_1 \ge 0,\ x_2 \ge 0. \tag{3.5}$$

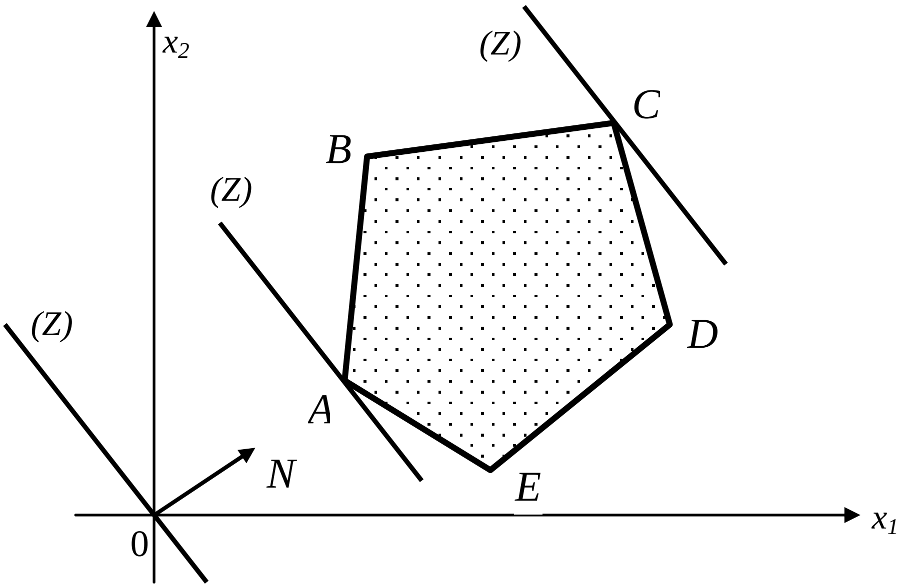

Рис. 3.2.

Допустим, что система (3.4) при условии (3.5) совместна, т.е. система (3.4) имеет решения, удовлетворяющие условию (3.5), и ее многоугольник решений ограничен. Линейная функция

(3.3) при фиксированных значениях Z является уравнением прямой линии $c_1x_1 + c_2x_2 = const$. Построим многоугольник решений системы ограничений (3.4) и график линейной функции (3.3) при $Z = 0$ (рис. 3.2). Тогда поставленной задаче линейного программирования можно дать следующую интерпретацию. Найти точку многоугольника решений, в которой прямая $c_1x_1 + c_2x_2 = const$ является касательной к этому многоугольнику, а функция Z достигает своего максимального (минимального) значения.

Значения функции $Z = c_1x_1 + c_2x_2$ возрастают в направлении вектора $\mathbf{N} = (c_1, c_2)$, поэтому прямую $Z = 0$ передвигаем параллельно самой себе в направлении вектора $\mathbf{N}$.

Из рис. 3.2 следует, что в точках A и C прямая $c_1x_1 + c_2x_2 = const$ является касательной к многоугольнику решений, причем функция Z минимальное значение принимает в точке A, а максимальное — в точке C. Координаты точки A определяются из уравнений, выражающих прямые AB и AE, а точки C — прямые CB и CD.

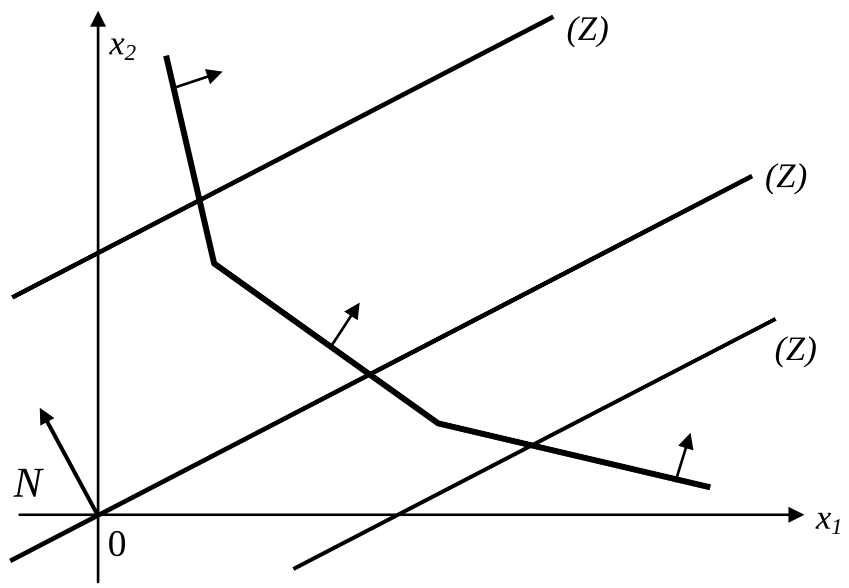

Рис. 3.3.

Если многоугольник решений не ограничен, то возможны два случая.

С л у ч а й 1. Прямая $c_1x_1 + c_2x_2 = const$, передвигаясь в направлении вектора $\mathbf{N}$ или противоположно ему, постоянно пересекает многоугольник решений и ни в какой точке не является касательной к нему. В этом случае линейная функция $Z = c_1x_1 + c_2x_2$ не ограничена на многоугольнике решений как сверху, так и снизу (рис. 3.3).

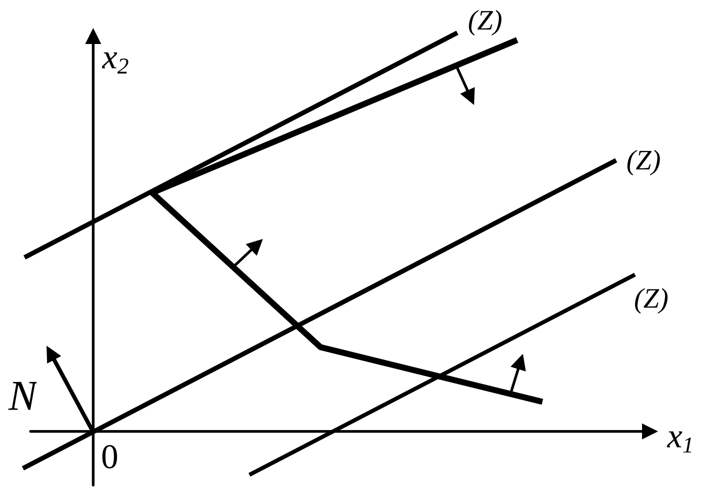

Рис. 3.4.

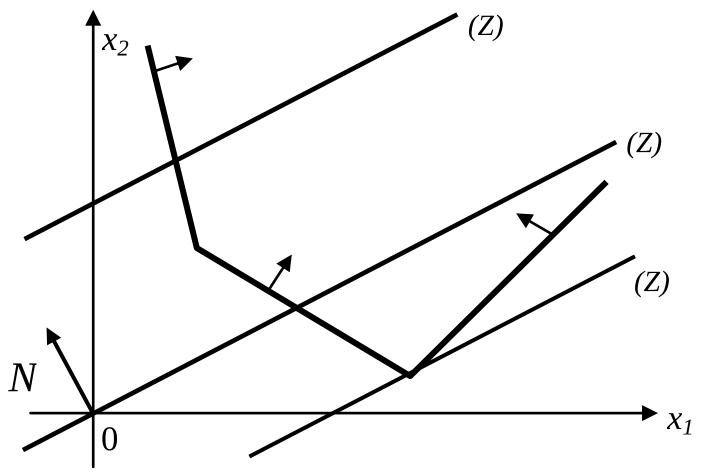

Рис. 3.5.

С л у ч а й 2. Прямая $c_1x_1 + c_2x_2 = const$, передвигаясь в направлении вектора **N** или противоположно ему, становится касательной к многоугольнику решений в его угловой точке. Тогда в зависимости от вида многоугольника решений линейная функция $Z = c_1x_1 + c_2x_2$ может быть ограниченной сверху, но неограниченной снизу (рис. 3.4), ограниченной снизу, но неограниченной сверху (рис. 3.5), ограниченной как снизу, так и сверху (рис. 3.6).

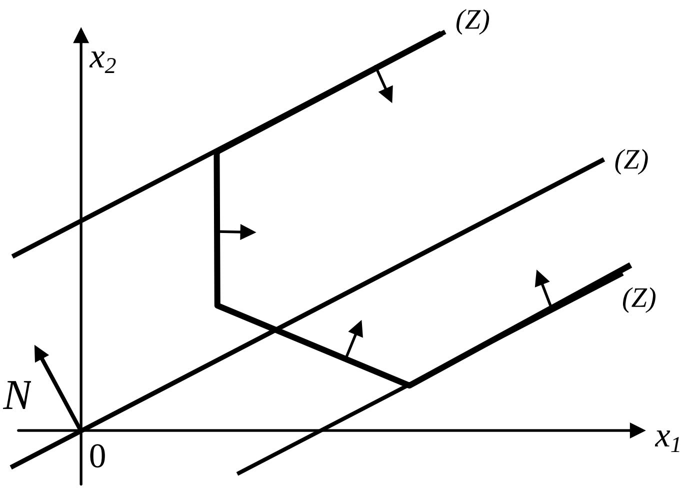

Рис. 3.6.

Теперь решим графическим методом задачу использования сырья, приведенную в теме № 1.

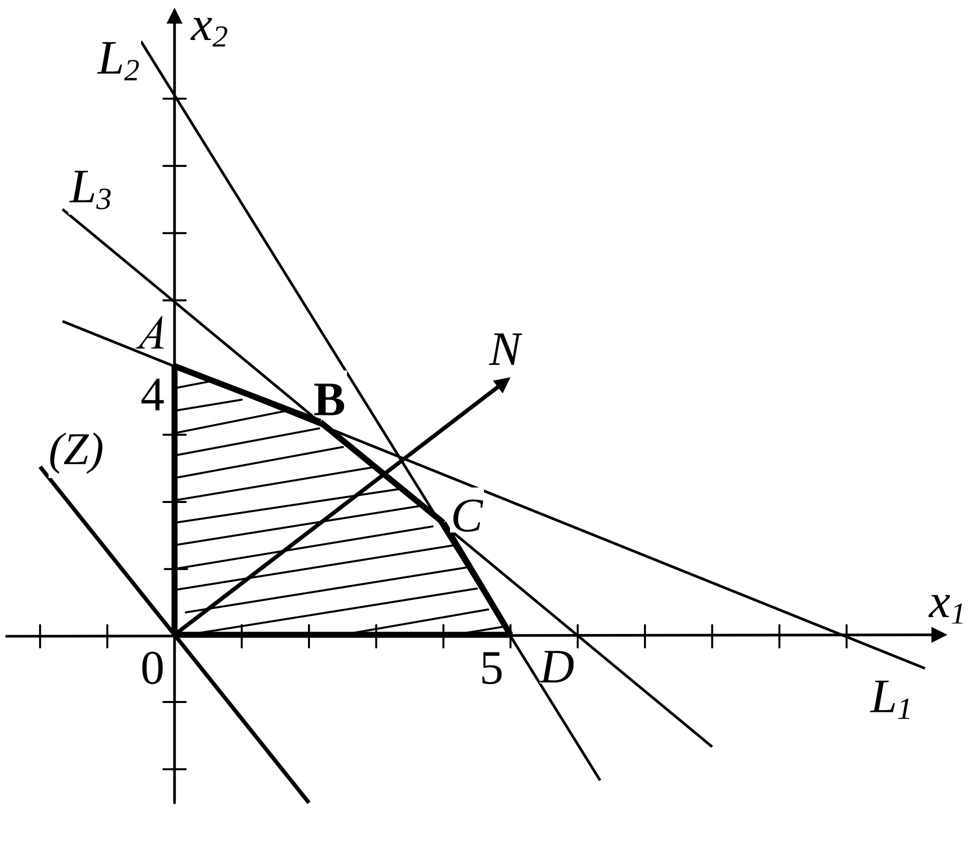

Рис. 3.7.

Пример 3.1. Найти максимальное значение линейной функции $Z = 500\,x_1 + 400\,x_2$ при огранчениях

$$\begin{cases} 2x_1 + 5x_2 \le 20 \\ 8x_1 + 5x_2 \le 40, \quad x_1 \ge 0, \quad x_2 \ge 0. \\ 5x_1 + 6x_2 \le 30 \end{cases}$$

Решение. Построим многоугольник решений (рис. 3.7). Для этого в системе координат x_1Ox_2 на плоскости изобразим граничные прямые

$$\begin{cases} 2x_1 + 5x_2 = 20 \quad (L_1) \\ 8x_1 + 5x_2 = 40 \quad (L_2), \quad x_1 = 0, \quad x_2 = 0 \\ 5x_1 + 6x_2 = 30 \quad (L_3) \end{cases}$$

Установим, какую полуплоскость определяет соответствующее неравенство. Многоугольником решений данной задачи является ограниченный пятиугольник *OABCD*.

Для построения прямой $500x_1 + 400x_2 = 0$ строим вектор $\mathbf{N} = (5;4) = 0{,}01 \cdot (500;400)$ и через точку *O* проводим прямую, перпендикулярную ему. Построенную прямую $Z = 0$ перемещаем параллельно самой себе в направлении вектора $\mathbf{N}$. Из рис. 3.7 следует, что функция *Z* достигает максимального значения на многоугольнике решений в точке *C*. Точка *C* лежит на пересечении прямых L_2 и L_3. Для определения ее координат решается система уравнений

$$\begin{cases} 8x_1 + 5x_2 = 40 \\ 5x_1 + 6x_2 = 30 \end{cases}.$$

Оптимальный план задачи: $x_1 = 90/23 \approx 3{,}91$; $x_2 = 40/23 \approx 1{,}74$. Подставляя значения x_1 и x_2 в линейную функцию, получаем $Z_{\max} = 500 \cdot 3{,}91 + 400 \cdot 1{,}74 \approx 2651.$

Таким образом, для того, чтобы получить максимальную прибыль в размере 2651 сум, необходимо запланировать производство 3,91 единиц продукции P_1 и 1,74 единиц продукции P_2.

Вопросы для повторения и контроля :

1. Что называется допустимым решением (планом), опорным решением, невырожденным опорным решением,

вырожденным опорным решением, оптимальным решением задачи линейного программирования?
2. Что такое выпуклая линейная комбинация векторов, точек?
3. Что называется угловыми или крайними точками отрезка?
4. Какое множество называется выпуклым множеством?
5. Что называется угловыми или крайними точками выпуклого множества?
6. Что вы знаете о теореме 3.1 и ее доказательстве?
7. О чем идет речь в теоремах 3.2 и 3.3?
8. О чем идет речь в теореме 3.4 и следствиях 3.1 – 3.3?
9. Что вы знаете о геометрической интерпретации задачи линейного программирования; что означают понятия многоугольника решений, многогранника решений, гиперплоскости?
10. Как решается задача линейного программирования графическим методом?
11. Какие случаи возможны при решении задачи линейного программирования графическим методом, если многоугольник решений не ограничен?

Опорные слова :

Допустимое решение (план) задачи линейного программирования, опорное решение (план) задачи линейного программирования, невырожденное опорное решение (план) задачи линейного программирования, вырожденное опорное решение (план) задачи линейного программирования, оптимальное решение (план) задачи линейного программирования, выпуклая линейная комбинация векторов, выпуклая линейная комбинация точек, угловые или крайние точки отрезка, выпуклое множество, угловые или крайние точки выпуклого множества, множество всех решений задачи линейного программирования, геометрическая интерпретация задачи линейного программирования, многоугольник решений, многогранник решений, гиперплоскость, графический метод решения задачи линейного программирования.

Тема № 4

Симплексный метод для решения задачи линейного программирования

План :

1. Нахождение базисного решения системы линейных уравнений методом Жордана–Гаусса.
2. Алгоритм симплексного метода для решения задачи линейного программирования. Критерий оптимальности.
3. Признак неограниченности целевой функции. Правила перехода к новому опорному решению.
4. Формулы преобразования при переходе к новой симплексной таблице. Условия единственности оптимального решения.

Пусть задана задача линейного программирования (2.4) – (2.6). Оптимальное решение этой задачи следует искать среди опорных решений. Каждое опорное решение определяется системой k линейно независимых векторов среди заданных n векторов $\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_n$. Один из способов нахождения такой системы векторов основывается на нахождении неотрицательного базисного решения системы линейных уравнений.

Пусть задана система линейных уравнений

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n = b_2 \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n = b_m \end{cases}. \qquad (4.1)$$

Решим эту систему методом Жордана–Гаусса.

Для этого в первом уравнении системы выбирается произвольное неизвестное с коэффициентом, отличным от нуля, и первое уравнение делится на этот коэффициент. С помощью первого уравнения выбранное неизвестное исключается из всех уравнений, кроме первого. Затем во втором уравнении выбирается произвольное неизвестное с коэффициентом, отличным от нуля, и второе уравнение делится на этот коэффициент. С помощью второго уравнения выбранное неизвестное исключается из всех уравнений, кроме второго. Этот процесс продолжается до тех пор, пока не будут использованы все уравнения системы.

В процессе последовательного исключения неизвестных возможны следующие случаи.

1. Возможно, что левая часть некоторого i-го уравнения обращается в нуль, а правая часть равна некоторому числу $b_i \neq 0$, т.е. это уравнение имеет вид $0 = b_i \neq 0$. В этом случае система не имеет решений, так как i-му уравнению не могут удовлетворить никакие значения неизвестных. Такое условие называется *условием отсутствия* или *критерием отсутствия* решений системы.

2. Возможно, что как левая, так и правая часть i-го уравнения обращаются в нуль, т.е. уравнение принимает вид $0 = 0$. Так как этому уравнению удовлетворяют любые значения неизвестных, его можно отбросить.

3. После того, как все уравнения системы использованы для исключения неизвестных, либо будет получено решение, либо доказано, что система несовместна.

После завершения процесса последовательного исключения неизвестных система линейных уравнений (4.1) примет вид

$$\begin{cases} x_1 + \qquad\qquad a'_{1m+1}x_{m+1} + \ldots + a'_{1n}x_n = b'_1 \\ \quad x_2 + \qquad\quad a'_{2m+1}x_{m+1} + \ldots + a'_{2n}x_n = b'_2 \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ \qquad\qquad x_m + a'_{mm+1}x_{m+1} + \ldots + a'_{mn}x_n = b'_m \end{cases}, \qquad (4.2)$$

здесь мы без ограничения общности выбрали в качестве исключенных неизвестных $x_1, x_2, \ldots, x_m$.

Неизвестные (переменные) $x_1, x_2, \ldots, x_m$ в системе (4.2) называются *выделенными* или *базисными* неизвестными (переменными), так как в векторном выражении

$$\mathbf{A}_1 x_1 + \mathbf{A}_2 x_2 + \ldots + \mathbf{A}_m x_m + \mathbf{A}_{m+1} x_{m+1} + \ldots + \mathbf{A}_n x_n = \mathbf{B}$$

системы (4.2) переменным $x_1, x_2, \ldots, x_m$ соответствуют базисные вектора

$$\mathbf{A}_1=\begin{pmatrix}1\\0\\\vdots\\0\end{pmatrix},\ \mathbf{A}_2=\begin{pmatrix}0\\1\\\vdots\\0\end{pmatrix},\ \ldots,\ \mathbf{A}_m=\begin{pmatrix}0\\0\\\vdots\\1\end{pmatrix}$$

m-мерного пространства, а неизвестные (переменные) $x_{m+1}, x_{m+2}, \ldots, x_n$ называются *невыделенными* или *свободными* неизвестными (переменными).

Используя систему (4.2), можно найти общее решение исходной системы (4.1). Для этого приведем систему (4.2) к виду

$$\begin{cases} x_1 = b_1' - a_{1m+1}' x_{m+1} - \ldots - a_{1n}' x_n \\ x_2 = b_2' - a_{2m+1}' x_{m+1} - \ldots - a_{2n}' x_n \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ x_m = b_m' - a_{mm+1}' x_{m+1} - \ldots - a_{mn}' x_n \end{cases}. \tag{4.3}$$

Придавая свободным переменным этой системы произвольные значения, можно найти соответствующие значения базисных переменных.

Решение $x_1 = b_1'$, $x_2 = b_2'$, ... , $x_m = b_m'$, $x_{m+1} = 0$, ... , $x_n = 0$, найденное путем приравнения нулю свободных переменных, называется *базисным решением* исходной системы. Каждое уравнение системы (4.3) называется *x уравнением*.

Пример 4.1. Решить методом Жордана–Гаусса систему уравнений

$$\begin{cases} 5x_1 + 2x_2 + 3x_3 + 3x_4 = 1 \\ 2x_1 - 2x_2 + 5x_3 + 2x_4 = 4 \\ 3x_1 + 4x_2 + 2x_3 + 2x_4 = -2 \end{cases}.$$

<u>Решение</u>. В первом уравнении коэффициент при x_1 отличен от нуля, т.е. равен 5. Разделим первое уравнение на 5. Исключим x_1 из остальных уравнений. Для этого умножим первое уравнение на 2 и вычтем из второго, затем умножим первое уравнение на 3 и вычтем из третьего. В результате получим

$$\begin{cases} x_1 + 2/5\,x_2 + 3/5\,x_3 + 3/5\,x_4 = 1/5 \\ \quad -14/5\,x_2 + 19/5\,x_3 + 4/5\,x_4 = 18/5 \\ \quad\quad 14/5\,x_2 + 1/5\,x_3 + 1/5\,x_4 = -13/5 \end{cases}.$$

Здесь не выполняется критерий отсутствия решений системы, поэтому можем продолжить процесс исключения неизвестных.

Выбрав во втором уравнении x_2, разделим это уравнение на коэффициент –14/5 при x_2 и исключим x_2 из остальных уравнений:

$$\begin{cases} x_1 + \qquad 8/7\,x_3 + 5/7\,x_4 = 5/7 \\ \qquad x_2 - 19/14\,x_3 - 2/7\,x_4 = -9/7 \\ \qquad\qquad 4x_3 + \quad x_4 = 1 \end{cases}.$$

Повторив этот процесс и для x_3, в итоге придем к системе

$$\begin{cases} x_1 + \qquad\qquad 3/7\,x_4 = 3/7 \\ \qquad x_2 + \quad 3/56\,x_4 = -53/56 \\ \qquad\qquad x_3 + 1/4\,x_4 = 1/4 \end{cases}$$

или

$$\begin{cases} x_1 = 3/7 - 3/7\,x_4 \\ x_2 = -53/56 - 3/56\,x_4 \\ x_3 = 1/4 - 1/4\,x_4 \end{cases}.$$

Значит, здесь переменные x_1, x_2, x_3 являются базисными переменными, а x_4 — свободной переменной. Базисное решение системы таково: $x_1 = 3/7$, $x_2 = -53/56$, $x_3 = 1/4$, $x_4 = 0$.

