## §7. КОМПЬЮТЕРНОЕ МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ В ЭКОНОМИКЕ

### 7.1. ПОСТАНОВКА ЗAДAЧИ ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ

В последние годы мы особенно отчетливо ощутили, что нет ничего важнее для общества, чем здоровая экономика Научное исследование основ функционирования экономики - сложная и интересная деятельность. Математические методы в ней играют возрастающую с каждым десятилетием роль, а реализация возникающих при этом математических моделей и получение практически важных результатов невозможны без ЭВМ.

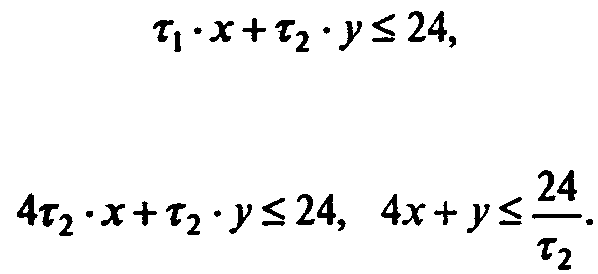
В данном параграфе рассматривается лишь один из разделов - оптимальное планирование - и внутри него одна из моделей, так называемое, линейное программирование. Это связано с относительной простотой и ясностью как содержательной постановки соответствующих задач, так и методов решения. О таких интересных, но более сложных проблемах, как выпуклое программирование, динамическое программирование, теория игр мы лишь упомянем, отсылая читателей за подробностями к специальной литературе. Отметим еще, что термин «программирование» в названии этих разделов теории оптимального планирования весьма условен, связан с историческими обстоятельствами и к программированию в общепринятом сейчас смысле прямого отношения не имеет.

Общеизвестно, сколь важно для решения экономических задач планирование -как при рыночной, так и при плановой экономике. Обычно для решения экономической проблемы существует много способов (стратегий), отнюдь не равноценных по затратам финансов, людских ресурсов, времени исполнения, а также по достигаемым результатам. Наилучший из способов (по отношению к выбранному критерию - одному или нескольким) называют оптимальным. Приведем простейший пример такого рода задач.

*Пример 1.* На некотором предприятии могут выпускать изделия двух видов (например, мотоциклы и велосипеды). В силу ограниченности возможностей сборочного цеха в нем могут собирать за день либо 25 мотоциклов (если не собирать вообще велосипеды), либо 100 велосипедов (если не собирать вообще мотоциклы), либо какую-нибудь комбинацию тех и других, определяемою приемлемыми трудозатратами. Склад может принять не более 70 изделий любого вида в сутки. Известно, что мотоцикл стоит в 2 раза дороже велосипеда. Требуется найти такой план выпуска продукции, который обеспечил бы предприятию наибольшею выручку.

Такого рода задачи возникают повседневно в огромном количестве, но в реальности число изделий гораздо больше двух, да и дополнительных условий тоже больше. Решить подобную задачу путем перебора всех мыслимых вариантов часто невозможно даже на ЭВМ. В нашем примере, однако, в ЭВМ нет необходимости -задача решается очень легко.

Обозначим число выпускаемых за день мотоциклов *х,* велосипедов *- у.* Пусть *τ1* -время (в часах), уходящее на производство одного мотоцикла, а *τ2* - одного велосипеда. Из условия задачи следует, что *τ1* = 4*τ2*. Если завод работает круглосуточно, то, очевидно, при одновременном выпуске обоих изделий



Но  **-** число максимально производимых велосипедов, равное 100. Итак, возможности производства определяют условие

**4*x + y ≤* 100*.***

Еще одно условие - ограниченная емкость склада:

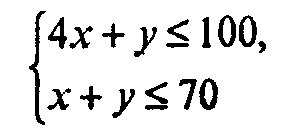
***x + y ≤* 70**

Обозначим цену мотоцикла *а1* (руб.), цену велосипеда – *а2* (руб.). По условию *а1* = 2*a2*. Общая цена дневной продукции

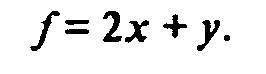
***S = а1 ∙ х + a2 ∙ у* = 2*a2* ∙ *х* + *а2 ∙ у* = *а2 ∙ (*2*х* + *у).***

Поскольку *a2 -* заданная положительная константа, то наибольшего значения следует добиваться отвеличины *f* = *2х* + *у.*