Как мы видели в предыдушей теме, оптимальное решение задачи линейного программирования следует искать среди угловых точек выпуклого множества всех решений этой задачи. Число таких точек, т.е. число опорных решений задачи ограничено сверху числом C_n^m сочетаний по m из n. Когда число неизвестных (n) и число уравнений (m) задачи велико, очень трудно проверить на оптимальность все опорные решения. Поэтому требуется задание схемы решения, которая упорядоченным образом проверяя опорные решения, выявляет среди них оптимальное решение.

Одним из самых универсальных методов, применяемых при решении задач линейного программирования, является симплексный метод. Этот метод указывает способ получения оптимального решения из первоначального опорного решения за конечное число итераций, и каждая последующая итерация дает более близкое к оптимальному решение, чем предыдущая. С помощью этого метода находится решение экономических задач или выявляется, что решение не существует.

Идея симплексного метода заключается в следующем. Находится любое первоначальное опорное решение данной задачи линейного программирования. Если это решение удовлетворяет условию оптимальности, оно является оптимальным решением, в противном случае первоначальное опорное решение заменяется более близким к оптимальному решению опорным решением. Процесс замены опорных решений продолжается до получения оптимального решения или установления отсутствия оптимального решения данной задачи. Симплексный метод в процессе решения позволяет выявить то, что задача не имеет решения или ее линейная функция не ограничена на многограннике решений.

Симплексный метод был предложен в 1949 году американским ученым Данцигом. Бывший советский ученый, академик Канторович, параллельно с Данцигом разработав метод «двойственных оценок», являющимся одним из видов симплексного метода, внес большой вклад в развитие науки линейного программирования. Впоследствии симплексный метод был развит различными учеными.

Симплексный метод Данцига основан на условии наличия в каждом уравнении по одному выделенному неизвестному (базисной переменной). Другими словами, считается, что в задаче линейного программирования существуют m линейно независимых векторов. Без ограничения общности будем считать, что эти вектора — первые m векторов $\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_m$. Тогда задача будет иметь следующий вид:

найти значения неизвестных, доставляющих минимальное значение функции

$$Z = c_1 x_1 + c_2 x_2 + \ldots + c_n x_n \tag{4.4}$$

и удовлетворяющих условиям

$$\begin{cases} x_1 + a_{1m+1} x_{m+1} + \ldots + a_{1n} x_n = b_1 \\ x_2 + a_{2m+1} x_{m+1} + \ldots + a_{2n} x_n = b_2 \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ x_m + a_{mm+1} x_{m+1} + \ldots + a_{mn} x_n = b_m \end{cases}, \tag{4.5}$$

$$x_1 \geq 0,\ x_2 \geq 0, \ldots,\ x_n \geq 0, \tag{4.6}$$
где $b_i \geq 0 \ \ (i = 1,2,\ldots,m)$.

Запишем систему линейных уравнений (4.5) в векторном виде

$$\mathbf{A}_1 x_1 + \mathbf{A}_2 x_2 + \ldots + \mathbf{A}_m x_m + \mathbf{A}_{m+1} x_{m+1} + \ldots + \mathbf{A}_n x_n = \mathbf{A}_0,$$

где

$$\mathbf{A}_1 = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \mathbf{A}_2 = \begin{pmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{pmatrix}, \ldots, \mathbf{A}_m = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{pmatrix},$$

$$\mathbf{A}_{m+1} = \begin{pmatrix} a_{1,m+1} \\ a_{2,m+1} \\ \vdots \\ a_{m,m+1} \end{pmatrix}, \ldots, \ \mathbf{A}_n = \begin{pmatrix} a_{1n} \\ a_{2n} \\ \vdots \\ a_{mn} \end{pmatrix}, \mathbf{A}_0 = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{pmatrix}.$$

Система векторов $\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_m$ является системой линейно независимых единичных векторов в m-мерном пространстве. Они образуют базис m-мерного пространства. Переменные $x_1, x_2, \ldots, x_m$, соответствующие этим векторам, будем рассматривать в качестве базисных переменных. Таким образом, если свободные переменные $x_{m+1}, x_{m+2}, \ldots, x_n$ приравнять нулю, то базисные переменные будут равны свободным членам. В результате получим первоначальное решение

$$\mathbf{X}_0 = (b_1, b_2, \ldots, b_m, 0, \ldots, 0).$$

Этому решению соответствует разложение

$$b_1 \mathbf{A}_1 + b_2 \mathbf{A}_2 + \ldots + b_m \mathbf{A}_m = \mathbf{A}_0.$$

Так как векторы $\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_m$ в этом разложении линейно независимы, то найденное первоначальное решение является и опорным.

Для удобства дальнейших вычислений условия задачи и первоначальные данные, полученные после определения первоначального опорного решения, записываются в следующую таблицу, которая называется *симплексной таблицей*:

Таблица 4.1

i	Базис	$\mathbf{C}_{баз}$	$\mathbf{A}_0$	c_1	c_2	$\ldots$	c_m	c_{m+1}	$\ldots$	c_k	$\ldots$	c_n
				$\mathbf{A}_1$	$\mathbf{A}_2$	$\ldots$	$\mathbf{A}_m$	$\mathbf{A}_{m+1}$	$\ldots$	$\mathbf{A}_k$	$\ldots$	$\mathbf{A}_n$
1	$\mathbf{A}_1$	c_1	b_1	1	0	$\ldots$	0	$a_{1,m+1}$	$\ldots$	a_{1k}	$\ldots$	a_{1n}
2	$\mathbf{A}_2$	c_2	b_2	0	1	$\ldots$	0	$a_{2,m+1}$	$\ldots$	a_{2k}	$\ldots$	a_{2n}
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
l	$\mathbf{A}_l$	c_l	b_l	0	0	$\ldots$	0	$a_{l,m+1}$	$\ldots$	a_{lk}	$\ldots$	a_{ln}
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
m	$\mathbf{A}_m$	c_m	b_m	0	0	$\ldots$	1	$a_{m,m+1}$	$\ldots$	a_{mk}	$\ldots$	a_{mn}
Δ_j			Z_0	0	0	$\ldots$	0	Δ_{m+1}	$\ldots$	Δ_k	$\ldots$	Δ_n

В столбце $\mathbf{C}_{баз}$ таблицы размещаются коэффициенты базисных переменных $x_1, x_2, \ldots, x_m$ в линейной функции (4.4). В столбец $\mathbf{A}_0$ записывается первоначальное опорное решение $\mathbf{X}_0$, в нем же в результате вычислений получаем оптимальное решение.

В таблице над каждым вектором $\mathbf{A}_j$ записан коэффициент c_j переменной x_j в линейной функции (4.4) ($j = 1,2,\ldots,n$).

Кроме того, в $m+1$ ой строке таблицы в столбце $\mathbf{A}_0$ записывается значение Z_0 линейной функции (4.4) при найденном опорном решении, а в столбцах $\mathbf{A}_j$ — величины $\Delta_j = Z_j - c_j$ ($j = 1,2,\ldots,n$), где

$$Z_0 = c_1 b_1 + c_2 b_2 + \ldots + c_m b_m, \tag{4.7}$$

$$Z_j = c_1 a_{1j} + c_2 a_{2j} + \ldots + c_m a_{mj} \qquad (j = 1,2,\ldots,n). \tag{4.8}$$

Для опорного решения проверим *критерий оптимальности*: **если все $\Delta_j \le 0$ ($j = 1,2,\ldots,n$), то опорное решение $\mathbf{X}_0$ будет оптимальным решением и минимальное значение линейной функции Z равно Z_0.**

Если хотя бы для одного j имеет место $\Delta_j = Z_j - c_j > 0$, то оптимальное решение задачи еще не найдено. Поэтому для замены найденного опорного решения более близким к оптимальному решению опорным решением в базис следует включить вектор $\mathbf{A}_k$, удовлетворяющий условию

$$\Delta_k = \max_{\Delta_j > 0} \Delta_j . \tag{4.9}$$

Однако, если при этом все координаты вектора $\mathbf{A}_k$ неположительны, те. $a_{ik} \le 0$ $(i = 1,2,\ldots,m)$, то *целевая функция задачи линейного программирования не имеет конечного минимума*.

При включении в базис вектора $\mathbf{A}_k$, из базиса следует исключить один из старых базисных векторов. Из базиса исключается вектор $\mathbf{A}_l$, удовлетворяющий условию

$$b_l / a_{lk} = \min_{a_{ik}>0} \left(b_i / a_{ik} \right). \tag{4.10}$$

Коэффициент a_{lk} отмечается как разрешающий элемент. Вместо вектора $\mathbf{A}_l$ из l-й строки, где расположен этот элемент, в базис включается вектор $\mathbf{A}_k$ из столбца, где расположен разрешающий элемент. Для этого симплексная таблица преобразуется на основании следующих формул:

$$\begin{cases} \begin{cases} b_i' = b_i - \left(b_l / a_{lk} \right) \cdot a_{ik} \\ a_{ij}' = a_{ij} - \left(a_{lj} / a_{lk} \right) \cdot a_{ik} \end{cases} & i \ne l \\ \begin{cases} b_l' = b_l / a_{lk} \\ a_{lj}' = a_{lj} / a_{lk} \end{cases} & \end{cases} . \tag{4.11}$$

Суть этих формул следующая. В строке, где заменяется базисный вектор (l-я строка), новые коэффициенты a и свободный член b получаются из прежних путем деления их на разрешающий элемент a_{lk}. А для остальных строк новые коэффициенты a и свободный член b образуются путем умножения новых коэффициентов a и свободного члена b l-й строки на коэффициент, лежащий соответственно выше или ниже разрешающего элемента a_{lk} в каждой из этих строк, и вычитания затем этих произведений соответственно из старых коэффициентов a и свободного члена b рассматриваемой строки. В том числе, в k-м столбце вместо разрешающего элемента a_{lk} образуется 1, а вместо остальных коэффициентов — 0.

В новой симплексной таблице (табл. 4.2) повторим вычисления, проведенные в первой итерации, т.е. вновь вычислим величины Z_0', Δ_j' и Z_j' по формулам (4.7) и (4.8). После проверки критерия оптимальности опорного решения либо находится оптимальное решение, либо выявляется отсутствие конечного минимума целевой функции задачи, либо этот процесс продолжается до достижения одного из этих двух результатов. Если в оптимальном решении среди Δ'_j $(j=1,2,\ldots,n)$ нулевыми являются только соответствующие базисным векторам, это оптимальное решение будет *единственным*. Если же для какого-либо небазисного вектора Δ'_j равняется нулю, в общем случае оптимальное решение не является единственным.

Т а б л и ц а 4.2

i	Базис	$\mathbf{C}_{\text{баз}}$	$\mathbf{A}_0$	c_1	c_2	$\ldots$	c_m	c_{m+1}	$\ldots$	c_k	$\ldots$	c_n
				$\mathbf{A}_1$	$\mathbf{A}_2$	$\ldots$	$\mathbf{A}_m$	$\mathbf{A}_{m+1}$	$\ldots$	$\mathbf{A}_k$	$\ldots$	$\mathbf{A}_n$
1	$\mathbf{A}_1$	c_1	b'_1	1	0	$\ldots$	0	$a'_{1,m+1}$	$\ldots$	0	$\ldots$	a'_{1n}
2	$\mathbf{A}_2$	c_2	b'_2	0	1	$\ldots$	0	$a'_{2,m+1}$	$\ldots$	0	$\ldots$	a'_{2n}
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
l	$\mathbf{A}_k$	c_k	b'_l	0	0	$\ldots$	0	$a'_{l,m+1}$	$\ldots$	1	$\ldots$	a'_{ln}
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
m	$\mathbf{A}_m$	c_m	b'_m	0	0	$\ldots$	1	$a'_{m,m+1}$	$\ldots$	0	$\ldots$	a'_{mn}
Δ'_j			Z'_0	0	0	$\ldots$	0	Δ'_{m+1}	$\ldots$	Δ'_k	$\ldots$	Δ'_n

Пример 4.2. Решить с помощью симплексного метода следующую задачу линейного программирования:

$$Z = x_2 - 3x_3 + 2x_5 \qquad (\min)$$

$$\begin{cases} x_1 + 3x_2 - x_3 - 2x_5 = 7 \\ -2x_2 + 4x_3 + x_4 = 12 \\ -4x_2 + 3x_3 + 8x_5 + x_6 = 10 \end{cases}$$

$$x_j \geq 0 \quad (j=1,2,\ldots,6).$$

Решение. По условию

$$\mathbf{A}_1 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \mathbf{A}_2 = \begin{pmatrix} 3 \\ -2 \\ -4 \end{pmatrix}, \mathbf{A}_3 = \begin{pmatrix} -1 \\ 4 \\ 3 \end{pmatrix}, \mathbf{A}_4 = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \mathbf{A}_5 = \begin{pmatrix} -2 \\ 0 \\ 8 \end{pmatrix},$$

$$\mathbf{A}_6 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}, \ \mathbf{A}_0 = \begin{pmatrix} 7 \\ 12 \\ 10 \end{pmatrix}.$$

Воспользовавшись этими соотношениями, условиями задачи, а также формулами (4.7) и (4.8), составим симплексную таблицу задачи (табл. 4.3). В первой итерации симплексного метода для первоначального опорного решения $\mathbf{X}_0 = (7; 0; 0; 12; 0; 10)$ не выполняется критерий оптимальности. Поэтому по формулам (4.9) и (4.10) из базиса выводится вектор $\mathbf{A}_4$, а вместо него в базис вводится вектор $\mathbf{A}_3$ (здесь не выполняется условие того, что целевая функция задачи не имеет конечного минимума). Продолжение симплексной таблицы, соответствующее второй итерации, заполняется с помощью формул (4.7), (4.8) и (4.11).

Оптимальное решение задачи находится в третьей итерации:

$$\mathbf{X}^* = (0; 4; 5; 0; 0; 11), \qquad Z_{\min} = -11.$$

Так как из Δ_j равны нулю только соответствующие базисным векторам, то это оптимальное решение единственно.

Таблица 4.3

i	Базис	$\mathbf{C}_{баз}$	$\mathbf{A}_0$	0	1	-3	0	2	0
				$\mathbf{A}_1$	$\mathbf{A}_2$	$\mathbf{A}_3$	$\mathbf{A}_4$	$\mathbf{A}_5$	$\mathbf{A}_6$
1	$\mathbf{A}_1$	0	7	1	3	-1	0	-2	0
2	$\mathbf{A}_4$	0	12	0	-2	$\boxed{4}$	1	0	0
3	$\mathbf{A}_6$	0	10	0	-4	3	0	8	1
Δ_j			**0**	**0**	**-1**	**<u>3</u>**	**0**	**-2**	**0**
1	$\mathbf{A}_1$	0	10	1	$\boxed{5/2}$	0	1/4	-2	0
2	$\mathbf{A}_3$	-3	3	0	-1/2	1	1/4	0	0
3	$\mathbf{A}_6$	0	1	0	-5/2	0	-3/4	8	1
Δ_j			**-9**	**0**	**<u>1/2</u>**	**0**	**-3/4**	**-2**	**0**
1	$\mathbf{A}_2$	1	4	2/5	1	0	1/10	-4/5	0
2	$\mathbf{A}_3$	-3	5	1/5	0	1	3/10	-2/5	0
3	$\mathbf{A}_6$	0	11	1	0	0	-1/2	6	1
Δ_j			**-11**	**-1/5**	**0**	**0**	**-4/5**	**-8/5**	**0**

Вопросы для повторения и контроля :

1. Как решается система линейных уравнений методом Жордана–Гаусса?
2. Какие случаи возможны в процессе последовательного исключения неизвестных при решении системы линейных уравнений методом Жордана–Гаусса?
3. Что называется выделенными или базисными неизвестными (переменными), а также невыделенными или свободными неизвестными (переменными)?
4. Что называется базисным решением системы линейных уравнений и x уравнением?
5. Для чего необходим симплексный метод, в чем его идея и кто внес вклад в развитие симплексного метода?
6. В чем особенности симплексного метода Данцига?
7. Как находится первоначальное опорное решение в методе Данцига?
8. Как выглядит симплексная таблица?
9. В чем заключается критерий оптимальности опорного решения, условия отсутствия конечного минимума целевой функции задачи линейного программирования и единственности оптимального решения?
10. Как заменяется опорное решение другим, более близким к оптимальному решению опорным решением?

Опорные слова :

Метод Жордана–Гаусса решения системы линейных уравнений, процесс последовательного исключения неизвестных, критерий отсутствия решений системы, выделенные или базисные неизвестные (переменные), невыделенные или свободные неизвестные (переменные), базисное решение системы линейных уравнений, x уравнение, симплексный метод, первоначальное опорное решение, симплексная таблица, критерий оптимальности опорного решения, условие отсутствия конечного минимума целевой функции задачи линейного программирования, замена опорного решения другим, более близким к оптимальному решению опорным решением, формулы преобразования при переходе к новой симплексной таблице, условие единственности оптимального решения.

Тема № 5

Метод искусственного базиса

План :

1. Искусственные базисные векторы и переменные. Расширенная задача линейного программирования.
2. Условие оптимальности опорного решения. Критерий отсутствия оптимального решения. Переход к новому опорному решению.
3. Задача линейного программирования со смешанными ограничениями.

Как мы видели в прошлой теме, если в системе ограничений задачи линейного программирования существуют m линейно независимых векторов, например, векторы $\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_m$, то при неотрицательных правых частях линейных уравнений определено первоначальное опорное решение $\mathbf{X}_0 = (b_1, b_2, \ldots, b_m, 0, \ldots, 0)$, и, исходя из этого решения, с помощью симплексного метода находится оптимальное решение.

Предположим, что задача линейного программирования задана в следующем виде:

найти значения неизвестных, доставляющих минимальное значение функции

$$Z = c_1 x_1 + c_2 x_2 + \ldots + c_n x_n \tag{5.1}$$

и удовлетворяющих условиям

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n \le b_1 \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n \le b_2 \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n \le b_m \end{cases}, \tag{5.2}$$

$$x_1 \ge 0,\ x_2 \ge 0,\ \ldots,\ x_n \ge 0, \tag{5.3}$$

где $b_i \ge 0 \ \ (i = 1, 2, \ldots, m)$.

Все ограничения (5.2) этой задачи являются линейными неравенствами вида «≤» и каждое из них преобразуется в линейное уравнение путем прибавления к левой стороне дополнительной переменной. Векторы $\mathbf{A}_{n+1}, \mathbf{A}_{n+2}, \ldots, \mathbf{A}_{n+m}$, соответствующие этим дополнительным переменным $x_{n+1}, x_{n+2}, \ldots, x_{n+m}$, являются единичными базисными векторами, а сами эти переменные — базисными переменными. Значит, в этом случае также можно применить симплексный метод Данцига.

Однако система ограничений многих задач линейного программирования не приводится к виду (4.5), когда в каждом уравнении присутствует по одному выделенному неизвестному (базисной переменной). В этом случае единичные базисные векторы вводятся искусственным образом.

Например, пусть задача линейного программирования задана в следующем виде:

найти значения неизвестных, доставляющих минимальное значение функции

$$Z = c_1x_1 + c_2x_2 + \ldots + c_nx_n \tag{5.4}$$

и удовлетворяющих условиям

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n = b_2 \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n = b_m \end{cases}, \tag{5.5}$$

$$x_1 \ge 0,\ x_2 \ge 0,\ \ldots,\ x_n \ge 0, \tag{5.6}$$

где $b_i \ge 0 \ \ (i = 1, 2, \ldots, m)$.

В этой задаче для образования единичных базисных векторов прибавим к левой стороне каждого уравнения по одной переменной $x_{n+i} \ge 0 \ \ (i = 1, 2, \ldots, m)$. Эти переменные называются *искусственными переменными*. В этом случае получаем следующую *расширенную задачу линейного программирования*:

найти значения неизвестных, доставляющих минимальное значение функции

$$\bar{Z} = c_1x_1 + c_2x_2 + \ldots + c_nx_n + M\left(x_{n+1} + \ldots + x_{n+m}\right) \tag{5.7}$$

и удовлетворяющих условиям

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n + x_{n+1} = b_1 \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n + x_{n+2} = b_2 \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n + x_{n+m} = b_m \end{cases}, \tag{5.8}$$

$$x_1 \geq 0,\ x_2 \geq 0,\ \dots,\ x_n \geq 0,\ x_{n+1} \geq 0,\ \dots,\ x_{n+m} \geq 0, \qquad (5.9)$$

где M — достаточно большое положительное число, $b_i \geq 0$ $(i = 1, 2, \dots, m)$.

Ясно, что искусственные переменные $x_{n+1}, x_{n+2}, \dots, x_{n+m}$ этой задачи являются базисными переменными. Векторы $\mathbf{A}_{n+1}, \mathbf{A}_{n+2}, \dots, \mathbf{A}_{n+m}$, соответствующие этим искусственным базисным переменным, называются *искусственными базисными векторами*.

Оптимальное решение исходной задачи (5.4) – (5.6) находится на основании следующей теоремы.

Теорема 5.1. *Если в оптимальном решении* $\overline{\mathbf{X}} = (x_1, x_2, \dots, x_n, 0, \dots, 0)$ *расширенной задачи (5.7) – (5.9) искусственные базисные переменные равны нулю, т.е.* $x_{n+i} = 0$ *(*$i = 1, 2, \dots, m$*), то решение* $\mathbf{X} = (x_1, x_2, \dots, x_n)$ *является оптимальным решением исходной задачи (5.4) – (5.6).*

Если в оптимальном решении расширенной задачи, по крайней мере, одна из искусственных переменных отлична от нуля, то исходная задача не имеет решений.

Эту теорему принимаем без доказательства.

Для отыскания оптимального решения расширенной задачи применяется симплексный метод с составлением симплексной таблицы, имеющей на одну строку больше, чем обычная симплексная таблица (табл. 4.1), т.е. у нее существует m+2 я строка. Для заполнения m+1 й и m+2 й строк этой симплексной таблицы величины Z_0 и $\Delta_j = Z_j - c_j$ $(j = 1, 2, \dots, n)$ выражаются в виде $\alpha_j + \beta_j M$, и в m+1 ю строку помещается слагаемое (α_j), независимое от *M*, а в m+2 ю строку — коэффициент (β_j) при *M*.

Вначале, до исключения всех искусственных векторов из базиса используется m+2 я строка (где при переходе к новому опорному решению в базис включается вектор, соответствующий наибольшему элементу m+2 й строки), затем — m+1 я строка. По мере исключения из базиса искусственных векторов соответствующие им столбцы симплексной таблицы можно опускать, так как искусственные переменные обратно в базис не возвращают.

И, наконец, рассмотрим самый общий случай задачи линейного программирования. Здесь система ограничений будет состоять из k $(0 \le k \le m)$ линейных уравнений с выделенными базисными переменными, l $(0 \le l \le m)$ линейных неравенств вида «≤», r $(0 \le r \le m)$ линейных неравенств вида «≥» и $m–k–l–r$ $(0 \le k + l + r \le m)$ линейных уравнений, где базисные переменные не выделены. Такая задача называется *задачей линейного программирования со смешанными ограничениями*.

Если $m - k - l - r > 0$, то каждое линейное неравенство вида «≥» можно преобразовать в линейное уравнение с выделенной базисной переменной. Для этого следует взять одно из уравнений, где базисные переменные не выделены, например, i-е уравнение со свободным членом $b_i > 0$. Затем каждое неравенство вида «≥» нужно разделить на произвольное число так, чтобы его свободный член был меньше b_i, и путем вычитания из левой стороны неотрицательной дополнительной переменной такое неравенство преобразуется в линейное уравнение. После чего такие уравнения почленно вычитаются из i-го уравнения и в полученных уравнениях дополнительные переменные превращаются в базисные переменные. В этом случае базис получившейся задачи линейного программирования состоит из k имеющихся, $l+r$ дополнительных и $m–k–l–r$ искусственных единичных векторов.

Если же $m - k - l - r = 0$, то линейные неравенства вида «≥» преобразуются в линейные уравнения, и в них вводятся искусственные переменные. Тогда базис состоит из k имеющихся, l дополнительных и $m–k–l$ искусственных единичных векторов.

Пример 5.1. Решить методом искусственного базиса следующую задачу линейного программирования:

$$Z = 5x_1 + 3x_2 + 4x_3 - x_4 \qquad (\max)$$

$$\begin{cases} x_1 + 3x_2 + 2x_3 + 2x_4 = 3 \\ 2x_1 + 2x_2 + x_3 + x_4 = 3 \end{cases}$$

$$x_j \ge 0 \qquad (j = 1,2,3,4).$$

<u>Решение</u>. Сначала приведем данную задачу к задаче минимизации целевой функции:

$$Z' = -5x_1 - 3x_2 - 4x_3 + x_4 \qquad (\min)$$

$$\begin{cases} x_1 + 3x_2 + 2x_3 + 2x_4 = 3 \\ 2x_1 + 2x_2 + x_3 + x_4 = 3 \end{cases}$$

$$x_j \geq 0 \quad (j = 1,2,3,4).$$

Введем в задачу искусственные переменные $x_5 \geq 0$ и $x_6 \geq 0$ и получим следующую расширенную задачу:

$$\bar{Z} = -5x_1 - 3x_2 - 4x_3 + x_4 + M(x_5 + x_6) \qquad (\min)$$

$$\begin{cases} x_1 + 3x_2 + 2x_3 + 2x_4 + x_5 = 3 \\ 2x_1 + 2x_2 + x_3 + x_4 + x_6 = 3 \end{cases}$$

$$x_j \geq 0 \quad (j = 1,2,\ldots,6).$$

Полученную задачу разместим в симплексную таблицу и решим ее симплексным методом.

Т а б л и ц а 5.1

i	Базис	$\mathbf{C}_{\text{баз}}$	$\mathbf{A}_0$	-5	-3	-4	1	M	M
				$\mathbf{A}_1$	$\mathbf{A}_2$	$\mathbf{A}_3$	$\mathbf{A}_4$	$\mathbf{A}_5$	$\mathbf{A}_6$
1	$\mathbf{A}_5$	M	3	1	$\boxed{3}$	2	2	1	0
2	$\mathbf{A}_6$	M	3	2	2	1	1	0	1
Δ_j			**0**	**5**	**3**	**4**	**-1**	**0**	**0**
			6	**3**	**$\underline{5}$**	**3**	**3**	**0**	**0**
1	$\mathbf{A}_2$	*-3*	1	1/3	1	2/3	2/3		0
2	$\mathbf{A}_6$	M	1	$\boxed{4/3}$	0	-1/3	-1/3		1
Δ_j			**-3**	**4**	**0**	**2**	**-3**		**0**
			1	**$\underline{4/3}$**	**0**	**-1/3**	**-1/3**		**0**
1	$\mathbf{A}_2$	*-3*	3/4	0	1	$\boxed{3/4}$	3/4		
2	$\mathbf{A}_1$	-5	3/4	1	0	-1/4	-1/4		
Δ_j			**-6**	**0**	**0**	**$\underline{3}$**	**-2**		
			0	**0**	**0**	**0**	**0**		
1	$\mathbf{A}_3$	*-4*	1	0	4/3	1	1		
2	$\mathbf{A}_1$	-5	1	1	1/3	0	0		
Δ_j			**-9**	**0**	**-4**	**0**	**-5**		

Так как в последней, четвертой итерации все $\Delta_j \leq 0$ $(j = 1,2,\ldots,6)$, то решение $\bar{\mathbf{X}}^* = (1;\, 0;\, 1;\, 0;\, 0;\, 0)$ является оптимальным решением расширенной задачи и $\bar{Z}_{\min} = -9$.

Искусственные переменные в оптимальном решении расширенной задачи равны нулю ($x_5 = 0$, $x_6 = 0$). Поэтому на основании теоремы 5.1 оптимальным решением исходной задачи является $\mathbf{X}^* = (1; 0; 1; 0)$, и имеет место $Z'_{\min} = -9$ и $Z_{\max} = 9$.

Вопросы для повторения и контроля :

1. Как можно применить симплексный метод Данцига в случае, когда все ограничения задачи линейного программирования состоят из линейных неравенств вида «≤»?
2. Что такое искусственные базисные векторы и искусственные переменные, для какого вида задачи линейного программирования они применяются и что вы понимаете под расширенной задачей линейного программирования?
3. Что вы знаете о теореме 5.1 и какого вида бывает симплексная таблица для нахождения оптимального решения расширенной задачи?
4. Что такое задача линейного программирования со смешанными ограничениями и как она решается?

Опорные слова :

Искусственные базисные векторы, искусственные переменные, расширенная задача линейного программирования, задача линейного программирования со смешанными ограничениями.

Тема № 6

Теория двойственности в линейном программировании

План :

1. Понятие двойственности в линейном программировании.
2. Симметричные и несимметричные двойственные задачи.
3. Математические модели двойственных задач.
4. Оптимальное решение двойственной задачи. Теоремы двойственности.
5. Двойственные оценки и анализирование на их основе решений экономических задач.

Каждой задаче линейного программирования можно поставить в соответствие другую задачу линейного программирования, называемую *двойственной задачей*. Первоначальная задача называется *исходной задачей*.

Если в исходной задаче требуется, например, максимизировать целевую функцию при заданных ограничениях, то в двойственной ей задаче требуется минимизировать другую целевую функцию. Целевая функция и ограничения, наложенные на переменные двойственной задачи, полностью определяются условиями исходной задачи.

Связь исходной и двойственной задач состоит в том, что коэффициенты c_j $(j = 1,2,\ldots,n)$ целевой функции исходной задачи являются свободными членами системы ограничений двойственной задачи, свободные члены b_i $(i = 1,2,\ldots,m)$ системы ограничений исходной задачи служат коэффициентами целевой функции двойственной задачи, а матрица коэффициентов системы ограничений двойственной задачи является транспонированной матрицей коэффициентов системы ограничений исходной задачи.

Решение двойственной задачи может быть получено из решения исходной и, наоборот, при помощи решения двойственной задачи можно получить решение исходной задачи. Другими словами, исходная и двойственная к ней задача образуют пару взаимно двойственных задач, и любую из них можно рассматривать как исходную или прямую задачу.

Чтобы легче было разобраться в понятии двойственности в линейном программировании, рассмотрим следующую задачу использования сырья (планирования производства).

Предприятие для производства n видов продукции использует m видов сырья, запасы которых составляют b_i $(i = 1,2,\ldots,m)$ единиц. Выручка от реализации единицы продукции j-го вида равна c_j $(j = 1,2,\ldots,n)$ сум, и на ее производство расходуется a_{ij} единиц i-го сырья. Требуется составить такой план выпуска продукции, при котором расход сырья не превышал бы их запасов, а суммарная выручка была бы максимальной.

Обозначим через x_j $(j = 1,2,\ldots,n)$ количество продукции j-го вида. Тогда математическая модель исходной задачи будет следующей:

найти вектор $\mathbf{X} = (x_1, x_2, \ldots, x_n)$, максимизирующий функцию

$$Z = c_1x_1 + c_2x_2 + \ldots + c_nx_n \tag{6.1}$$

и удовлетворяющий условиям

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n \le b_1 \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n \le b_2 \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n \le b_m \end{cases}, \tag{6.2}$$

$$x_1 \ge 0,\ x_2 \ge 0,\ \ldots,\ x_n \ge 0. \tag{6.3}$$

Теперь предположим, что предприятие по некоторым обстоятельствам решило реализовать имеющееся сырье, не выпуская готовую продукцию. При этом оно заинтересовано получить выручку не менее той суммы, которую можно получить при изготовлении готовой продукции. С другой стороны, предприятие, закупающее данное сырье, стремится к минимуму затрат по его приобретению. Если обозначить через y_i $(i = 1,2,\ldots,m)$ цену единицы i-го сырья, то необходимо установить такие оптимальные цены на сырье, которые удовлетворяли бы указанным условиям.

Для общей цены всего сырья, затраченного на производство единицы продукции j-го вида, имеет место соотношение

$\sum_{i=1}^{m} a_{ij} y_i \geq c_j \quad (j = 1, 2, \ldots, n)$. Цена всего наличного сырья выражается величиной $\sum_{i=1}^{m} b_i y_i$. В этом случае двойственная задача в математической форме выражается следующим образом:

найти вектор $\mathbf{Y} = (y_1, y_2, \ldots, y_m)$, минимизирующий функцию

$$W = b_1 y_1 + b_2 y_2 + \ldots + b_m y_m \tag{6.4}$$

и удовлетворяющий условиям

$$\begin{cases} a_{11} y_1 + a_{21} y_2 + \ldots + a_{m1} y_m \geq c_1 \\ a_{12} y_1 + a_{22} y_2 + \ldots + a_{m2} y_m \geq c_2 \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ a_{1n} y_1 + a_{2n} y_2 + \ldots + a_{mn} y_m \geq c_n \end{cases}, \tag{6.5}$$

$$y_1 \geq 0,\ y_2 \geq 0,\ \ldots,\ y_m \geq 0. \tag{6.6}$$

Теперь запишем в матричном виде и задачу (6.1) – (6.3), и задачу (6.4) – (6.6).

Исходная задача. Найти матрицу-столбец $\mathbf{X} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}$, максимизирующую линейную функцию $Z = \mathbf{CX}$ и удовлетворяющую условиям $\mathbf{AX} \leq \mathbf{A}_0$, $\mathbf{X} \geq 0$.

Двойственная задача. Найти матрицу-строку $\mathbf{Y} = (y_1, y_2, \ldots, y_m)$, минимизирующую линейную функцию $W = \mathbf{YA}_0$ и удовлетворяющую условиям $\mathbf{YA} \geq \mathbf{C}$, $\mathbf{Y} \geq 0$.

Ясно, что в обеих этих задачах система ограничений задается неравенствами и на переменные налагаются условия неотрицательности. Задачи (6.1) – (6.3) и (6.4) – (6.6) образуют пару *симметричных двойственных задач*.

Пример 6.1. Дана задача линейного программирования:

$$Z = 5x_1 + 3x_2 + 4x_3 - x_4 \qquad \text{(max)}$$

$$\begin{cases} x_1 + 3x_2 - 2x_3 + 2x_4 \le 3 \\ 2x_1 + 2x_2 + x_3 - x_4 \le 3 \end{cases}$$

$$x_j \ge 0 \quad (j = 1,2,3,4).$$

Составить двойственную к ней задачу.

Решение. Двойственная к исходной задача имеет вид:

$$W = 3y_1 + 3y_2 \qquad (\min)$$

$$\begin{cases} y_1 + 2y_2 \ge 5 \\ 3y_1 + 2y_2 \ge 3 \\ -2y_1 + y_2 \ge 4 \\ 2y_1 - y_2 \ge -1 \end{cases}$$

$$y_i \ge 0 \quad (i = \overline{1,2}).$$

В паре *несимметричных двойственных задач* система ограничений исходной задачи задается в виде уравнений, а двойственной задачи — в виде неравенств, причем переменные двойственной задачи могут быть и отрицательными. Выразим эти задачи в матричном виде.

Исходная задача. Найти матрицу-столбец $\mathbf{X} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}$, минимизирующую линейную функцию $Z = \mathbf{CX}$ и удовлетворяющую условиям $\mathbf{AX} = \mathbf{A}_0$, $\mathbf{X} \ge 0$.

Двойственная задача. Найти матрицу-строку $\mathbf{Y} = (y_1, y_2, \ldots, y_m)$, максимизирующую линейную функцию $W = \mathbf{YA}_0$ и удовлетворяющую условиям $\mathbf{YA} \le \mathbf{C}$.

В обоих парах задач $\mathbf{C} = (c_1, c_2, \ldots, c_n)$ — матрица-строка; $\mathbf{A} = (a_{ij})$ — матрица системы ограничений; $\mathbf{A}_0 = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{pmatrix}$ — матрица-столбец.

Пример 6.2. Дана задача линейного программирования:

$$Z = 2x_1 + x_2 + 3x_3 \qquad (\min)$$

$$\begin{cases} -x_1 + 3x_2 - 5x_3 = 12 \\ 2x_1 - x_2 + 4x_3 = 24 \\ 3x_1 + x_2 + x_3 = 18 \end{cases}$$

$$x_1 \geq 0,\ x_2 \geq 0,\ x_3 \geq 0.$$

Составить двойственную к ней задачу.

Решение. Двойственная к исходной задача имеет вид:

$$W = 12y_1 + 24y_2 + 18y_3 \qquad (\max)$$

$$\begin{cases} -y_1 + 2y_2 + 3y_3 \leq 2 \\ 3y_1 - y_2 + y_3 \leq 1 \\ -5y_1 + 4y_2 + y_3 \leq 3 \end{cases}.$$

Таким образом, в паре двойственных задач своеобразие связи между исходной и двойственной задачами заключается в том, что каждому ограничению-неравенству исходной задачи соответствует неотрицательная переменная двойственной, а каждому ограничению-равенству — переменная произвольного знака.

На основе рассмотренных выше симметричных и несимметричных двойственных задач можно сделать вывод о том, что математические модели пар двойственных задач могут быть в следующих формах:

Симметричные задачи

(1) *Исходная задача*	*Двойственная задача*
$Z = \mathbf{CX}$ (min)	$W = \mathbf{YA}_0$ (max)
$\mathbf{AX} \geq \mathbf{A}_0$	$\mathbf{YA} \leq \mathbf{C}$
$\mathbf{X} \geq 0$	$\mathbf{Y} \geq 0$
(2) *Исходная задача*	*Двойственная задача*
$Z = \mathbf{CX}$ (max)	$W = \mathbf{YA}_0$ (min)
$\mathbf{AX} \leq \mathbf{A}_0$	$\mathbf{YA} \geq \mathbf{C}$
$\mathbf{X} \geq 0$	$\mathbf{Y} \geq 0$

(1) *Исходная задача*

$$Z = \mathbf{CX} \quad (\min)$$
$$\mathbf{AX} = \mathbf{A}_0$$
$$\mathbf{X} \geq 0$$

Двойственная задача

$$W = \mathbf{YA}_0 \quad (\max)$$
$$\mathbf{YA} \leq \mathbf{C}$$

(2) *Исходная задача*

$$Z = \mathbf{CX} \quad (\max)$$
$$\mathbf{AX} = \mathbf{A}_0$$
$$\mathbf{X} \geq 0$$

Двойственная задача

$$W = \mathbf{YA}_0 \quad (\min)$$
$$\mathbf{YA} \geq \mathbf{C}$$

Таким образом, прежде чем составлять двойственную к исходной задачу, необходимо привести систему ограничений исходной задачи к соответствующему виду.

Связь между оптимальными решениями пары двойственных задач устанавливает следующая теорема.

Теорема 6.1 (первая теорема двойственности). *Если из пары двойственных задач одна обладает оптимальным решением, то и другая имеет решение, причем для экстремальных значений целевых функций выполняется соотношение:*

$$\min Z = \max W.$$

Если целевая функция одной из задач не ограничена, то другая не имеет решения.

Эту теорему принимаем без доказательства.

Если оптимальное решение $\mathbf{X}^* = (x_1^*, x_2^*, \ldots, x_n^*)$ исходной задачи найдено симплексным методом, то оптимальное решение двойственной задачи находится по формуле $\mathbf{Y}^* = \mathbf{C}^*\mathbf{D}^{-1}$. Здесь $\mathbf{C}^*$ — вектор коэффициентов целевой функции, соответствующих базисным векторам последней симплексной таблицы, $\mathbf{D}^{-1}$ — матрица, обратная к матрице $\mathbf{D}$, составленной из координат базисных векторов последней симплексной таблицы, в которой найдено оптимальное решение исходной задачи. При этом матрица $\mathbf{D}^{-1}$ составлена из координат базисных векторов первой симплексной таблицы в последней симплексной таблице. Если исходная и двойственная задачи симметричны, то для оптимального решения двойственной задачи берутся абсолютные значения Δ_j.

Пример 6.3. Найти оптимальное решение исходной и двойственной к ней задач:

$$Z = x_2 - 3x_3 + 2x_5 \qquad (\min)$$

$$\begin{cases} x_1 + 3x_2 - x_3 - 2x_5 = 7 \\ -2x_2 + 4x_3 + x_4 = 12 \\ -4x_2 + 3x_3 + 8x_5 + x_6 = 10 \end{cases}$$

$$x_j \ge 0 \quad (j = 1, 2, \ldots, 6).$$

<u>Решение</u>. Составим двойственную к исходной задачу:

$$W = 7y_1 + 12y_2 + 10y_3 \qquad (\max)$$

$$\begin{cases} y_1 \le 0 \\ 3y_1 - 2y_2 - 4y_3 \le 1 \\ -y_1 + 4y_2 + 3y_3 \le -3 \\ y_2 \le 0 \\ -2y_1 + 8y_3 \le 2 \\ y_3 \le 0 \end{cases}.$$

Решим исходную задачу симплексным методом (табл. 4.3). В результате получим оптимальное решение $\mathbf{X}^* = (0; 4; 5; 0; 0; 11)$ исходной задачи. Из таблицы 4.3 следует $\mathbf{C}^* = (1; -3; 0)$ и $\mathbf{D}^{-1} = \begin{pmatrix} 2/5 & 1/10 & 0 \\ 1/5 & 3/10 & 0 \\ 1 & -1/2 & 1 \end{pmatrix}$. Значит, оптимальное решение двойственной задачи равно

$$\mathbf{Y}^* = (1; -3; 0) \times \begin{pmatrix} 2/5 & 1/10 & 0 \\ 1/5 & 3/10 & 0 \\ 1 & -1/2 & 1 \end{pmatrix} = (-1/5; -4/5; 0).$$

По теореме 6.1 $\min Z = \max W = -11$.

С экономической точки зрения решение исходной задачи позволяет получить оптимальный план выпуска продукции, а решение двойственной задачи — оптимальную систему условных оценок используемых ресурсов.

Оптимальные решения пары взаимно двойственных симметричных задач связаны не только равенством значений целевых функций. В них соблюдаются и другие соотношения, которые устанавливает следующая теорема.

Теорема 6.2 (вторая теорема двойственности). *Для оптимальности допустимых решений* $\mathbf{X}^*$ *исходной и* $\mathbf{Y}^*$ *двойственной задач (6.1) – (6.3) и (6.4) – (6.6) необходимо и достаточно, чтобы выполнялись соотношения:*

$$\textit{если} \ \sum_{j=1}^{n} a_{ij} x_j^* < b_i \ (i = 1,2,\ldots,m), \ \ \textit{то} \ \ y_i^* = 0;$$

$$\textit{если} \ \sum_{i=1}^{m} a_{ij} y_i^* > c_j \ (j = 1,2,\ldots,n), \ \ \textit{то} \ \ x_j^* = 0,$$

где $\mathbf{X}^* = (x_1^*, x_2^*, \ldots, x_n^*)$ *и* $\mathbf{Y}^* = (y_1^*, y_2^*, \ldots, y_m^*)$ — *оптимальные решения соответственно исходной и двойственной задач.*

Эту теорему также принимаем без доказательства.

На основании этой теоремы переменные и ограничения пары взаимно двойственных симметричных задач для оптимального решения связаны следующим образом: если по оптимальному решению исходной задачи какой-то i-й ресурс используется не полностью, т.е. $\sum_{j=1}^{n} a_{ij} x_j^* < b_i$, то его оптимальная оценка равна нулю, т.е. $y_i^* = 0$.

С экономической точки зрения это означает, что положительную двойственную оценку могут иметь лишь ресурсы, полностью используемые в оптимальном решении; оценки не полностью используемых ресурсов всегда равны нулю. Наибольшей величине оценки соответствует наиболее дефицитный ресурс. Недефицитному ресурсу соответствует оценка, равная нулю.

Величина двойственной оценки того или иного ресурса показывает, насколько возросло бы максимальное значение целевой функции, если бы объем данного ресурса увеличился на одну единицу.

Если же затраты на ресурсы, потребляемые при производстве единицы какого-то j-го продукта, оцененные условно в оптимальных оценках, превышают доход от реализации единицы этого продукта, т.е. $\sum_{i=1}^{m} a_{ij} y_i^* > c_j$, то этот продукт по оптимальному решению не производится, т.е. $x_j^* = 0$.

Это означает, что если j-я переменная исходной задачи входит в оптимальное решение с положительным значением, то соответствующее ограничение в оптимальном решении двойственной задачи будет выполняться как строгое равенство. А если же j-я переменная исходной задачи в оптимальном решении равна нулю, то в оптимальном решении двойственной задачи соответствующее j-е ограничение будет, как правило, обращаться в строгое неравенство.

Пример 6.4. На предприятии имеется четыре вида сырья в количествах: вида A — 18 единиц; вида B — 16; вида C — 8 и вида D — 6 единиц. Нормы затрат каждого вида сырья на единицу продукции вида №1 составляют соответственно 1, 2, 1, 0; вида №2 — соответственно 2, 1, 1, 1 и вида №3 — соответственно 1, 1, 0, 1 единиц. Прибыль от реализации одной единицы продукции №1 равна 3 у.е., №2 — 4 у.е., №3 — 2 у.е.

Составить план производства трех видов продукции, при котором доход предприятия от реализации всей продукции был бы максимальным. Найти двойственные оценки и сделать анализ решения.

Решение. Обозначим через x_1 — количество выпускаемых изделий вида №1, через x_2 — количество выпускаемых изделий вида №2, через x_3 — количество выпускаемых изделий вида №3.

Экономико-математическая модель исходной задачи имеет вид:

$$Z = 3x_1 + 4x_2 + 2x_3 \qquad (\max)$$

$$\begin{cases} x_1 + 2x_2 + x_3 \le 18 \\ 2x_1 + x_2 + x_3 \le 16 \\ x_1 + x_2 \le 8 \\ x_2 + x_3 \le 6 \end{cases}$$

$$x_j \geq 0 \quad (j = 1,2,3).$$

Оценки двойственной задачи обозначим через y_i $(i = \overline{1,4})$, где y_1 — оценка 1 единицы сырья *A*, y_2 — *B*, y_3 — *C*, y_4 — *D*.

Тогда экономико-математическая модель двойственной задачи примет вид:

$$W = 18y_1 + 16y_2 + 8y_3 + 6y_4 \qquad (\min)$$

$$\begin{cases} y_1 + 2y_2 + y_3 \geq 3 \\ 2y_1 + y_2 + y_3 + y_4 \geq 4 \\ y_1 + y_2 + y_4 \geq 2 \end{cases}$$

$$y_i \geq 0 \quad (i = \overline{1,4}).$$

Решим исходную задачу симплексным методом (табл.6.1).

Оптимальное решение исходной задачи имеет вид

$$\mathbf{X}^* = (x_1^*; x_2^*; x_3^*; x_4^*; x_5^*; x_6^*; x_7^*) = (5; 3; 3; 4; 0; 0; 0).$$

Значение целевой функции для этого решения равно $Z_{\max} = 33$.

С учетом соответствия между переменными пары взаимно двойственных задач значения оптимальных двойственных оценок будут:

$$y_1^* = 0, \; y_2^* = 1/2, \; y_3^* = 2, \; y_4^* = 3/2.$$

Оценка для сырья вида *A* равна нулю, так как сырье этого вида в оптимальном решении используется не полностью — соответствующее ограничение в исходной задаче выполняется как строгое неравенство:

$$5 + 2 \cdot 3 + 3 < 18 \text{ или } 14 < 18.$$

Для сырья остальных видов оценки положительны, что указывает на полное использование сырья видов *B*, *C*, *D*. Здесь соответствующие ограничения в исходной задаче выполняются как равенства.

С другой стороны, в оптимальном решении продукция всех трех видов производится, так как соответствующие ограничения в двойственной задаче выполняются как равенства.

Теперь проведем анализ решения с учетом двойственных оценок.

Увеличение объема сырья *B* на одну единицу позволило бы получить оптимальное решение, для которого доход увеличился бы на 0,5 у.е., а увеличение объема сырья *C* на одну единицу дало бы увеличение дохода на 2 у.е. Аналогично при увеличении объема сырья *D* на одну единицу доход увеличится на 1,5 у.е. Отсюда понятна и нулевая оценка, полученная для сырья *A*, т.е. $y_1^* = 0$. Поскольку сырье *A* не полностью используется, то дальнейшее увеличение объема этого вида сырья не повлияет на оптимальный план производства и на сумму дохода.

Т а б л и ц а 6.1

i	Базис	$\mathbf{C}_{\text{баз}}$	$\mathbf{A}_0$	-3	-4	-2	0	0	0	0
				$\mathbf{A}_1$	$\mathbf{A}_2$	$\mathbf{A}_3$	$\mathbf{A}_4$	$\mathbf{A}_5$	$\mathbf{A}_6$	$\mathbf{A}_7$
1	$\mathbf{A}_4$	0	18	1	2	1	1	0	0	0
2	$\mathbf{A}_5$	0	16	2	1	1	0	1	0	0
3	$\mathbf{A}_6$	0	8	1	1	0	0	0	1	0
4	$\mathbf{A}_7$	0	6	0	$\boxed{1}$	1	0	0	0	1
Δ_j			**0**	**-3**	**<u>4</u>**	**2**	**0**	**0**	**0**	**0**
1	$\mathbf{A}_4$	0	6	1	0	-1	1	0	0	-2
2	$\mathbf{A}_5$	0	10	2	0	0	0	1	0	-1
3	$\mathbf{A}_6$	0	2	$\boxed{1}$	0	-1	0	0	1	-1
4	$\mathbf{A}_2$	-4	6	0	1	1	0	0	0	1
Δ_j			**-24**	**<u>3</u>**	**0**	**2**	**0**	**0**	**0**	**-4**
1	$\mathbf{A}_4$	0	4	0	0	0	1	0	-1	-1
2	$\mathbf{A}_5$	0	6	0	0	$\boxed{2}$	0	1	-2	1
3	$\mathbf{A}_1$	-3	2	1	0	-1	0	0	1	-1
4	$\mathbf{A}_2$	-4	6	0	1	1	0	0	0	1
Δ_j			**-30**	**0**	**0**	**<u>1</u>**	**0**	**0**	**-3**	**-1**
1	$\mathbf{A}_4$	0	4	0	0	0	1	0	-1	-1
2	$\mathbf{A}_3$	-2	3	0	0	1	0	1/2	-1	1/2
3	$\mathbf{A}_1$	-3	5	1	0	0	0	1/2	0	-1/2
4	$\mathbf{A}_2$	-4	3	0	1	0	0	-1/2	1	1/2
Δ_j			**-33**	**0**	**0**	**0**	**0**	**-1/2**	**-2**	**-3/2**

По двойственным оценкам можно судить о дефицитности ресурсов или сырья. Самые высокие значения оценок указывают на более дефицитные виды сырья. В нашем случае наиболее дефицитным сырьем является сырье вида *C*, для которого двойственная оценка $y_3^* = 2$. Наименее дефицитным является сырье вида B, для которого оценка $y_2^* = 1/2$. Совсем недефицитным (избыточным) является сырье вида A, оценка его $y_1^* = 0$.

Вопросы для повторения и контроля :

1. Что называется исходной задачей и двойственной задачей, какова связь между ними?
2. Как составляется двойственная задача к задаче использования сырья?
3. Что называется симметричными двойственными задачами и как они выражаются в матричной форме?
4. Что называется несимметричными двойственными задачами и как они выражаются в матричной форме?
5. Какие виды математических моделей пар двойственных задач вы знаете?
6. Что вы знаете о первой теореме двойственности (теорема 6.1) и как находится оптимальное решение двойственной задачи?
7. Что вы знаете о второй теореме двойственности (теорема 6.2) и как связаны переменные и ограничения пары взаимно двойственных симметричных задач для оптимального решения?
8. Что такое двойственные оценки и каков экономический смысл оптимальных решений пары взаимно двойственных задач?

Опорные слова :

Исходная задача, двойственная задача, симметричные двойственные задачи, несимметричные двойственные задачи, математические модели пар двойственных задач, оптимальное решение двойственной задачи, теоремы двойственности, двойственные оценки.

Тема № 7

Двойственный симплексный метод

План :

1. Двойственный симплексный метод.
2. Условия оптимальности и отсутствия решения.

Как мы видели в предыдущей теме, для нахождения решения исходной задачи линейного программирования можно перейти к двойственной задаче и, воспользовавшись ее оптимальным решением, найти оптимальное решение исходной задачи.

А в этой теме познакомимся с еще одним методом, позволяющим найти оптимальное решение как двойственной, так и исходной задачи — *двойственным симплексным методом*, основанном на теории двойственности. У этого метода есть ряд преимуществ по сравнению с обычным симплексным методом:

1) в условиях задачи, решаемой двойственным симплексным методом, свободные члены b_i $(i = 1,2,\ldots,m)$ могут быть произвольного знака;

2) с помощью двойственного симплексного метода одновременно находится решение как двойственной, так и исходной задачи, или устанавливается отсутствие решения обоих задач;

3) при решении двойственным симплексным методом, когда некоторые ограничения в исходной задаче являются линейными неравенствами вида «≥» или некоторые ее свободные члены отрицательны, уменьшается количество вычислений и размер симплексной таблицы.

Пусть задача линейного программирования задана в следующем виде:

найти вектор $\mathbf{X} = (x_1, x_2, \ldots, x_n)$, минимизирующий функцию

$$Z = c_1x_1 + c_2x_2 + \ldots + c_nx_n \tag{7.1}$$

и удовлетворяющий условиям

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n = b_2 \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n = b_m \end{cases}, \tag{7.2}$$

$$x_1 \geq 0,\ x_2 \geq 0,\ \ldots,\ x_n \geq 0, \tag{7.3}$$

где по крайней мере один из свободных членов b_i $(i = 1,2,\ldots,m)$ отрицателен.

В этом случае двойственная задача выражается в следующем виде:

найти вектор $\mathbf{Y} = (y_1, y_2, \ldots, y_m)$, максимизирующий функцию

$$W = b_1 y_1 + b_2 y_2 + \ldots + b_m y_m$$

и удовлетворяющий условиям

$$\begin{cases} a_{11}y_1 + a_{21}y_2 + \ldots + a_{m1}y_m \leq c_1 \\ a_{12}y_1 + a_{22}y_2 + \ldots + a_{m2}y_m \leq c_2 \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ a_{1n}y_1 + a_{2n}y_2 + \ldots + a_{mn}y_m \leq c_n \end{cases}.$$

Составим для задачи (7.1) – (7.3) симплексную таблицу, подобную таблице 4.1. В симплексном методе Данцига при переходе к новому опорному решению для ввода в базис сначала определяется вектор $\mathbf{A}_k$, удовлетворяющий условию $\Delta_k = \max\limits_{\Delta_j > 0} \Delta_j$, затем из базиса исключается вектор $\mathbf{A}_l$, удовлетворяющий условию $b_l / a_{lk} = \min\limits_{a_{ik} > 0} (b_i / a_{ik})$.

В отличие от этого метода, в двойственном симплексном методе сначала определяется вектор, исключаемый из базиса, затем — вектор, вводимый вместо него в базис. Из базиса исключается вектор $\mathbf{A}_l$, удовлетворяющий условию

$$b_l = \min_{b_i < 0} b_i.$$

Если для $j = 1,2,\ldots,n$ $a_{lj} \geq 0$, то исходная и двойственная к ней задача не имеют решения.

В базис вводится вектор $\mathbf{A}_k$, удовлетворяющий условию

$$\Delta_k / a_{lk} = \min_{\substack{a_{lj}<0 \\ \Delta_j<0}} \left(\Delta_j / a_{lj}\right).$$

Затем симплексная таблица преобразовывается как в обычном симплексном методе. Этот процесс продолжается до тех пор, пока не будет $\Delta_j \le 0$ для всех $j = 1,2,\ldots,n$ и $b_i \ge 0$ для всех $i = 1,2,\ldots,m$, т.е. до нахождения оптимального решения или установления того, что задачи не имеют решения.

Оптимальное решение двойственной задачи также, как и в предыдущей теме, будет состоять из Δ_j, соответствующих в последней симплексной таблице базисным векторам первой симплексной таблицы. Если исходная и двойственная задачи симметричны, то для оптимального решения двойственной задачи берутся абсолютные значения Δ_j.

Пример 7.1. Решить задачу линейного программирования двойственным симплексным методом:

$$Z = 3x_1 + 4x_2 + 5x_3 + 6x_4 \qquad (\min)$$

$$\begin{cases} 2x_1 + x_2 - x_3 + 5x_4 \ge 5 \\ 3x_1 - 2x_2 + x_3 + 4x_4 \ge 4 \end{cases}$$

$$x_j \ge 0 \quad (j = 1,2,3,4).$$

Т а б л и ц а 7.1

i	Базис	$\mathbf{C}_{\text{баз}}$	$\mathbf{A}_0$	3	4	5	6	0	0
				$\mathbf{A}_1$	$\mathbf{A}_2$	$\mathbf{A}_3$	$\mathbf{A}_4$	$\mathbf{A}_5$	$\mathbf{A}_6$
1	$\mathbf{A}_5$	0	-5	-2	-1	1	$\boxed{-5}$	1	0
2	$\mathbf{A}_6$	0	-4	-3	2	-1	-4	0	1
Δ_j			**0**	**-3**	**-4**	**-5**	**<u>-6</u>**	**0**	**0**
1	$\mathbf{A}_4$	6	1	2/5	1/5	-1/5	1	-1/5	0
2	$\mathbf{A}_6$	0	0	-7/5	14/5	-9/5	0	-4/5	1
Δ_j			**6**	**-3/5**	**-14/5**	**-31/5**	**0**	**-6/5**	**0**

<u>Решение</u>. Двойственная к исходной задача будет иметь следующий вид:

$$W = 5y_1 + 4y_2 \qquad (\max)$$

$$\begin{cases} 2y_1 + 3y_2 \le 3 \\ y_1 - 2y_2 \le 4 \\ -y_1 + y_2 \le 5 \\ 5y_1 + 4y_2 \le 6 \end{cases}$$

$$y_i \ge 0 \quad (i = \overline{1,2}).$$

Приведем исходную задачу к каноническому виду:

$$Z = 3x_1 + 4x_2 + 5x_3 + 6x_4 + 0x_5 + 0x_6 \qquad (\min)$$

$$\begin{cases} 2x_1 + x_2 - x_3 + 5x_4 - x_5 = 5 \\ 3x_1 - 2x_2 + x_3 + 4x_4 - x_6 = 4 \end{cases}$$

$$x_j \ge 0 \quad (j = \overline{1,6}).$$

Т а б л и ц а 7.2

i	Базис	$\mathbf{C}_{\text{баз}}$	$\mathbf{A}_0$	-5	-4	0	0	0	0
				$\mathbf{A}_1$	$\mathbf{A}_2$	$\mathbf{A}_3$	$\mathbf{A}_4$	$\mathbf{A}_5$	$\mathbf{A}_6$
1	$\mathbf{A}_3$	0	3	2	3	1	0	0	0
2	$\mathbf{A}_4$	0	4	1	-2	0	1	0	0
3	$\mathbf{A}_5$	0	5	-1	1	0	0	1	0
4	$\mathbf{A}_6$	0	6	$\boxed{5}$	4	0	0	0	1
Δ_j			**0**	**<u>5</u>**	**4**	**0**	**0**	**0**	**0**
1	$\mathbf{A}_3$	0	3/5	0	7/5	1	0	0	-2/5
2	$\mathbf{A}_4$	0	14/5	0	-14/5	0	1	0	-1/5
3	$\mathbf{A}_5$	0	31/5	0	9/5	0	0	1	1/5
4	$\mathbf{A}_1$	-5	6/5	1	4/5	0	0	0	1/5
Δ_j			**-6**	**0**	**0**	**0**	**0**	**0**	**-1**

Для выделения базисных векторов в условиях данной задачи оба уравнения в ограничениях умножим на –1 и получим следующую задачу:

$$Z = 3x_1 + 4x_2 + 5x_3 + 6x_4 + 0x_5 + 0x_6 \qquad (\min)$$

$$\begin{cases} -2x_1 - x_2 + x_3 - 5x_4 + x_5 = -5 \\ -3x_1 + 2x_2 - x_3 - 4x_4 + x_6 = -4 \end{cases}$$

$$x_j \ge 0 \quad (j = \overline{1,6}).$$

Составим для этой задачи симплексную таблицу и решим ее двойственным симплексным методом (табл. 7.1).

Оптимальное решение задачи найдено во второй итерации:

$$\mathbf{X}^* = (0;\, 0;\, 0;\, 1), \qquad Z_{\min} = 6.$$

Оптимальное решение двойственной задачи следующее:

$$\mathbf{Y}^* = (6/5;\, 0), \qquad W_{\max} = 6.$$

Если сначала решим симплексным методом двойственную задачу, то получим этот же результат (табл. 7.2).

Вопросы для повторения и контроля :

1. В чем сущность двойственного симплексного метода?
2. Каковы условия оптимальности и отсутствия решения, а также правила перехода к новому базису в двойственном симплексном методе?

Опорные слова :

Двойственный симплексный метод, условия оптимальности в двойственном симплексном методе, условия отсутствия решения в двойственном симплексном методе, правила перехода к новому базису в двойственном симплексном методе.

Тема № 8

Транспортная задача. Методы нахождения первоначального опорного решения

План :

1. Постановка и математическая модель транспортной задачи.
2. Закрытая и открытая модели транспортной задачи.
3. Приложения транспортной задачи к решению некоторых экономических задач.
4. Нахождение первоначального опорного решения.
5. Метод северо-западного угла.
6. Методы минимальной стоимости и двойного предпочтения.

Транспортная задача — задача о наиболее экономном плане перевозок однородного или взаимозаменяемого продукта из пунктов производства (станций отправления) в пункты потребления (станции назначения) — является важнейшей частной задачей линейного программирования, имеющей обширные практические приложения не только к проблемам транспорта.

В настоящее время транспортная задача линейного программирования широко применяется как в теоретических разработках, так и в практике планирования различных экономических процессов. Особо важное значение она имеет при решении вопросов рационализации поставок важнейших видов промышленной и сельскохозяйственной продукции, а также оптимального планирования грузопотоков и различных видов транспорта.

Транспортная задача выделяется в линейном программировании определенностью экономической характеристики, особенностями математической модели, наличием специфических методов решения.

Рассмотрим теперь постановку и математическую модель транспортной задачи. Некоторый однородный продукт, сосредоточенный у m поставщиков A_i в количестве a_i $(i = 1,2,\ldots,m)$ единиц соответственно, необходимо доставить n потребителям B_j в количестве b_j $(j = 1,2,\ldots,n)$ единиц соответственно. Известна стоимость c_{ij} перевозки единицы груза от i-го поставщика к j-му потребителю.

Требуется составить план перевозок, позволяющий вывезти все грузы, полностью удовлетворить потребности и имеющий минимальную стоимость, т.е. найти, сколько единиц груза должно быть отправлено из i-го пункта поставки в j-й пункт потребления.

Обозначим через x_{ij} количество единиц груза, запланированных к перевозке от i-го поставщика к j-му потребителю. Тогда условие транспортной задачи можно записать в виде следующей таблицы:

Таблица 8.1

Поставщики	Потребители				Запасы
	B_1	B_2	…	B_n	
A_1	c_{11} x_{11}	c_{12} x_{12}	…	c_{1n} x_{1n}	a_1
A_2	c_{21} x_{21}	c_{22} x_{22}	…	c_{2n} x_{2n}	a_2
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
A_m	c_{m1} x_{m1}	c_{m2} x_{m2}	…	c_{mn} x_{mn}	a_m
Потребности	b_1	b_2	…	b_n	$\sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_j$

Составим теперь математическую модель транспортной задачи. Если от i-го поставщика к j-му потребителю запланировано к перевозке x_{ij} единиц груза, то стоимость перевозки составит $c_{ij}x_{ij}$. А стоимость всего плана перевозок равна $Z = \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij}$.

Систему ограничений получаем из следующих условий задачи:

а) все грузы должны быть вывезены, т.е. должно выполняться $\sum_{j=1}^{n} x_{ij} = a_i \ (i = 1,2,\ldots,m)$ (эти уравнения получаются из строк таблицы 8.1);

б) все потребности должны быть удовлетворены, т.е. должно выполняться $\sum_{i=1}^{m} x_{ij} = b_j \ \ (j = 1,2,\ldots,n)$ (эти уравнения получаются из столбцов таблицы 8.1).

Таким образом, математическая модель транспортной задачи имеет следующий вид:

найти значения неизвестных, минимизирующих линейную функцию

$$Z = \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij}$$

и удовлетворяющих условиям

$$\sum_{j=1}^{n} x_{ij} = a_i \quad (i = 1,2,\ldots,m), \tag{8.1}$$

$$\sum_{i=1}^{m} x_{ij} = b_j \quad (j = 1,2,\ldots,n), \tag{8.2}$$

$$x_{ij} \geq 0 \quad (i = 1,2,\ldots,m; \ \ j = 1,2,\ldots,n). \tag{8.3}$$

В этой модели предполагается, что суммарные запасы равны суммарным потребностям, т.е.

$$\sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_j \,. \tag{8.4}$$

Такая модель называется *закрытой моделью транспортной задачи*.

Теорема 8.1. *Любая транспортная задача с закрытой моделью имеет допустимое решение*.

Доказательство. По условию

$$\sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_j = A > 0 .$$

Тогда набор чисел $x_{ij} = \dfrac{a_i b_j}{A}$ $(i = 1,2,\ldots,m; \;\; j = 1,2,\ldots,n)$ будет допустимым решением транспортной задачи. Действительно, эти числа удовлетворяют условиям (8.1) – (8.3):

$$x_{ij} = \frac{a_i b_j}{A} \geq 0, \text{ так как } a_i \geq 0,\ b_j \geq 0,\ A > 0;$$

$$\sum_{j=1}^{n} x_{ij} = \sum_{j=1}^{n} \frac{a_i b_j}{A} = a_i \frac{\sum\limits_{j=1}^{n} b_j}{A} = a_i;$$

$$\sum_{i=1}^{m} x_{ij} = \sum_{i=1}^{m} \frac{a_i b_j}{A} = b_j \frac{\sum\limits_{i=1}^{m} a_i}{A} = b_j.$$

Если в модели транспортной задачи не выполняется соотношение (8.4), то такая модель называется *открытой моделью транспортной задачи*. Здесь может быть два случая:

а) суммарные запасы превышают суммарные потребности, т.е. имеет место $\sum\limits_{i=1}^{m} a_i > \sum\limits_{j=1}^{n} b_j$;

б) суммарные запасы меньше суммарных потребностей, т.е. имеет место $\sum\limits_{i=1}^{m} a_i < \sum\limits_{j=1}^{n} b_j$.

Открытая модель решается приведением к закрытой модели. Для этого в случае а) вводится фиктивный потребитель B_{n+1}, потребности которого равны $b_{n+1} = \sum\limits_{i=1}^{m} a_i - \sum\limits_{j=1}^{n} b_j$, а в случае б) вводится фиктивный поставщик A_{m+1}, запасы которого равны $a_{m+1} = \sum\limits_{j=1}^{n} b_j - \sum\limits_{i=1}^{m} a_i$.

Как стоимость перевозки единицы груза до фиктивного потребителя, так и стоимость перевозки единицы груза от фиктивного поставщика полагаются равными нулю, так как груз в обоих случаях не перевозится.

Транспортная задача применяется также при решении экономических задач, которые по своему характеру не имеют ничего общего с транспортировкой груза, поэтому величины c_{ij} могут иметь различный смысл в зависимости от конкретной задачи. Например, эти величины могут означать стоимость, расстояние, время, производительность и т.д.

Для примера рассмотрим задачу оптимальных назначений.

Пусть имеется m групп лиц A_i $(i = 1,2,\ldots,m)$ по a_i человек в каждой и n категорий работ B_j $(j = 1,2,\ldots,n)$ по b_j единиц в каждой. Известна производительность c_{ij} лица i-й группы при выполнении j-й категории работ. Необходимо определить, сколько лиц, из какой группы и на какую категорию работ назначить, чтобы суммарная производительность была максимальной при условии, что каждое лицо может быть назначено только на одну работу.

Если обозначить через x_{ij} количество лиц i-й группы, назначенных для выполнения работ j-й категории, то производительность при выполнении этих работ составит $c_{ij}x_{ij}$. А суммарную производительность можно выразить функцией $Z = \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij}$. Тогда задача имеет следующую математическую модель:

найти значения неизвестных, максимизирующих линейную функцию

$$Z = \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij}$$

и удовлетворяющих условиям

$$\sum_{j=1}^{n} x_{ij} = a_i \quad (i = 1,2,\ldots,m),$$

$$\sum_{i=1}^{m} x_{ij} = b_j \quad (j = 1,2,\ldots,n),$$

$$x_{ij} \ge 0 \quad (i = 1,2,\ldots,m; \;\; j = 1,2,\ldots,n).$$

В этой задаче общее число работ равно общему числу лиц, т.е. $\sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_j$.

Как и для других задач линейного программирования, процесс по отысканию оптимального решения транспортной задачи начинается с нахождения опорного решения. Рассмотрим систему ограничений (8.1) и (8.2) транспортной задачи. Она содержит mn неизвестных и $m+n$ уравнений, связанных соотношением (8.4). Если сложить почленно уравнения отдельно подсистемы (8.1) и отдельно подсистемы (8.2), то получим два одинаковых уравнения. В таблице 8.1 такое сложение равнозначно соответственно почленному сложению столбцов и почленному сложению строк.

Такая ситуация свидетельствует о линейной зависимости системы ограничений. Если одно из уравнений этой системы отбросить, то система ограничений должна содержать не более $m+n-1$ линейно независимых уравнений или столько же базисных переменных, следовательно, невырожденное опорное решение транспортной задачи содержит $m+n-1$ положительных компонент или перевозок.

Таким образом, если каким-либо способом получено невырожденное опорное решение транспортной задачи, то в матрице (x_{ij}) ($i=1,2,\ldots,m$; $j=1,2,\ldots,n$) значений его компонент положительными являются только $m+n-1$, а остальные равны нулю.

Если условия транспортной задачи и ее опорное решение записаны в виде таблицы (табл. 8.1), то клетки, в которых находятся отличные от нуля перевозки x_{ij}, называются *занятыми клетками*, остальные — *свободными клетками*. Занятые клетки соответствуют базисным переменным, и для невырожденного опорного решения их количество равно $m+n-1$.

Циклом называется набор клеток таблицы перевозок, в котором две и только две соседние клетки расположены в одном столбце или одной строке, причем последняя клетка находится в той же строке или столбце, что и первая. Построение цикла начинают с какой-либо занятой клетки и переходят по столбцу (строке) к другой занятой клетке, в которой делают поворот на

90° и движутся по строке (столбцу) к следующей занятой клетке и т.д., пытаясь возвратиться к первоначальной клетке.

Если из занятых допустимым решением транспортной задачи клеток нельзя образовать ни одного цикла, то такое решение является опорным решением. Если к клеткам, определяющим невырожденное опорное решение, добавить какую-либо свободную клетку, то образуется цикл, содержащий данную клетку и некоторую часть занятых клеток, и это решение перестает быть опорным. Таким образом, для любого опорного решения транспортной задачи должно быть не более $m+n-1$ занятых клеток, в противном случае это решение не является опорным.

Первоначальное опорное решение транспортной задачи как задачи линейного программирования можно построить рассмотренными в предыдущих темах методами, однако это сопряжено с большими вычислениями. Поэтому предложены несколько более простых методов построения первоначального опорного решения транспортной задачи.

Для нахождения первоначального опорного решения транспортной задачи удобно воспользоваться *методом северо-западного угла*, которое состоит в следующем.

Пусть условия транспортной задачи заданы в таблице 8.1. Будем заполнять эту таблицу, начиная с левого верхнего угла, условно называемого «северо-западным углом». Не учитывая стоимости перевозки единицы груза, начинаем удовлетворение потребностей первого потребителя B_1 за счет запаса первого поставщика A_1. Для этого записываем в левый нижний угол клетки A_1B_1 меньшее из чисел a_1 и b_1, т.е. $x_{11}=\min(a_1,b_1)$.

На первом шагу может быть два случая. Если $a_1 \geq b_1$, то $x_{11}=b_1$ и первый столбец «закрыт», т.е. спрос первого потребителя полностью удовлетворен. Это означает, что для всех остальных клеток первого столбца $x_{i1}=0$ $(i=2,\ldots,m)$. Двигаясь дальше по первой строке таблицы, переходим к удовлетворению потребностей потребителя B_2 за счет оставшегося запаса поставщика A_1. Здесь в клетку A_1B_2 записываем меньшее из чисел a_1-b_1 и b_2, т.е. $x_{12}=\min(a_1-b_1,b_2)$.

Как и выше, если $(a_1 - b_1) \geq b_2$, то $x_{12} = b_2$, второй столбец «закрыт», т.е. $x_{i2} = 0$ $(i = 2, \ldots, m)$, и теперь переходим к удовлетворению потребностей потребителя B_3. Если $(a_1 - b_1) < b_2$, то $x_{12} = a_1 - b_1$ и первая строка «закрыта», т.е. запасы первого поставщика полностью израсходованы. Это означает, что для всех остальных клеток первой строки $x_{1j} = 0$ $(j = 3, \ldots, n)$. В этом случае удовлетворение потребностей потребителя B_2 начинаем теперь за счет запаса поставщика A_2. Процесс аналогичным образом продолжаем дальше.

Если же на первом шагу $a_1 < b_1$, то $x_{11} = a_1$ и первая строка «закрыта», т.е. $x_{1j} = 0$ $(j = 2, \ldots, n)$, и теперь двигаясь дальше по первому столбцу таблицы, переходим к удовлетворению неудовлетворенных потребностей потребителя B_1 за счет запаса поставщика A_2. Здесь в клетку A_2B_1 записываем меньшее из чисел a_2 и $b_1 - a_1$, т.е. $x_{21} = \min(a_2, b_1 - a_1)$.

Таблица 8.2

Поставщики	Потребители					Запасы
	B_1	B_2	B_3	B_4	B_5	
A_1	10	7	4	1	4	100
A_2	2	7	10	6	11	250
A_3	8	5	3	2	2	200
A_4	11	8	12	16	13	300
Потребности	200	200	100	100	250	850

Этот процесс продолжаем до исчерпания всех запасов a_i $(i = 1, 2, \ldots, m)$ и удовлетворения всех потребностей b_j $(j = 1, 2, \ldots, n)$. Последняя заполненная клетка окажется лежащей в n-м столбце и в m-й строке. Начиная движение от клетки A_1B_1 только по занятым клеткам, невозможно вернуться не только в нее, но и в любую другую занятую клетку, т.е. нельзя построить цикл. Значит, решение, найденное методом северо-западного угла, является опорным решением.

Пример 8.1. Найти первоначальное опорное решение транспортной задачи, заданной в таблице 8.2, методом северо-западного угла.

Решение.

Таблица 8.3

Поставщики	Потребители					Запасы
	B_1	B_2	B_3	B_4	B_5	
A_1	10 100	7	4	1	4	100
A_2	2 100	7 150	10	6	11	250
A_3	8	5 50	3 100	2 50	2	200
A_4	11	8	12	16 50	13 250	300
Потребности	200	200	100	100	250	850

Найденное решение является невырожденным опорным решением, так как содержит точно $m+n-1=8$ занятых клеток.

Найдем общую стоимость плана перевозок как сумму произведений объемов перевозок, стоящих в левом нижнем углу занятых клеток, на соответствующие стоимости перевозки единицы груза в этих же клетках:

$$Z = 100\cdot 10 + 100\cdot 2 + 150\cdot 7 + 50\cdot 5 + 100\cdot 3 + 50\cdot 2 + 50\cdot 16 + \\ + 250\cdot 13 = 6950 \text{ единиц стоимости.}$$

Первоначальное опорное решение, найденное методом северо-западного угла, обычно оказывается весьма далеким от оптимального, так как при его нахождении не учитываются стоимости перевозки единицы груза c_{ij}. Поэтому в дальнейших расчетах потребуется много итераций для достижения оптимального решения.

Число этих итераций можно сократить, если первоначальное опорное решение найти *методом минимальной стоимости*. Сущность его состоит в том, что на каждом шаге осуществляется максимально возможное «перемещение» груза в клетку с минимальной стоимостью перевозки единицы груза c_{ij}.

Как и выше, пусть условия транспортной задачи заданы в таблице 8.1. Заполнение таблицы начинаем с клетки, в которой наименьшая величина затрат c_{ij}. Запишем в нее меньшее из чисел a_i и b_j. Затем либо строку, соответствующую поставщику, запасы которого полностью израсходованы, либо столбец, соответствующий потребителю, спрос которого полностью удовлетворен, либо и строку, и столбец, если израсходованы запасы поставщика и удовлетворены потребности потребителя, «вычеркивают», т.е. они больше не участвуют в процессе построения первоначального опорного решения. Далее из оставшихся клеток таблицы снова выбирают клетку с наименьшей величиной затрат и процесс распределения запасов продолжается до тех пор, пока они все не будут израсходованы, а все потребности удовлетворены.

Пример 8.2. Найти первоначальное опорное решение транспортной задачи, заданной в таблице 8.2, методом минимальной стоимости.

Решение.

Т а б л и ц а 8.4

Поставщики	Потребители					Запасы
	B_1	B_2	B_3	B_4	B_5	
A_1	10	7	4	1 100	4	100
A_2	2 200	7 50	10	6	11	250
A_3	8	5	3	2	2 200	200
A_4	11	8 150	12 100	16	13 50	300
Потребности	200	200	100	100	250	850

Это решение не содержит циклов и состоит из 7 занятых клеток, т.е. является вырожденным опорным решением.

Найдем общую стоимость плана перевозок:

$$Z = 100\cdot 1 + 200\cdot 2 + 50\cdot 7 + 200\cdot 2 + 150\cdot 8 + 100\cdot 12 + 50\cdot 13 =$$
$$= 4300 \text{ единиц стоимости.}$$

Ясно, что общая стоимость плана перевозок в примере 8.2 значительно меньше, чем в примере 8.1, следовательно, он ближе к оптимальному.

Если таблица стоимостей велика, то перебор всех элементов затруднителен. В этом случае используется *метод двойного предпочтения*, суть которого заключается в следующем.

В каждом столбце отмечается знаком $\vee$ клетка с наименьшей стоимостью перевозок. Затем то же проделывается в каждой строке. В результате некоторые клетки имеют отметку $\vee\vee$. В них находится минимальная стоимость, как по столбцу, так и по строке. В эти клетки помещаются максимально возможные объемы перевозок, и каждый раз соответствующие столбцы или строки «вычеркиваются». Затем перевозки распределяются по клеткам, отмеченным знаком $\vee$. В оставшейся части таблицы перевозки распределяются по наименьшей стоимости.

Пример 8.3. Найти первоначальное опорное решение транспортной задачи, заданной в таблице 8.2, методом двойного предпочтения.

Решение.

Таблица 8.5

Поставщики	Потребители					Запасы
	B_1	B_2	B_3	B_4	B_5	
A_1	10	7	4	$\vee\vee$ 1 100	4	100
A_2	$\vee\vee$ 2 200	7	10 50	6	11	250
A_3	8	$\vee$ 5	$\vee$ 3	$\vee$ 2	$\vee\vee$ 2 200	200
A_4	11	$\vee$ 8 200	12 50	16	13 50	300
Потребности	200	200	100	100	250	850

Это решение также не содержит циклов и состоит из 7 занятых клеток, т.е. является вырожденным опорным решением.

Найдем общую стоимость плана перевозок:

$$Z = 100\cdot 1 + 200\cdot 2 + 50\cdot 10 + 200\cdot 2 + 200\cdot 8 + 50\cdot 12 + 50\cdot 13 =$$
$$= 4250 \text{ единиц стоимости.}$$

Вопросы для повторения и контроля :

1. Что такое транспортная задача и каково ее значение?
2. Какова постановка транспортной задачи?
3. В чем состоит математическая модель транспортной задачи?
4. Что вы знаете о закрытой модели транспортной задачи и о теореме 8.1?
5. Что такое открытая модель транспортной задачи и как она решается?
6. Что вы знаете о задаче оптимальных назначений?
7. Как представляется опорное решение транспортной задачи в таблице, что такое свободные клетки и занятые клетки?
8. Что такое цикл в таблице транспортной задачи, как он связан с опорным решением?
9. В чем сущность метода северо-западного угла?
10. В чем сущность метода минимальной стоимости?
11. В чем сущность метода двойного предпочтения?

Опорные слова :

Транспортная задача, математическая модель транспортной задачи, закрытая модель транспортной задачи, открытая модель транспортной задачи, задача оптимальных назначений, свободные клетки, занятые клетки, цикл в таблице транспортной задачи, метод северо-западного угла, метод минимальной стоимости, метод двойного предпочтения.

Тема № 9

Метод потенциалов для нахождения оптимального решения транспортной задачи

План :

4. Построение системы потенциалов.
5. Проверка выполнения условия оптимальности для свободных клеток.
6. Переход к новому опорному решению.

С помощью рассмотренных в прошлой теме методов можно получить первоначальное вырожденное или невырожденное опорное решение транспортной задачи. Это опорное решение как решение задачи линейного программирования можно было бы довести до оптимального с помощью симплексного метода. Однако из-за громоздкости симплексных таблиц, содержащих mn неизвестных, и большего объема вычислительных работ для получения оптимального решения транспортной задачи используются более простые методы. Самым распространенным из этих методов считается *метод потенциалов* (модифицированный распределительный метод).

С помощью метода потенциалов после конечного числа шагов находится оптимальное решение транспортной задачи путем перехода на каждом этапе к более близким к оптимальному решению опорным решениям, начиная с первоначального опорного решения. Для проверки на оптимальность найденного на каждом этапе опорного решения каждому поставщику A_i и каждому потребителю B_j ставятся в соответствие числа u_i и v_j $(i=1,2,\dots,m;\quad j=1,2,\dots,n)$, называемые соответственно их *потенциалами*. Соотношения между этими потенциалами устанавливаются с помощью следующей теоремы.

Теорема 9.1. *Если решение* $\mathbf{X}^*=(x_{ij}^*)$ *транспортной задачи является оптимальным, то ему соответствуют* $m+n$ *чисел* u_i^* *и* v_j^*, *удовлетворяющих условиям:*

$$u_i^* + v_j^* = c_{ij} \quad \text{для} \quad x_{ij}^* > 0,$$
$$u_i^* + v_j^* \le c_{ij} \quad \text{для} \quad x_{ij}^* = 0$$
$$(i = 1,2,\ldots,m;\ j = 1,2,\ldots,n).$$

Эту теорему примем без доказательства.

На основании теоремы 9.1 для того, чтобы опорное решение транспортной задачи было оптимальным, необходимо выполнение следующих условий:

а) для каждой занятой клетки сумма потенциалов должна быть равна стоимости перевозки единицы груза, стоящей в этой клетке:

$$u_i + v_j = c_{ij}\ ; \tag{9.1}$$

б) для каждой свободной клетки сумма потенциалов должна быть меньше или равна стоимости перевозки единицы груза, стоящей в этой клетке:

$$u_i + v_j \le c_{ij}\ . \tag{9.2}$$

Если хотя бы одна свободная клетка не удовлетворяет условию (9.2), то найденное опорное решение не является оптимальным решением и его можно улучшить. Для проверки решения на оптимальность необходимо сначала построить систему потенциалов.

Систему потенциалов можно построить только для невырожденного опорного решения. Такое решение содержит $m+n-1$ занятых клеток, поэтому для него можно составить систему из $m+n-1$ линейно независимых уравнений вида (9.1) с $m+n$ неизвестными. В этой системе уравнений на одно меньше, чем неизвестных, поэтому придав одному из потенциалов нулевое значение, можно однозначно определить остальные. Затем необходимо проверить правильность построения системы потенциалов, для чего следует проверить выполнения условия (9.1) для всех занятых клеток.

Если опорное решение является вырожденным, т.е. содержит менее $m+n-1$ занятых клеток, то для построения системы потенциалов количество занятых клеток дополняется до $m+n-1$ путем ввода в нужное количество свободных клеток нулевых перевозок таким образом, чтобы не образовалось ни одного цикла. Клетки, в которые введены нулевые перевозки, называются *фиктивно занятыми клетками*. Так как в транспортной задаче требуется нахождение минимального значения целевой функции, то целесообразно сделать фиктивно занятыми клетки, в которых стоит по мере возможности наименьшая стоимость.

После того, как определены числовые значения всех потенциалов, для всех свободных клеток вычисляются величины

$$\Delta_{ij} = u_i + v_j - c_{ij}. \tag{9.3}$$

Если для всех i и j $(i = 1,2,\ldots,m; \; j = 1,2,\ldots,n)$ имеет место $\Delta_{ij} \le 0$, то найденное опорное решение будет оптимальным.

Если же, по крайней мере, для одной пары *i* и *j* имеет место $\Delta_{ij} > 0$, то найденное опорное решение не будет оптимальным, и оно заменяется на новое опорное решение. Для этого следует загрузить свободную клетку $A_k B_l$ (ввести в базис переменную x_{kl}), удовлетворяющую условию

$$\Delta_{kl} = \max_{\Delta_{ij} > 0} \Delta_{ij}. \tag{9.4}$$

С этой целью строится цикл, вершины которого лежат в клетке $A_k B_l$ и в части занятых клеток. В клетку $A_k B_l$ ставится знак «+», а в остальные вершины цикла — поочередно знаки «–» и «+».

Наименьшая перевозка в вершинах этого цикла, отмеченных знаком «–» (в клетке $A_r B_s$), обозначается через θ и эта величина «перемещается» по клеткам цикла, т.е. величина θ вычитается из объемов перевозок в клетках, отмеченных знаком «–», и прибавляется к объемам перевозок в клетках, отмеченных знаком «+». В частности, объем перевозок в клетке $A_k B_l$ становится равным θ, а в клетке $A_r B_s$ — нулю. В результате клетка $A_r B_s$ становится свободной клеткой (если после перемещения перевозок в нескольких занятых клетках образуются нулевые перевозки, то из этих клеток только одна превращается в свободную клетку), а клетка $A_k B_l$ — занятой клеткой и получается новое опорное решение.

Для полученного нового опорного решения также определяются числовые значения потенциалов и опять для свободных клеток вычисляются величины вида (9.3). Этот процесс продолжается до тех пор, пока среди Δ_{ij} не останется ни одного положительного, и решение, найденное на последнем этапе, будет оптимальным решением.

Пример 9.1. Найти методом потенциалов оптимальное решение транспортной задачи, заданной в таблице 8.2.

<u>Решение</u>. Пусть первоначальное опорное решение данной транспортной задачи найдено методом минимальной стоимости, тогда прибавив к таблице 8.4 по одной строке и столбцу для записи потенциалов, получим следующую таблицу:

Т а б л и ц а 9.1

Поставщики	Потребители						Запасы
	u_i	B_1	B_2	B_3	B_4	B_5	
v_j		$v_1 =$ 3	$v_2 =$ 8	$v_3 =$ 12	$v_4 =$ 13	$v_5 =$ 13	
A_1	$u_1 =$ –12	10	7	4	1 100	4	100
A_2	$u_2 =$ –1	2 200	7 50	10	6	11	250
A_3	$u_3 =$ –11	8	5	3	2 **0**	2 200	200
A_4	$u_4 =$ 0	11	8 150	12 100	16	13 50	300
Потребности		200	200	100	100	250	850

Найденное первоначальное опорное решение является вырожденным опорным решением, так как число занятых клеток равно 7 и на одно меньше, чем $m+n-1=8$. Поэтому, вследствие того, что среди свободных клеток наименьшая стоимость перевозки единицы груза стоит в клетке A_3B_4 и с участием этой клетки невозможно построить ни одного цикла, в эту клетку вводится нулевая перевозка (клетка A_3B_4 делается фиктивно занятой).

Выберем строку, содержащую наибольшее количество занятых клеток, т.е. строку A_4, и для нее положим $u_4=0$. В строке A_4 три занятые клетки A_4B_2, A_4B_3 и A_4B_5 связывают потенциал u_4 соответственно с потенциалами v_2, v_3 и v_5. Определим эти потенциалы по формуле (9.1): $v_2=c_{42}-u_4=8-0=8$; $v_3=c_{43}-u_4=12-0=12$; $v_5=c_{45}-u_4=13-0=13$. Остальные потенциалы невозможно определить с помощью потенциала u_4. Теперь, поочередно просматривая столбцы B_2, B_3 и B_5, из занятых в них клетках с помощью соответственно потенциалов v_2, v_3 и v_5 определим потециалы $u_2 = 7 - 8 = -1$ и $u_3 = 2 - 13 = -11$. Остальные потенциалы найдем подобным образом и запишем в таблицу 9.1.

Затем для всех свободных клеток по формуле (9.3) определим знаки Δ_{ij}. Например, для строки A_1 $\Delta_{11} = -12 + 3 - 10 = -19$; $\Delta_{12} = -12 + 8 - 7 = -11$; $\Delta_{13} = -12 + 12 - 4 = -4$; $\Delta_{15} = -12 + 13 - 4 = -3$.

Для трех свободных клеток Δ_{ij} будут положительными, т.е. $\Delta_{23} = 1$, $\Delta_{24} = 6$ и $\Delta_{25} = 1$. Значит, найденное первоначальное опорное решение не является оптимальным. При переходе к новому опорному решению вновь загружаемая клетка находится из условия (9.4). В данном случае такой является клетка A_2B_4.

Теперь строится цикл с вершинами в клетке A_2B_4, а также в занятых клетках A_3B_4, A_3B_5, A_4B_5, A_4B_2 и A_2B_2; в клетку A_2B_4 ставится знак «+», а в остальные вершины цикла — поочередно знаки «–» и «+» (табл. 9.2).

Т а б л и ц а 9.2

Поставщики	Потребители						Запасы
	u_i	B_1	B_2	B_3	B_4	B_5	
v_j		$v_1 =$ 3	$v_2 =$ 8	$v_3 =$ 12	$v_4 =$ 13	$v_5 =$ 13	
A_1	$u_1 =$ –12	10	7	4	1 100	4	100
A_2	$u_2 =$ –1	2 200	– 7 50	10	+ 6	11	250
A_3	$u_3 =$ –11	8	5	3	– 2 0	+ 2 200	200
A_4	$u_4 =$ 0	11	+ 8 150	12 100	16	– 13 50	300
Потребности		200	200	100	100	250	850

Находится наименьшая перевозка θ в вершинах этого цикла, отмеченных знаком «–» (клетки A_3B_4, A_4B_5 и A_2B_2): $\theta = \min(0; 50; 50) = 0$. Эта величина вычитается из объемов перевозок в клетках, отмеченных знаком «–», и прибавляется к объемам перевозок в клетках, отмеченных знаком «+» (клетки A_2B_4, A_3B_5 и A_4B_2). Затем изменяются значения потенциалов (табл. 9.3).

Таблица 9.3

Поставщики	Потребители						Запасы
	u_i	B_1	B_2	B_3	B_4	B_5	
v_j		$v_1 =$ 3	$v_2 =$ 8	$v_3 =$ 12	$v_4 =$ 7	$v_5 =$ 13	
A_1	$u_1 =$ –6	10	7	4	1 100	4	100
A_2	$u_2 =$ –1	2 200	7 50	10	6 0	11	250
A_3	$u_3 =$ –11	8	5	3	2	2 200	200
A_4	$u_4 =$ 0	11	8 150	12 100	16	13 50	300
Потребности		200	200	100	100	250	850

Теперь для всех свободных клеток по формуле (9.3) определим знаки Δ_{ij}. Для четырех свободных клеток Δ_{ij} будут положительными, т.е. $\Delta_{13} = 2$, $\Delta_{15} = 3$, $\Delta_{23} = 1$ и $\Delta_{25} = 1$. Значит, найденное первоначальное опорное решение не является оптимальным. При переходе к новому опорному решению вновь загружаемая клетка находится из условия (9.4). В данном случае такой является клетка A_1B_5.

Затем строится цикл с вершинами в клетке A_1B_5, а также в занятых клетках A_4B_5, A_4B_2, A_2B_2, A_2B_4 и A_1B_4; в клетку A_1B_5 ставится знак «+», а в остальные вершины цикла — поочередно знаки «–» и «+» (табл. 9.4).

Находится наименьшая перевозка θ в вершинах этого цикла, отмеченных знаком «–» (клетки A_4B_5, A_2B_2 и A_1B_4): $\theta = \min(50; 50; 100) = 50$. Эта величина вычитается из объемов перевозок в клетках, отмеченных знаком «–», и прибавляется к объемам перевозок в клетках, отмеченных знаком «+» (клетки A_1B_5, A_4B_2 и A_2B_4). Затем изменяются значения потенциалов (табл. 9.5).

Т а б л и ц а 9.4

Поставщики	Потребители						Запасы
	u_i	B_1	B_2	B_3	B_4	B_5	
v_j		$v_1 =$ 3	$v_2 =$ 8	$v_3 =$ 12	$v_4 =$ 7	$v_5 =$ 13	
A_1	$u_1 =$ –6	10	7	4	– 1 100	+ 4	100
A_2	$u_2 =$ –1	2 200	– 7 50	10	+ 6 0	11	250
A_3	$u_3 =$ –11	8	5	3	2	2 200	200
A_4	$u_4 =$ 0	11	+ 8 150	12 100	16	– 13 50	300
Потребности		200	200	100	100	250	850

Т а б л и ц а 9.5

Поставщики	Потребители						Запасы
	u_i	B_1	B_2	B_3	B_4	B_5	
v_j		$v_1 =$ 3	$v_2 =$ 8	$v_3 =$ 12	$v_4 =$ 7	$v_5 =$ 10	
A_1	$u_1 =$ –6	10	7	4	1 50	4 50	100
A_2	$u_2 =$ –1	2 200	7 0	10	6 50	11	250
A_3	$u_3 =$ –8	8	5	3	2	2 200	200
A_4	$u_4 =$ 0	11	8 200	12 100	16	13	300
Потребности		200	200	100	100	250	850

Теперь для всех свободных клеток по формуле (9.3) определим знаки Δ_{ij}. Для трех свободных клеток Δ_{ij} будут положительными, т.е. $\Delta_{13}=2$, $\Delta_{23}=1$ и $\Delta_{33}=1$. Значит, найденное первоначальное опорное решение не является оптимальным. При переходе к новому опорному решению вновь загружаемая клетка находится из условия (9.4). В данном случае такой является клетка A_1B_3.

Затем строится цикл с вершинами в клетке A_1B_3, а также в занятых клетках A_1B_4, A_2B_4, A_2B_2, A_4B_2 и A_4B_3; в клетку A_1B_3 ставится знак «+», а в остальные вершины цикла — поочередно знаки «–» и «+» (табл. 9.6).

Таблица 9.6

Поставщики	Потребители						Запасы
	u_i	B_1	B_2	B_3	B_4	B_5	
v_j		$v_1=3$	$v_2=8$	$v_3=12$	$v_4=7$	$v_5=10$	
A_1	$u_1=-6$	10	7	+ 4	– 1 50	4 50	100
A_2	$u_2=-1$	2 200	– 7 0	10	+ 6 50	11	250
A_3	$u_3=-8$	8	5	3	2	2 200	200
A_4	$u_4=0$	11	+ 8 200	– 12 100	16	13	300
Потребности		200	200	100	100	250	850

Находится наименьшая перевозка θ в вершинах этого цикла, отмеченных знаком «–» (клетки A_1B_4, A_2B_2 и A_4B_3): $\theta=\min(50;0;100)=0$. Эта величина вычитается из объемов перевозок в клетках, отмеченных знаком «–», и прибавляется к объемам перевозок в клетках, отмеченных знаком «+» (клетки A_2B_4, A_4B_2 и A_1B_3). Затем изменяются значения потенциалов (табл. 9.7).

Затем для всех свободных клеток по формуле (9.3) определим знаки Δ_{ij}. Для всех свободных клеток Δ_{ij} будут неположительными. Значит, найденное первоначальное опорное решение является оптимальным. Найдем общую стоимость плана перевозок:

$$Z = 50 \cdot 1 + 50 \cdot 4 + 200 \cdot 2 + 50 \cdot 6 + 200 \cdot 2 + 200 \cdot 8 + 100 \cdot 12 =$$
$$= 4150 \text{ единиц стоимости.}$$

Т а б л и ц а 9.7

Поставщики	Потребители						Запасы
	u_i	B_1	B_2	B_3	B_4	B_5	
v_j		$v_1 = 3$	$v_2 = 6$	$v_3 = 10$	$v_4 = 7$	$v_5 = 10$	
A_1	$u_1 = -6$	10	7	4 0	1 50	4 50	100
A_2	$u_2 = -1$	2 200	7	10	6 50	11	250
A_3	$u_3 = -8$	8	5	3	2	2 200	200
A_4	$u_4 = 2$	11	8 200	12 100	16	13	300
Потребности		200	200	100	100	250	850

Вопросы для повторения и контроля :

1. С какой целью применяется метод потенциалов?
2. Что такое потенциалы, какая связь существует между ними, что вы знаете о теореме 9.1?
3. В чем сущность метода потенциалов?
4. Как строится система потенциалов, что называется фиктивно занятыми клетками?
5. Как переходить к новому опорному решению в методе потенциалов, как строится цикл?

Опорные слова :

Метод потенциалов, потенциалы, фиктивно занятые клетки, переход к новому опорному решению, цикл при переходе к новому опорному решению.

Тема № 10

Задача нелинейного программирования, ее геометрическая интерпретация. Метод множителей Лагранжа

План :

1. Постановка, виды и геометрическая интерпретация задачи нелинейного программирования.
2. Метод множителей Лагранжа.

Возникновение нелинейного программирования связано с тем, что предположение о линейной зависимости различных характеристик рассматриваемого процесса от планируемых параметров при исследовании многих практических задач, в том числе экономических, оказывается весьма приблизительным. Например, различного рода затраты труда и материальных ресурсов в большинстве случаев являются нелинейными функциями от точности, надежности и других характеристик качества выпускаемых продуктов.

Задача нелинейного программирования формулируется следующим образом:

найти значения неизвестных, доставляющих экстремум функции

$$Z = f(x_1, x_2, \ldots, x_n), \tag{10.1}$$

и удовлетворяющих условиям

$$g_i(x_1, x_2, \ldots, x_n)\{\leq, =, \geq\} b_i \qquad (i = 1,2,\ldots,m), \tag{10.2}$$

где по крайней мере, одна из функций f и g_i нелинейная, а на переменные $x_1, x_2, \ldots, x_n$ накладываются условия неотрицательности.

Класс задач нелинейного программирования значительно шире класса задач линейного программирования. Нелинейное программирование охватывает широкий класс настолько сложных задач, что до сих пор невозможно разработать общие методы, подобные симплексному методу в линейном программировании, которые позволяли бы решать любые задачи нелинейного программирования.

Трудность решения нелинейных задач состоит прежде всего в том, что область допустимых решений здесь в отличие от задач линейного программирования может оказаться невыпуклой или иметь бесконечное число угловых точек, а возможно и то и другое одновременно. При нелинейности целевой функции возникают другие затруднения: оптимальное значение может достигаться не только на границе, но и внутри допустимой области.

Основные результаты в нелинейном программировании получены при рассмотрении задач, в которых система ограничений линейная, а целевая функция нелинейная. Даже в таких задачах оптимальное решение может быть найдено только для узкого класса целевых функций.

Если и целевая функция (10.1), и функции g_i в ограничениях (10.2) задачи нелинейного программирования являются выпуклыми или вогнутыми функциями, то эта задача становится *задачей выпуклого программирования*.

Если же целевая функция (10.1) является суммой линейной и квадратичной функций, а функции g_i в ограничениях (10.2) задачи нелинейного программирования являются линейными функциями, то эта задача становится *задачей квадратичного программирования*.

Теперь познакомимся с классическим методом решения задачи нелинейного программирования, а именно с методом множителей Лагранжа.

Пусть задана задача математического программирования:

найти экстремум функции

$$Z = f(x_1, x_2, \ldots, x_n)$$

при ограничениях

$$g_i(x_1, x_2, \ldots, x_n) = 0 \qquad (i = 1, 2, \ldots, m).$$

При этом предполагается, что функции $f(x_1, x_2, \ldots, x_n)$ и $g_i(x_1, x_2, \ldots, x_n)$ $(i = 1, 2, \ldots, m)$ непрерывны вместе со своими первыми частными производными.

Для решения задачи составим функцию

$$F(x_1, x_2, \ldots, x_n, \lambda_1, \lambda_2, \ldots, \lambda_m) =$$
$$= f(x_1, x_2, \ldots, x_n) + \sum_{i=1}^{m} \lambda_i g_i(x_1, x_2, \ldots, x_n), \qquad (10.3)$$

определим частные производные $\frac{\partial F}{\partial x_j}$ $(j=1,2,\ldots,n)$, $\frac{\partial F}{\partial \lambda_i}$ $(i=1,2,\ldots,m)$ и приравняем их нулю. В результате получим систему уравнений

$$\begin{cases} \frac{\partial F}{\partial x_j} = \frac{\partial f}{\partial x_j} + \sum_{i=1}^{m} \lambda_i \frac{\partial g_i}{\partial x_j} = 0, \quad j=1,2,\ldots,n \\ \frac{\partial F}{\partial \lambda_i} = g_i(x_1, x_2, \ldots, x_n) = 0, \quad i=1,2,\ldots,m \end{cases}. \tag{10.4}$$

Функция (10.3) называется *функцией Лагранжа*, а числа λ_i — *множителями Лагранжа*.

Если функция $Z = f(x_1, x_2, \ldots, x_n)$ в точке $\mathbf{X}^{(0)} = \left(x_1^{(0)}, x_2^{(0)}, \ldots, x_n^{(0)}\right)$ имеет экстремум, то существует такой вектор $\mathbf{L}^{(0)} = \left(\lambda_1^{(0)}, \lambda_2^{(0)}, \ldots, \lambda_m^{(0)}\right)$, что точка $\left(x_1^{(0)}, x_2^{(0)}, \ldots, x_n^{(0)}, \lambda_1^{(0)}, \lambda_2^{(0)}, \ldots, \lambda_m^{(0)}\right)$ является решением системы (10.4). Следовательно, решая систему (10.4), получаем множество точек, в которых целевая функция может иметь экстремальные значения.

Если в ограничениях (10.2) $m=0$, т.е. нет никаких ограничений на переменные, за исключением, может быть, условий неотрицательности, то задача (10.1) – (10.2) становится обычной задачей нахождения безусловного экстремума функции или задачей безусловной оптимизации. При этом функция Лагранжа F совпадает с функцией Z.

Пример 10.1. Найти точку условного экстремума функции $Z = x_1 x_2 + x_2 x_3$ при ограничениях

$$\begin{cases} x_1 + x_2 = 2 \\ x_2 + x_3 = 2 \end{cases}.$$

Решение. Составим функцию Лагранжа $F(x_1, x_2, x_3, \lambda_1, \lambda_2) = x_1 x_2 + x_2 x_3 + \lambda_1(x_1 + x_2 - 2) + \lambda_2(x_2 + x_3 - 2)$ и продифференцируем ее по переменным x_1, x_2, x_3, λ_1 и λ_2. Приравнивая полученные выражения нулю, получаем следующую систему уравнений:

$$\begin{cases} x_2 + \lambda_1 = 0 \\ x_1 + x_3 + \lambda_1 + \lambda_2 = 0 \\ x_2 + \lambda_2 = 0. \\ x_1 + x_2 - 2 = 0 \\ x_2 + x_3 - 2 = 0 \end{cases}$$

Из первого и третьего уравнения следует, что $\lambda_1 = \lambda_2 = -x_2$. Тогда

$$\begin{cases} x_1 - 2x_2 + x_3 = 0 \\ x_1 + x_2 = 2 \\ x_2 + x_3 = 2 \end{cases}.$$

Решая данную систему, находим: $x_1 = x_2 = x_3 = 1,\ Z = 2$.

Вопросы для повторения и контроля :

1. С чем связано возникновение нелинейного программирования и как формулируется задача нелинейного программирования?
2. Какие сложности возникают при решении задач нелинейного программирования?
3. Какие виды задач нелинейного программирования вы знаете?
4. Что такое функция Лагранжа и что такое множители Лагранжа?
5. Что вы знаете о методе множителей Лагранжа?

Опорные слова :

Задача нелинейного программирования, задача выпуклого программирования, задача квадратичного программирования, функция Лагранжа, множители Лагранжа, метод множителей Лагранжа.

Тема № 11

Выпуклое программирование. Условия Куна–Таккера. Теорема Куна–Таккера

План :

1. Выпуклые и вогнутые функции.
2. Свойства выпуклых и вогнутых функций.
3. Теорема Куна–Таккера.
4. Условия Куна–Таккера.

Функция $f(\mathbf{X})$, определенная на выпуклом множестве C, называется *выпуклой (выпуклой книзу) функцией*, если для любых точек $\mathbf{X}^{(1)}$ и $\mathbf{X}^{(2)}$ из этого множества и любого $0 \le \lambda \le 1$ справедливо соотношение

$$f((1-\lambda)\mathbf{X}^{(1)} + \lambda\mathbf{X}^{(2)}) \le (1-\lambda) f(\mathbf{X}^{(1)}) + \lambda f(\mathbf{X}^{(2)}). \quad (11.1)$$

Если неравенство (11.1) при $0 < \lambda < 1$ является строгим для любых $\mathbf{X}^{(1)} \in C$, $\mathbf{X}^{(2)} \in C$, $\mathbf{X}^{(1)} \not\equiv \mathbf{X}^{(2)}$, то функция $f(\mathbf{X})$ называется *строго выпуклой функцией*.

Функция $f(\mathbf{X})$, определенная на выпуклом множестве C, называется *вогнутой (выпуклой кверху) функцией*, если для любых точек $\mathbf{X}^{(1)}$ и $\mathbf{X}^{(2)}$ из этого множества и любого $0 \le \lambda \le 1$ справедливо соотношение

$$f((1-\lambda)\mathbf{X}^{(1)} + \lambda\mathbf{X}^{(2)}) \ge (1-\lambda) f(\mathbf{X}^{(1)}) + \lambda f(\mathbf{X}^{(2)}). \quad (11.2)$$

Если неравенство (11.2) при $0 < \lambda < 1$ является строгим для любых $\mathbf{X}^{(1)} \in C$, $\mathbf{X}^{(2)} \in C$, $\mathbf{X}^{(1)} \not\equiv \mathbf{X}^{(2)}$, то функция $f(\mathbf{X})$ называется *строго вогнутой функцией*.

Из этих определений ясно, что если функция $f(\mathbf{X})$ — выпуклая функция, то функция $-f(\mathbf{X})$ — вогнутая функция.

Например, функция $f(x) = x^2$ на числовой оси $(-\infty;\ \infty)$ является выпуклой функцией, а функция $f(x) = -x^2$ — вогнутой функцией. Функция $f(x) = x^3$ является вогнутой функцией на $(-\infty;\ 0)$ и выпуклой функцией на $(0;\ \infty)$.

Нетрудно показать, что неотрицательная линейная комбинация $\sum_{j=1}^{k}\mu_j f_j(\mathbf{X})$ при $\mu_j \ge 0$ функций $f_j(\mathbf{X})$ ($j = 1,2,\ldots,k$), являющихся выпуклыми на некотором выпуклом множестве *C*, является выпуклой функцией. В частности, выпуклой является и сумма $\sum_{j=1}^{k} f_j(\mathbf{X})$ этих функций. Аналогичное утверждение справедливо и для вогнутых функций.

Теорема 11.1. *Если* $f(\mathbf{X})$ *— выпуклая функция при всех* $\mathbf{X} \ge 0$*, то будет выпуклым и множество V всех точек, удовлетворяющих условиям* $f(\mathbf{X}) \le b$, $\mathbf{X} \ge 0$.

Доказательство. Покажем, что множество *V* вместе с точками $\mathbf{X}^{(1)}$ и $\mathbf{X}^{(2)}$ при любом $0 \le \lambda \le 1$ содержит точку $\mathbf{X}^{(0)} = (1-\lambda)\mathbf{X}^{(1)} + \lambda\mathbf{X}^{(2)}$. Из условий $\mathbf{X}^{(1)} \ge 0$ и $\mathbf{X}^{(2)} \ge 0$ вытекает $\mathbf{X}^{(0)} \ge 0$.

Так как $f(\mathbf{X})$ — выпуклая функция, то

$$f(\mathbf{X}^{(0)}) = f((1-\lambda)\mathbf{X}^{(1)} + \lambda\mathbf{X}^{(2)}) \le (1-\lambda) f(\mathbf{X}^{(1)}) + \lambda f(\mathbf{X}^{(2)}).$$

Для точек $\mathbf{X}^{(1)}$ и $\mathbf{X}^{(2)}$ выполняются условия $f(\mathbf{X}^{(1)}) \le b$ и $f(\mathbf{X}^{(2)}) \le b$. Тогда для всех $0 \le \lambda \le 1$ имеют место соотношения

$$(1-\lambda) f(\mathbf{X}^{(1)}) \le (1-\lambda) b \ \text{ и } \ \lambda f(\mathbf{X}^{(2)}) \le \lambda b.$$

Следовательно,

$$f(\mathbf{X}^{(0)}) \le (1-\lambda) f(\mathbf{X}^{(1)}) + \lambda f(\mathbf{X}^{(2)}) \le (1-\lambda) b + \lambda b = b.$$

Значит, точка $\mathbf{X}^{(0)}$ принадлежит множеству *V*. Доказательство теоремы завершено.

Аналогичная теорема доказывается и для вогнутых функций.

В теории выпуклого программирования в качестве основной обычно рассматривается задача минимизации выпуклой функции

$$Z = f(x_1, x_2, \ldots, x_n), \tag{11.3}$$

при условиях

$$g_i(x_1, x_2, \ldots, x_n) \le 0 \qquad (i = 1,2,\ldots,m), \tag{11.4}$$

$$x_1, x_2, \ldots, x_n \ge 0, \tag{11.5}$$

где функции $g_i(x_1, x_2, \ldots, x_n)$ предполагаются выпуклыми. Выпуклость множества допустимых решений этой задачи вытекает из теоремы 11.1.

Если $f(x_1, x_2, \ldots, x_n)$ и $g_i(x_1, x_2, \ldots, x_n)$ являются вогнутыми функциями, то имеем задачу максимизации $Z = f(x_1, x_2, \ldots, x_n)$ при ограничениях $g_i(x_1, x_2, \ldots, x_n) \geq 0$ и $x_1, x_2, \ldots, x_n \geq 0$.

Составим функцию Лагранжа для данной задачи:

$$F(\mathbf{X}, \mathbf{L}) = F(x_1, x_2, \ldots, x_n, \lambda_1, \lambda_2, \ldots, \lambda_m) = f(x_1, x_2, \ldots, x_n) +$$

$$+ \sum_{i=1}^{m} \lambda_i g_i(x_1, x_2, \ldots, x_n) = f(\mathbf{X}) + \sum_{i=1}^{m} \lambda_i g_i(\mathbf{X}). \qquad (11.6)$$

Точка $\left(\mathbf{X}^{(0)}, \mathbf{L}^{(0)}\right) = \left(x_1^{(0)}, x_2^{(0)}, \ldots, x_n^{(0)}, \lambda_1^{(0)}, \lambda_2^{(0)}, \ldots, \lambda_m^{(0)}\right)$ называется *седловой точкой* функции (11.6), если точка $\mathbf{X}^{(0)} = \left(x_1^{(0)}, x_2^{(0)}, \ldots, x_n^{(0)}\right)$ является точкой минимума функции $F(\mathbf{X}, \mathbf{L}^{(0)})$, а точка $\mathbf{L}^{(0)} = \left(\lambda_1^{(0)}, \lambda_2^{(0)}, \ldots, \lambda_m^{(0)}\right)$ — точкой максимума функции $F(\mathbf{X}^{(0)}, \mathbf{L})$. Другими словами, для седловой точки при всех $\mathbf{X}$ и $\mathbf{L}$ выполняется соотношение

$$F(\mathbf{X}^{(0)}, \mathbf{L}) \leq F(\mathbf{X}^{(0)}, \mathbf{L}^{(0)}) \leq F(\mathbf{X}, \mathbf{L}^{(0)}). \qquad (11.7)$$

Теорема 11.2 (теорема Куна–Таккера). *Пусть существует по крайней мере одна точка* $\mathbf{X} \geq 0$*, для которой* $g_i(\mathbf{X}) \leq 0$ $(i = 1,2,\ldots,m)$. *Тогда необходимым и достаточным условием оптимальности вектора* $\mathbf{X}^{(0)}$*, принадлежащего области допустимых решений задачи (11.3) – (11.5), является существование такого вектора* $\mathbf{L}^{(0)}$*, что для всех* $\mathbf{X} \geq 0$ *и* $\mathbf{L} \geq 0$ *имеют место неравенства (11.7)*.

Эту теорему примем без доказательства.

Теорема Куна–Таккера называется также *теоремой о седловой точке*.

Если $f(\mathbf{X})$ и $g_i(\mathbf{X})$ — дифференцируемые функции, то неравенства в (11.7) эквивалентны следующим локальным условиям Куна–Таккера:

$$\left(\frac{\partial F}{\partial x_j}\right)_{\mathbf{X}^{(0)}, \mathbf{L}^{(0)}} \geq 0, \qquad x_j^{(0)} \left(\frac{\partial F}{\partial x_j}\right)_{\mathbf{X}^{(0)}, \mathbf{L}^{(0)}} = 0,$$

$$x_j^{(0)} \geq 0, \qquad (j = 1,2,\ldots,n); \tag{11.8}$$

$$\left(\frac{\partial F}{\partial \lambda_i}\right)_{\mathbf{X}^{(0)},\mathbf{L}^{(0)}} \leq 0, \qquad \lambda_i^{(0)}\left(\frac{\partial F}{\partial \lambda_i}\right)_{\mathbf{X}^{(0)},\mathbf{L}^{(0)}} = 0,$$

$$\lambda_i^{(0)} \geq 0, \qquad (i = 1,2,\ldots,m). \tag{11.9}$$

Выражения $\left(\frac{\partial F}{\partial x_j}\right)_{\mathbf{X}^{(0)},\mathbf{L}^{(0)}}$ и $\left(\frac{\partial F}{\partial \lambda_i}\right)_{\mathbf{X}^{(0)},\mathbf{L}^{(0)}}$ означают, что значения частных производных функции Лагранжа берутся в точке $\left(\mathbf{X}^{(0)},\mathbf{L}^{(0)}\right)$.

Вопросы для повторения и контроля :

1. Что такое выпуклая функция и строго выпуклая функция?
2. Что такое вогнутая функция и строго вогнутая функция?
3. Что вы знаете о свойствах выпуклых и вогнутых функций, а также о теореме 11.1?
4. Какая задача обычно рассматривается в качестве основной в теории выпуклого программирования?
5. Что вы знаете о седловой точке?
6. О чем идет речь в теореме Куна–Таккера и что такое условия Куна–Таккера?

Опорные слова :

Выпуклая функция, строго выпуклая функция, вогнутая функция, строго вогнутая функция, задача выпуклого программирования, седловая точка, теорема Куна–Таккера, условия Куна–Таккера.

Тема № 12

Квадратичное программирование. Методы решения задач квадратичного программирования

План :

1. Постановка задачи квадратичного программирования и условия Куна–Таккера в этой задаче.
2. Решение задачи квадратичного программирования методом Баранкина-Дорфмана.

Частным случаем задачи выпуклого программирования является задача квадратичного программирования. В качестве основной в квадратичном программировании рассматривается задача минимизации функции Z, являющуюся суммой линейной и квадратичной функции:

$$Z = c_1x_1 + c_2x_2 + \ldots + c_nx_n + d_{11}x_1^2 + d_{22}x_2^2 + \ldots + d_{nn}x_n^2 + \\ + 2\left(d_{12}x_1x_2 + d_{13}x_1x_3 + \ldots + d_{n-1,n}x_{n-1}x_n\right) = \\ = \sum_{j=1}^{n} c_jx_j + \sum_{j=1}^{n}\sum_{k=1}^{n} d_{jk}x_jx_k \qquad (12.1)$$

при линейных ограничениях

$$\sum_{j=1}^{n} a_{ij}x_j \le b_i, \quad (i = 1,2,\ldots,m), \qquad (12.2)$$

$$x_j \ge 0, \quad (j = 1,2,\ldots,n). \qquad (12.3)$$

Матрица $D = \left\| d_{jk} \right\|$ предполагается симметрической и неотрицательно определенной. В этом случае функция (12.1) будет выпуклой.

Составим для задачи (12.1) – (12.3) локальные условия Куна–Таккера (11.8) – (11.9), являющиеся необходимыми и достаточными условиями оптимальности решения задачи (12.1) – (12.3).

Функция Лагранжа в данном случае имеет вид

$$F(x_1, x_2, \ldots, x_n, \lambda_1, \lambda_2, \ldots, \lambda_m) =$$

$$= \sum_{j=1}^{n} c_j x_j + \sum_{j=1}^{n}\sum_{k=1}^{n} d_{jk} x_j x_k + \sum_{i=1}^{m} \lambda_i \left(\sum_{j=1}^{n} a_{ij} x_j - b_i \right).$$

Найдем частные производные этой функции:

$$\frac{\partial F}{\partial x_j} = c_j + 2\sum_{k=1}^{n} d_{jk} x_k + \sum_{i=1}^{m} \lambda_i a_{ij} \quad (j = 1,2,\ldots,n), \qquad (12.4)$$

$$\frac{\partial F}{\partial \lambda_i} = \sum_{j=1}^{n} a_{ij} x_j - b_i \quad (i = 1,2,\ldots,m). \qquad (12.5)$$

Обозначим

$$c_j + 2\sum_{k=1}^{n} d_{jk} x_k + \sum_{i=1}^{m} \lambda_i a_{ij} = v_j \quad (j = 1,2,\ldots,n),$$

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

С учетом этих обозначений, соотношений (12.4) и (12.5) условия Куна–Таккера (11.8) – (11.9) примут следующий вид

$$\sum_{j=1}^{n} a_{ij} x_j + y_i = b_i \quad (i = 1,2,\ldots,m), \qquad (12.6)$$

$$2\sum_{k=1}^{n} d_{jk} x_k - v_j + \sum_{i=1}^{m} \lambda_i a_{ij} = -c_j \quad (j = 1,2,\ldots,n), \qquad (12.7)$$

$$\begin{cases} x_j \ge 0, \quad v_j \ge 0 \quad (j = 1,2,\ldots,n), \\ y_i \ge 0, \quad \lambda_i \ge 0 \quad (i = 1,2,\ldots,m), \end{cases} \qquad (12.8)$$

$$\sum_{j=1}^{n} x_j v_j + \sum_{i=1}^{m} y_i \lambda_i = 0. \qquad (12.9)$$

Равенства (12.6) и (12.7) образуют систему $N = n + m$ линейных уравнений с $2N = 2(n+m)$ неизвестными: $x_1,\ldots,x_n, y_1,\ldots,y_m, v_1,\ldots,v_n, \lambda_1,\ldots,\lambda_m$.

Итак, в соответствии с теоремой Куна–Таккера решение $\mathbf{X} = (x_1,\ldots,x_n)$ задачи (12.1) – (12.3) квадратичного программирования является оптимальным тогда и только тогда, когда совместно с решением $\mathbf{V} = (v_1,\ldots,v_n)$ существуют решения $\mathbf{Y} = (y_1,\ldots,y_m)$ и $\mathbf{L} = (\lambda_1,\ldots,\lambda_m)$ такие, что $\mathbf{U} = (x_1,\ldots,x_n, y_1,\ldots,y_m, v_1,\ldots,v_n, \lambda_1,\ldots,\lambda_m)$ является решением системы (12.6) – (12.8) при условии выполнения равенства (12.9).

Условие (12.9) для задачи (12.1) – (12.3) равносильно выполнению условия

$$T = \mathbf{U}\widetilde{\mathbf{U}} = 0, \qquad (12.10)$$

где $\widetilde{\mathbf{U}} = (v_1, \ldots, v_n, \lambda_1, \ldots, \lambda_m, x_1, \ldots, x_n, y_1, \ldots, y_m)$.

Существует несколько методов решения задачи квадратичного программирования. Рассмотрим один из них — метод Баранкина-Дорфмана.

При этом методе сначала система уравнений (12.6) и (12.7) приводится к виду

$$\begin{cases} z_1 = h_{10} - \sum_{k=1}^{N} h_{1k} z_{N+k} \\ z_2 = h_{20} - \sum_{k=1}^{N} h_{2k} z_{N+k} \\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot \\ z_N = h_{N0} - \sum_{k=1}^{N} h_{Nk} z_{N+k} \end{cases}, \qquad (12.11)$$

где z_i (базисные переменные) и z_{N+i} (свободные переменные) являются р а з л и ч н ы м и элементами некоторой перестановки переменных $x_1, \ldots, x_n, y_1, \ldots, y_m, v_1, \ldots, v_n, \lambda_1, \ldots, \lambda_m$, а все h_{i0} являются неотрицательными числами ($i = 1, 2, \ldots, N$).

Затем из системы (12.11) находится начальное базисное решение

$\mathbf{U}_0 = (z_1 = h_{10};\ z_2 = h_{20};\ \ldots;\ z_N = h_{N,0};\ z_{N+1} = 0;\ z_{N+2} = 0;\ \ldots;\ z_{2N} = 0)$ системы (12.6) – (12.8), причем компоненты вектора $\mathbf{U}_0$ расположены в порядке $x_1, \ldots, x_n, y_1, \ldots, y_m, v_1, \ldots, v_n, \lambda_1, \ldots, \lambda_m$.

Если для решения $\mathbf{U}_0$ выполняется условие (12.10), то задача (12.1) – (12.3) решена и ее оптимальное решение $\mathbf{X}^* = (x_1^*, \ldots, x_n^*)$ находится из соответствующих компонент вектора $\mathbf{U}_0$. Кроме того, вычисляется $Z_{\min} = \sum_{j=1}^{n} c_j x_j^* +$

$+ \sum_{j=1}^{n} \sum_{k=1}^{n} d_{jk} x_j^* x_k^*$.

Если же для $\mathbf{U}_0$ условие (12.10) не выполняется, то для перехода к другому базисному решению составляется нижеследующая таблица (табл. 12.1). В основную часть этой таблицы включаются строки для всех переменных, расположенных в порядке $x_1, \ldots, x_n, y_1, \ldots, y_m, v_1, \ldots, v_n, \lambda_1, \ldots, \lambda_m$. Для базисных переменных элементы строк берутся из системы (12.11), а для свободных переменных — из соотношений

$$z_{N+i} = 0 - (-1) \cdot z_{N+i} \quad (i = 1, 2, \ldots, N).$$

Таблица 12.1

	1	$-z_{N+1}$	$-z_{N+2}$	$\ldots$	$-z_{2N}$
$z_1 =$	h_{10}	h_{11}	h_{12}	$\ldots$	$h_{1,N}$
$z_2 =$	h_{20}	h_{21}	h_{22}	$\ldots$	$h_{2,N}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$z_N =$	h_{N0}	h_{N1}	h_{N2}	$\ldots$	$h_{N,N}$
$z_{N+1} =$	0	–1	0	$\ldots$	0
$z_{N+2} =$	0	0	–1	$\ldots$	0
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$z_{2N} =$	0	0	0	$\ldots$	–1
T	$\mathbf{U}\tilde{\mathbf{U}}$				
α_s		α_1	α_2	$\ldots$	α_N
β_s		β_1	β_2	$\ldots$	β_N
θ_s		θ_1	θ_2	$\ldots$	θ_N
K_s		K_1	K_2	$\ldots$	K_N

Параметры же дополнительной части таблицы 12.1 находятся следующим образом:

а) α_s находятся из формулы $\alpha_s = \tilde{\mathbf{U}}\mathbf{H}_s$, где $\mathbf{H}_s$ — вектор, составленный из элементов s-го столбца основной части таблицы;

б) для тех s, для которых $\alpha_s > 0$, вычисляются остальные параметры: $\beta_s = \mathbf{H}_s \tilde{\mathbf{H}}_s$, $\theta_s = \min\limits_{h_{is}>0} \dfrac{h_{i0}}{h_{is}}$ (элементы соответствующих столбцов, доставляющих минимум θ_s, отмечаются звездочкой), $K_s = -2\alpha_s + \theta_s \beta_s$.

Столбец, которому соответствует наименьший из отрицательных K_s, назначается разрешающим столбцом, строка с отмеченным звездочкой элементом этого столбца — разрешающей строкой, а сам этот элемент — разрешающим элементом, и с их помощью выполняется симплексное преобразование табл. 12.1.

При этом:

1) все элементы основной части таблицы, за исключением элементов разрешающей строки и разрешающего столбца, вычисляются по формуле $h'_{ij} = h_{ij} - \dfrac{h_{il}h_{kj}}{h_{kl}}$ ($i = 1, 2, \ldots, 2N$, $i \neq k$, $j = 0, 1, \ldots, N$, $j \neq l$), где h'_{ij} — элемент нового j го столбца таблицы, а h_{ij} — элемент прежнего j го столбца таблицы, k — номер разрешающей строки, а l — номер разрешающего столбца;

2) все элементы разрешающего столбца, за исключением самого разрешающего элемента, делятся на разрешающий элемент и меняют свои знаки;

3) разрешающая строка заменяется прежней строкой, соответствовавшей переменной, ставшей теперь базисной.

В результате получается новое базисное решение системы (12.6) – (12.8). Этот процесс продолжается до тех пор, пока не будет выполнено условие (12.10). Если же все $K_s > 0$, а $T > 0$, то в качестве начального выбирается другое базисное решение.

Пример 12.1. Минимизировать функцию

$$Z = -2x_1 - 3x_2 + 0{,}2\,x_1^2 + 0{,}2\,x_2^2$$

при ограничениях:

$$\begin{cases} 2x_1 + 3x_2 \leq 13 \\ 2x_1 + x_2 \leq 10 \end{cases},$$

$$x_1 \geq 0, \quad x_2 \geq 0.$$

Решение. Из соотношений (12.6) и (12.7) получаем следующую систему уравнений:

$$\begin{cases} 2x_1 + 3x_2 + y_1 & & & = 13 \\ 2x_1 + x_2 \quad + y_2 & & & = 10 \\ 2/5x_1 & -v_1 & +2\lambda_1 + 2\lambda_2 & = 2 \\ \quad 2/5x_2 & -v_2 & +3\lambda_1 \quad + \lambda_2 & = 3 \end{cases}.$$

После несложных преобразований приводим эту систему к виду

$$\begin{cases} x_2 = 3/2 - 1/2y_1 + 1/2y_2 \\ x_1 = 17/4 + 1/4y_1 - 3/4y_2 \\ \lambda_2 = 12/5 + 1/5y_1 - 1/5y_2 + v_2 - 3\lambda_1 \\ v_1 = 9/2 + 1/2y_1 - 7/10y_2 + 2v_2 - 4\lambda_1 \end{cases}, \qquad (12.12)$$

откуда с учетом условий неотрицательности

$$x_1 \ge 0,\ x_2 \ge 0,\ y_1 \ge 0,\ y_2 \ge 0,\ v_1 \ge 0,\ v_2 \ge 0,\ \lambda_1 \ge 0,\ \lambda_2 \ge 0 \quad (12.13)$$

находим начальное базисное решение $\mathbf{U}_0 = (17/4;\ 3/2;\ 0;\ 0;\ 9/2;\ 0;\ 0;\ 12/5)$ системы (12.12) – (12.13). При этом значение целевой функции равно $Z_0 = -143/16 = -8{,}9375.$

Так как

$$T_0 = \mathbf{U}_0\tilde{\mathbf{U}}_0 =$$
$$= (17/4; 3/2; 0; 0; 9/2; 0; 0; 12/5)\cdot(9/2; 0; 0; 12/5; 17/4; 3/2; 0; 0) =$$
$$= 17/4\cdot 9/2 + 9/2\cdot 17/4 = 153/4 \ne 0,$$

то не выполняется условие (12.10) и, следовательно, решение $\mathbf{U}_0 = (17/4;\ 3/2;\ 0;\ 0;\ 9/2;\ 0;\ 0;\ 12/5)$ не является оптимальным.

Поэтому составим таблицу 12.2 для перехода к новому базисному решению системы (12.12) – (12.13). Основную часть этой таблицы заполним, используя систему (12.12). Для дополнительной же части этой таблицы:

а) найдем $\alpha_1 = \tilde{\mathbf{U}}_0\mathbf{H}_1 = (9/2; 0; 0; 12/5; 17/4; 3/2; 0; 0)\times$ $\times(-1/4; 1/2; -1; 0; -1/2; 0; 0; -1/5) = -13/4,$

$\alpha_2 = \tilde{\mathbf{U}}_0\mathbf{H}_2 = 79/20,\ \alpha_3 = \tilde{\mathbf{U}}_0\mathbf{H}_3 = -10,\ \alpha_4 = \tilde{\mathbf{U}}_0\mathbf{H}_4 = 17;$

б) для положительных α_2 и α_4 вычислим остальные параметры: $\beta_2 = \mathbf{H}_2\tilde{\mathbf{H}}_2 = (3/4; -1/2; 0; -1; 7/10; 0; 0; 1/5)\times$ $\times(7/10; 0; 0; 1/5; 3/4; -1/2; 0; -1) = 13/20,$ $\beta_4 = \mathbf{H}_4\tilde{\mathbf{H}}_4 = 0,$

$\theta_2 = \min(17/4:3/4; 9/2:7/10; 12/5:1/5) = 17/4:3/4 = 17/3$,
$\theta_4 = 12/5:3 = 4/5$, $\quad K_2 = -2 \cdot 79/20 + 17/3 \cdot 13/20 = -253/60$,
$K_4 = -34$.

Т а б л и ц а 12.2

	1	$-y_1$	$-y_2$	$-v_2$	$-\lambda_1$
$x_1 =$	17/4	–1/4	$3/4^*$	0	0
$x_2 =$	3/2	1/2	–1/2	0	0
$y_1 =$	0	–1	0	0	0
$y_2 =$	0	0	–1	0	0
$v_1 =$	9/2	–1/2	7/10	–2	4
$v_2 =$	0	0	0	–1	0
$\lambda_1 =$	0	0	0	0	–1
$\lambda_2 =$	12/5	–1/5	1/5	–1	$\boxed{3}^*$
T	153/4				
α_s		–13/4	79/20	–10	17
β_s			13/20		0
θ_s			17/3		4/5
K_s			–253/60		–34

Четвертый столбец, которому соответствует наименьший отрицательный K_s, т.е. $K_4 = -34$, назначаем разрешающим столбцом, строку с элементом 3 этого столбца — разрешающей строкой, а сам элемент 3 — разрешающим элементом, и с их помощью выполняем симплексное преобразование табл. 12.2.

В результате получаем таблицу 12.3, содержащую новое базисное решение $\mathbf{U}_1 = (17/4;\ 3/2;\ 0;\ 0;\ 13/10;\ 0;\ 4/5;\ 0)$. Для этого решения $T_1 = 17/4 \cdot 13/10 + 13/10 \cdot 17/4 = 221/20 \neq 0$.

Поэтому заполним дополнительную часть таблицы 12.3 аналогично тому, как это делалось в предыдущем случае, для перехода к новому базисному решению системы (12.12) – (12.13).

Подвергнув таблицу 12.3 симплексному преобразованию с разрешающим элементом 13/30, получим очередную таблицу с базисным решением $\mathbf{U}_2 = (2;\ 3;\ 0;\ 3;\ 0;\ 0;\ 1/5;\ 0)$, для которого $T_2 = 0$.

Т а б л и ц а 12.3

	1	$-y_1$	$-y_2$	$-v_2$	$-\lambda_2$
$x_1 =$	17/4	–1/4	3/4	0	0
$x_2 =$	3/2	1/2	–1/2	0	0
$y_1 =$	0	–1	0	0	0
$y_2 =$	0	0	–1	0	0
$v_1 =$	13/10	–7/30	$\boxed{13/30^*}$	–2/3	–4/3
$v_2 =$	0	0	0	–1	0
$\lambda_1 =$	4/5	–1/15	1/15	–1/3	1/3
$\lambda_2 =$	0	0	0	0	–1
T	221/20				
α_s		–127/60	169/60	–13/3	–17/3
β_s			13/20		
θ_s			3		
K_s			–221/60		

Тем самым найдено оптимальное решение $\mathbf{X}^* = (2; 3)$, при котором целевая функция Z данной задачи минимизируется. При этом $Z_{\min} = Z_2 = -10{,}4$.

Вопросы для повторения и контроля :

1. Какая задача рассматривается в качестве основной в квадратичном программировании?
2. Каковы условия Куна–Таккера для задачи квадратичного программирования?
3. Как находится начальное базисное решение задачи квадратичного программирования в методе Баранкина-Дорфмана и каково условие оптимальности ее решения?
4. Из каких частей состоит таблица задачи квадратичного программирования и как она заполняется?
5. Как осуществляется переход к новому базисному решению в задаче квадратичного программирования?

Опорные слова :

Задача квадратичного программирования, условия Куна–Таккера для задачи квадратичного программирования, метод Баранкина-Дорфмана для решения задачи квадратичного программирования, базисное решение задачи квадратичного программирования, оптимальное решение задачи квадратичного программирования, таблица задачи квадратичного программирования, симплексные преобразования в задаче квадратичного программирования.

Тема № 13

Динамическое программирование. Функциональные уравнения Беллмана

План :

1. Понятие динамического программирования.
2. Общая постановка задачи динамического программирования.
3. Функциональные уравнения Беллмана.

В задачах линейного и нелинейного программирования, рассмотренных в предыдущих темах, экономический процесс считался статическим, т.е. не зависящим от времени, поэтому оптимальное решение находилось только на один этап планирования.

Решение задач математического программирования, которые могут быть представлены в виде многошагового (многоэтапного) процесса, составляет предмет динамического программирования. В задачах динамического программирования экономический процесс зависит от времени или от нескольких периодов (этапов) времени, поэтому находится ряд оптимальных решений (последовательно для каждого этапа), обеспечивающих оптимальное развитие всего процесса в целом.

Динамическое программирование представляет собой математический аппарат, позволяющий осуществлять оптимальное планирование многошаговых управляемых процессов и процессов, зависящих от времени.

Суть метода динамического программирования состоит в том, что вместо поиска оптимального решения сразу для всей сложной задачи предпочитают находить оптимальные решения для нескольких более простых задач аналогичного содержания, на которые расчленяется исходная задача.

Другой важной особенностью метода динамического программирования является независимость оптимального решения, принимаемого на очередном шаге, от того, каким образом оптимизируемый процесс достиг теперешнего состояния. Оптимальное решение выбирается лишь с учетом факторов, характеризующих процесс в данный момент.

Метод динамического программирования характеризуется также тем, что выбор оптимального решения на каждом шаге должен производиться с учетом его последствий в будущем. Это означает, что, оптимизируя процесс на каждом отдельном шаге, ни в коем случае нельзя забывать обо всех последующих шагах. Планируя многоэтапный процесс, исходят из интересов всего процесса в целом, т.е. при принятии решения на отдельном этапе всегда необходимо иметь в виду конечную цель.

Поэтапное планирование многошагового процесса должно производиться так, чтобы при планировании каждого шага учитывалась не выгода, получаемая только на данном шаге, а общая выгода, получаемая по окончании всего процесса, и именно относительно общей выгоды производится оптимальное планирование. Этот принцип выбора решения в динамическом программировании является определяющим и носит название *принципа оптимальности*.

Примерами задач теории динамического программирования являются задачи распределения ресурсов, замены оборудования, задача о выборе наиболее экономного маршрута доставки груза.

Рассмотрим некоторый развивающийся во времени управляемый процесс, т.е. такой, на развитие которого можно влиять принимаемыми решениями (в экономических процессах управление заключается в перераспределении средств, изменении состава оборудования, изменении объемов поставок сырья и т.п.). Предполагается, что процесс распадается (естественно или искусственно) на N шагов (этапов). Состояние процесса на начало каждого шага будет характеризоваться вектором $\mathbf{s}_i = \left(s_{i1}; \ldots; s_{im}\right)$. Этот вектор называется *вектором состояния процесса*. Множество всех состояний, в которых может находиться процесс на начало i-го шага, обозначается через S_i. Считается, что начальное состояние процесса известно, т.е. задан вектор $\mathbf{s}_0$.

Развитие процесса заключается в последовательном переходе из одного состояния в другое. Если процесс находится в состоянии $\mathbf{s}_i$, то состояние его $\mathbf{s}_{i+1}$ на следующем шаге определяется не только вектором $\mathbf{s}_i$, но и решением $\mathbf{u}_i$, принятым на i-м шаге. Это записывается следующим образом: $\mathbf{s}_{i+1} = \varphi(\mathbf{s}_i;\ \mathbf{u}_i)$. Ясно, что решение на каждом шаге не может быть совершенно произвольным; его следует выбирать из некоторого множества U_i возможных решений. Развитие процесса в течение всего рассматриваемого периода можно однозначно описать последовательностью состояний $\mathbf{s}_0, \mathbf{s}_1, \ldots, \mathbf{s}_N$, где $\mathbf{s}_i \in S_i$.

Любая последовательность $\mathbf{u}_0, \mathbf{u}_1, \ldots, \mathbf{u}_{N-1}$ допустимых решений, переводящая процесс из начального состояния $\mathbf{s}_0$ в конечное $\mathbf{s}_N$, называется *стратегией*. Для полного описания N шагового процесса каждой стратегии надо сопоставить некоторую оценку — значение целевой функции $f_N(\mathbf{s})$. Зададим целевую функцию в виде суммы оценочных функций $r_i(\mathbf{s}_i;\ \mathbf{s}_{i+1})$, значения которых получаются на каждом шаге процесса при переходе из состояния $\mathbf{s}_i$ в состояние $\mathbf{s}_{i+1}$, т.е. $f_N(\mathbf{s}) = \sum_{i=0}^{N-1} r_i(\mathbf{s}_i;\ \mathbf{s}_{i+1})$.

Таким образом, общую задачу динамического программирования можно сформулировать следующим образом:

найти стратегию $\mathbf{U} = \{\mathbf{u}_0, \mathbf{u}_1, \ldots, \mathbf{u}_{N-1}\}$, доставляющую экстремум функции

$$f_N(\mathbf{s}) = \sum_{i=0}^{N-1} r_i(\mathbf{s}_i;\ \mathbf{s}_{i+1}) \tag{13.1}$$

при условиях

$$\mathbf{s}_0 \text{ — вектор начального состояния процесса,} \tag{13.2}$$

$$\mathbf{s}_{i+1} = \varphi(\mathbf{s}_i;\ \mathbf{u}_i),\ \mathbf{s}_i \in S_i,\ \mathbf{u}_i \in U_i\ (i = 0, \ldots, N-1). \tag{13.3}$$

Пусть $\mathbf{s}_0$ и $\mathbf{s}_N$ — соответственно начальное и конечное состояние N шагового процесса. Обозначим через $f_N(\mathbf{s}_0)$ экстремальное значение целевой функции, полученное за N шагов при оптимальной стратегии $\mathbf{U}$ управления процессом, находившемся сначала в состоянии $\mathbf{s}_0$.

Допустим, что на первом шаге было принято некоторое решение $\mathbf{u}_0$, под воздействием которого процесс перешел из состояния $\mathbf{s}_0$ в состояние $\mathbf{s}_1$. Полученный при этом эффект характеризуется значением $r_0(\mathbf{s}_0;\ \mathbf{u}_0)$ оценочной функции $r_i(\mathbf{s}_i;\ \mathbf{u}_i)$.

Предположим, что после первого шага для управления процессом применялась оптимальная стратегия, обеспечивающая на оставшихся N – 1 шагах экстремальное значение $f_{N-1}(\mathbf{s}_1)$ целевой функции. При описанных условиях общая оценка качества управления за N шагов выразится суммой

$$r_0(\mathbf{s}_0;\ \mathbf{u}_0) + f_{N-1}(\mathbf{s}_1). \qquad (13.4)$$

Экстремальное значение $f_N(\mathbf{s}_0)$ целевой функции за N шагов будет равно экстремуму суммы (13.4), который зависит от начального решения $\mathbf{u}_0$, т.е.

$$f_N(\mathbf{s}_0) = \underset{\mathbf{u}_0}{\operatorname{extr}}\left[r_0(\mathbf{s}_0;\ \mathbf{u}_0) + f_{N-1}(\mathbf{s}_1)\right]. \qquad (13.5)$$

Обозначим через $f_{N-i}(\mathbf{s}_i)$ экстремум целевой функции, полученный на последних N – i шагах, если сначала процесс находился в состоянии $\mathbf{s}_i$. Тогда по аналогии с равенством (13.5) получим

$$f_{N-i}(\mathbf{s}_i) = \underset{\mathbf{u}_i}{\operatorname{extr}}\left[r_i(\mathbf{s}_i;\ \mathbf{u}_i) + f_{N-(i+1)}(\mathbf{s}_{i+1})\right], \qquad (13.6)$$

где $i = 0,\ldots,N-1$.

Выражение (13.6) является математической записью принципа оптимальности и представляет собой уравнения, называемые *функциональными уравнениями Беллмана*. Оно является рекуррентным, поэтому вся последовательность вычислений, приводящая к $f_N(\mathbf{s}_0)$, может быть выполнена, если определены значения функций $f_{N-(i+1)}$.

Выражение (13.6) носит символический характер и непригодно для непосредственных вычислений. В нем не указан конкретный вид функций $r_i(\mathbf{s}_i;\ \mathbf{u}_i)$ и не определен характер аргументов $\mathbf{s}_i$, $\mathbf{u}_i$, $\mathbf{s}_{i+1}$, поэтому оно лишь указывает на общую принципиальную схему вычислений при использовании метода динамического программирования. Для каждой конкретной задачи функциональные уравнения имеют свой специфический вид, но в них непременно должен сохраняться рекуррентный характер, заложенный в выражении (13.6) и воплощающий основную идею принципа оптимальности.

Вопросы для повторения и контроля :

1. Что является предметом динамического программирования и в чем суть динамического программирования?
2. Как выбирается оптимальное решение для каждого этапа решения задачи динамического программирования и что вы знаете о принципе оптимальности?
3. Как происходит развитие процесса, как принимается решение на каждом шаге и что такое вектор состояния процесса?
4. Что вы знаете о стратегии и оценочных функциях, и как формулируется общая задача динамического программирования?
5. Чем являются функциональные уравнения Беллмана и как они выводятся?

Опорные слова :

Динамическое программирование, оптимальное решение задачи динамического программирования, принцип оптимальности, вектор состояния процесса, стратегия, оценочные функции, задача динамического программирования, функциональные уравнения Беллмана.

Тема № 14

Теория игр

План :

1. Предмет и некоторые основные понятия теории игр.
2. Решение матричных игр в различных стратегиях.
3. Связь между матричными играми и задачами линейного программирования.
4. Игры с природой.
5. Различные критерии выбора оптимальной стратегии.

В процессе человеческой деятельности возникают ситуации, в которых интересы отдельных лиц (участников, групп, сторон) либо прямо противоположны (антагонистичны), либо, не будучи непримиримыми, все же не совпадают, причем они достигают своей цели различными путями. Например, спортивные игры, арбитражные споры и т.п. Здесь каждый из участников сознательно стремится добиться наилучшего результата за счет другого участника. Подобного рода ситуации встречаются и в различных сферах производственной деятельности.

Столкновение противоположных интересов участников приводит к возникновению *конфликтных ситуаций*, т.е. ситуаций, в которых интересы участников (игроков) противоположны или не совпадают. Для таких ситуаций характерно, что эффективность решений, принимаемых в ходе конфликта каждой из сторон, существенно зависит от действий другой стороны. При этом ни одна из сторон не может полностью контролировать положение, так как и той, и другой стороне решения приходится принимать в условиях неопределенности.

Теория игр — это раздел математики, изучающий конфликтные ситуации. Теория игр является математической теорией конфликтных ситуаций, определяющей рекомендации по рациональному образу действий каждого из участников в ходе конфликтной ситуации, т.е. таких действий, которые обеспечивали бы ему наибольший выигрыш (наименьший проигрыш).

Чтобы исключить трудности, возникающие при анализе практических конфликтных ситуаций в результате наличия многих несущественных факторов, строится упрощенная модель ситуации, которая называется *игрой*. Игра отличается от реального конфликта тем, что ведется по определенным правилам.

В большинстве игр предполагается, что интересы участников поддаются количественному описанию, т.е. результат игры (*выигрыш*) определяется некоторым числом. Величина выигрыша зависит от стратегии, применямой игроком. *Стратегией* игрока называется план, по которому он совершает выбор в любой возможной ситуации и при любой возможной фактической информации. Задачей теории игр является выработка рекомендаций для игроков, т.е. определение для них оптимальной стратегии.

Всякая игра состоит из отдельных партий. *Партией* называется каждый вариант проведения игры определенным образом. В свою очередь, в партии игроки совершают определенные ходы. *Ходом* в теории игр называется выбор одного из предположенных правилами игры действий и его осуществление.

В игре могут сталкиваться интересы двух или более противников. В первом случае игра называется *парной*, во втором — *множественной*. В зависимости от числа возможных стратегий игры делятся на *конечные* и *бесконечные*. По характеру выигрышей игры делятся на игры *с нулевой суммой* и *ненулевой суммой*. В первых общий капитал игроков не меняется, а лишь перераспределяется в ходе игры, в связи с чем сумма выигрышей равна нулю. В играх с ненулевой суммой сумма выигрышей участников отлична от нуля.

Рассмотрим простейший вид стратегической игры — парную игру с нулевой суммой. Игра состоит из двух ходов: игрок A выбирает одну из своих возможных стратегий A_i $(i = 1, 2, \ldots, m)$, а игрок B — одну из своих возможных стратегий B_j $(j = 1, 2, \ldots, n)$, причем каждый выбор производится при полном незнании выбора другого игрока. Такие стратегии называются *чистыми стратегиями*.

Каждой паре чистых стратегий A_i и B_j сопоставляется число a_{ij} — выигрыш игрока A за счет игрока B или проигрыш игрока B. Составим из этих чисел следующую матрицу:

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix}. \qquad (14.1)$$

Строки этой матрицы соответствуют стратегиям A_i, а столбцы — стратегиям B_j. Матрица $\mathbf{A}$ называется *платежной матрицей* или *матрицей игры*.

Целью игроков является выбор наиболее выгодных стратегий, доставляющих игроку A максимальный выигрыш, а игроку B — минимальный проигрыш. Стратегия игрока A называется *оптимальной*, если при ее применении выигрыш игрока A не уменьшается, какими бы стратегиями не пользовался игрок B. Оптимальной для игрока B называется стратегия, при использовании которой проигрыш игрока B не увеличивается, какие бы стратегии не применял игрок A.

Если игрок A выбирает некоторую стратегию A_i, то в наихудшем случае (например, если выбор станет известным игроку B) он получит выигрыш, равный $\alpha_i = \min_j a_{ij}$. Поэтому он должен выбрать такую стратегию, при которой его минимальные выигрыши α_i достигают своего максимума:

$$\alpha = \max_i \alpha_i = \max_i \min_j a_{ij}.$$

Величина α (гарантированный выигрыш игрока A) называется *нижней ценой* игры. Стратегия A_{i_α}, обеспечивающая получение α, называется *максиминной*.

Аналогично, при выборе некоторой стратегии B_j проигрыш игрока B не превысит $\beta_j = \max_i a_{ij}$. Поэтому он естественно выберет такую стратегию, при которой его максимальные проигрыши β_j достигают своего минимума:

$$\beta = \min_j \beta_j = \min_j \max_i a_{ij}.$$

Величина β (гарантированный проигрыш игрока B) называется *верхней ценой* игры. Стратегия B_{j_β}, обеспечивающая получение β, называется *минимаксной*.

Итак, при разумных действиях противников выигрыш игрока A (проигрыш игрока B) ограничен нижней и верхней ценами игры. Если эти величины равны, то выигрыш игрока A — вполне определенное число

$$v = \max_i \min_j a_{ij} = \min_j \max_i a_{ij},$$

которое называется *ценой игры* и равно элементу $a_{i_\alpha j_\beta}$ матрицы $\mathbf{A}$. Этому элементу соответствуют оптимальные стратегии игроков: если один из игроков придерживается своей оптимальной стратегии, то для другого отклонение от его оптимальной стратегии не может быть выгодно.

В играх же, где $\alpha < \beta$, применение минимаксной или максиминной стратегии для каждого из игроков обеспечивает выигрыш, не превышающий α, и проигрыш, не меньший β. Для увеличения выигрыша (уменьшения проигрыша) игроки применяют не одну, а несколько стратегий. Выбор стратегий осуществляется случайным образом. Случайный выбор игроком своих стратегий называется *смешанной стратегией*.

В таких играх стратегии игрока A задаются вероятностями $\mathbf{p} = (p_i)$, с которыми он использует свои чистые стратегии A_i ($i = 1,2,\ldots,m$), а для стратегий игрока B аналогично определяются вероятности $\mathbf{q} = (q_j)$ ($j = 1,2,\ldots,n$). Для вероятностей p_i и q_j имеют место соотношения

$$\sum_{i=1}^{m} p_i = 1, \quad p_i \ge 0 \quad (i = 1,2,\ldots,m),$$

$$\sum_{j=1}^{n} q_j = 1, \quad q_j \ge 0 \quad (j = 1,2,\ldots,n).$$

Выигрыш игрока A при использовании смешанных стратегий определяется как математическое ожидание выигрыша, т.е. равен $\sum_{i=1}^{m}\sum_{j=1}^{n} a_{ij} p_i q_j$. Для оптимальных стратегий игроков имеет место соотношение

$$v=\max_{\mathbf{p}}\min_{\mathbf{q}}\sum_{i=1}^{m}\sum_{j=1}^{n}a_{ij}p_iq_j=\min_{\mathbf{q}}\max_{\mathbf{p}}\sum_{i=1}^{m}\sum_{j=1}^{n}a_{ij}p_iq_j\,. \quad (14.2)$$

Покажем связь между матричными играми и задачами линейного программирования, для чего рассмотрим задачу отыскания оптимальной смешанной стратегии игрока A. Из соотношения (14.2) получаем, что имеют место неравенства

$$\sum_{i=1}^{m}a_{ij}p_i\ge v \quad (j=1,2,\dots,n), \qquad (14.3)$$

которые можно получить, если в (14.2) в качестве $\mathbf{q}$ взять вектор, все компоненты которой равны нулю, за исключением j-ой, которая равна 1.

Если в матрице $\mathbf{A}$ все $a_{ij}>0$, то и цена игры $v>0$. Тогда, разделив обе части неравенств (14.3) на v и обозначив $p_i/v=x_i$, имеем

$$\sum_{i=1}^{m}a_{ij}x_i\ge 1 \quad (j=1,2,\dots,n),$$
$$x_i\ge 0 \quad (i=1,2,\dots,m).$$

Игрок A стремится сделать свой гарантированный выигрыш v возможно больше, а значит, возможно меньше величину

$$1/v=\sum_{i=1}^{m}x_i=z\,.$$

Таким образом, задача отыскания оптимальной смешанной стратегии игрока A сводится к следующей задаче линейного программирования:

минимизировать линейную функцию

$$z=\sum_{i=1}^{m}x_i$$

при линейных ограничениях

$$\sum_{i=1}^{m}a_{ij}x_i\ge 1 \quad (j=1,2,\dots,n),$$
$$x_i\ge 0 \quad (i=1,2,\dots,m).$$

Аналогичную задачу можно получить и для оптимальной смешанной стратегии игрока B.

В отличие от вышерассмотренных задач, во многих задачах, приводящихся к игровым, неопределенность вызвана отсутствием информации об условиях, в которых осуществляется действие. Эти условия зависят не от сознательных действий другого игрока, а от объективной действительности, называемой природой. Такие игры называются *играми с природой*.

Игрок A (человек) в играх с природой старается действовать осмотрительно, используя, например, минимаксную стратегию, позволяющую получить наименьший проигрыш. Второй игрок B (природа) действует совершенно случайно, возможные стратегии определяются как ее состояния (условия погоды, спрос на продукцию, объем перевозок и т.д.). Условия в таких играх задаются в виде матрицы (14.1). Элемент a_{ij} равен выигрышу игрока A, если он использует стратегию A_i, а природа находится в состоянии P_j.

В ряде случаев при решении игры рассматривается *матрица рисков* $\mathbf{R}$. Элементы r_{ij} этой матрицы представляют собой разность между выигрышем, который получил бы игрок A, если бы знал состояние P_j, и выигрышем, который он получит при этом же состоянии, не зная его и применяя стратегию A_i, т.е.

$$r_{ij} = \beta_j - a_{ij},$$

где $\beta_j = \max_i a_{ij}$.

При известном распределении вероятностей различных состояний природы критерием принятия решения является максимум математического ожидания выигрыша (минимум математического ожидания риска). Если вероятности состояний P_j равны q_j ($j = 1,2,\ldots,n$, $\sum_{j=1}^{n} q_j = 1$), то выбор стратегии A_i обеспечивает математическое ожидание выигрыша, равное $\sum_{j=1}^{n} a_{ij} q_j$.

Если же вероятности состояний природы неизвестны, то используются следующие критерии.

Критерий недостаточного основания Лапласа, согласно которому все состояния природы полагаются равновероятными.

Максиминный критерий Вальда, который совпадает с критерием выбора стратегии, позволяющим получить нижнюю цену игры для двух лиц с нулевой суммой. Согласно этому критерию выбирается стратегия, гарантирующая при любых условиях выигрыши, не меньшие чем $\max\limits_{i}\min\limits_{j} a_{ij}$.

Критерий минимального риска Севиджа, рекомендующий выбирать стратегию, при которой величина риска принимает наименьшее значение в самой неблагоприятной ситуации, т.е. $\min\limits_{j}\max\limits_{i} r_{ij}$.

Критерий Гурвица учитывает как пессимистический, так и оптимистический подход к ситуации. Принимается решение о выборе стратегии, при которой имеет место $\max\limits_{i}\left[\lambda \min\limits_{j} a_{ij} + (1-\lambda)\max\limits_{j} a_{ij}\right]$, где $0 \le \lambda \le 1$. Значение λ выбирается на основании субъективных соображений. Чем больше желание подстраховаться в данной ситуации, тем ближе к единице значение λ.

Вопросы для повторения и контроля :

1. Что вы знаете о конфликтных ситуациях и предмете теории игр, что такое игра?
2. Что называется выигрышем, стратегией, партией, ходом, какие виды игр вы знаете?
3. Что такое чистые стратегии, платежная матрица или матрица игры, оптимальная стратегия?
4. Что вы знаете о нижней цене и верхней цене игры, максиминной стратегии, минимаксной стратегии и о цене игры?
5. Что вы знаете о смешанных стратегиях?
6. Какова связь между матричными играми и задачами линейного программирования?
7. Что называется играми с природой, что такое матрица рисков и как находится решение таких игр, когда вероятности состояний природы заданы?
8. Какие вы знаете критерии для решения игр с природой в случае, когда вероятности состояний природы неизвестны?

Опорные слова :

Конфликтные ситуации, теория игр, игра, выигрыш, стратегия, партия, ход, парная игра, множественная игра, конечная игра, бесконечная игра, игра с нулевой суммой, игра с ненулевой суммой, чистые стратегии, платежная матрица или матрица игры, оптимальная стратегия, нижняя цена игры, верхняя цена игры, максиминная стратегия, минимаксная стратегия, цена игры, смешанные стратегии, игры с природой, матрица рисков, вероятности состояний природы, критерии решения игр с природой, критерий недостаточного основания Лапласа, максиминный критерий Вальда, критерий минимального риска Севиджа, критерий Гурвица.

Список литературы

1. Банди Б. Основы линейного программирования. – М.: Радио и связь, 1989.
2. Венцель Е.С. Исследование операции: задачи, принципы, методология. – М.: Высшая школа, 1986.
3. Гершгорн А.С. Математическое программирование и его применение в экономических расчетах. – М.: Высшая школа, 1968.
4. Джамилёв Н.И., Эйдельнант М.И. Сборник задач по математическому программированию. – Т.: Ўќитувчи, 1990.
5. Замков О.О. и др. Математические методы в экономике. – М.: ДИС, 1997.
6. Каршупова Н.И., Плясунова В.С. Математика в экономике. – М.: Вита-Пресс, 1996.
7. Кузнецов А.В., Холод Н.И. Математическое программирование. – Минск: Высшая школа, 1994.
8. Кузнецов Ю.Н., Кузубов В.И., Волощенко А.Б. Математическое программирование. – М.: Высшая школа, 1980.
9. Математическое программирование. / Под ред. Н.Ш. Кремера. – М.: Финстатинформ, 1996.
10. Мулен Э. Теория игр с примерами из математической экономики. – М.: Мир, 1999.
11. Райцкас Р.Л. и др. Количественный анализ в экономике. – М.: Мир, 1992.
12. Сафаева Ќ.С., Бекназарова Н. Операцияларни текширишнинг математик усуллари (ўќув ќўлланма). I ќисм. – Т.: Ўќитувчи, 1984.
13. Сафаева Ќ.С., Бекназарова Н.. Операцияларни текширишнинг математик усуллари (ўќув ќўлланма). II ќисм. – Т.: Ўќитувчи, 1990.
14. Сафаева Ќ.С., Икромов Ш.Р. «Математик программалаш» фанидан маъруза матнлари тўплами. – Т.: Тошкент молия институти, 2001.
15. Сафаева Ќ. Математик дастурлаш (дарслик). – Т.: «Ибн Сино», 2004.
16. Таха Х. Введение в исследование операции. Т. 1,2. – М.: Мир, 1991.

17. Эддоус М., Стенсфилд Р. Методы принятия решения. – М.: Аудит, 1997.
18. Юдин Л.В. и др. Экстремальные модели в экономике. – М.: Экономика, 1993.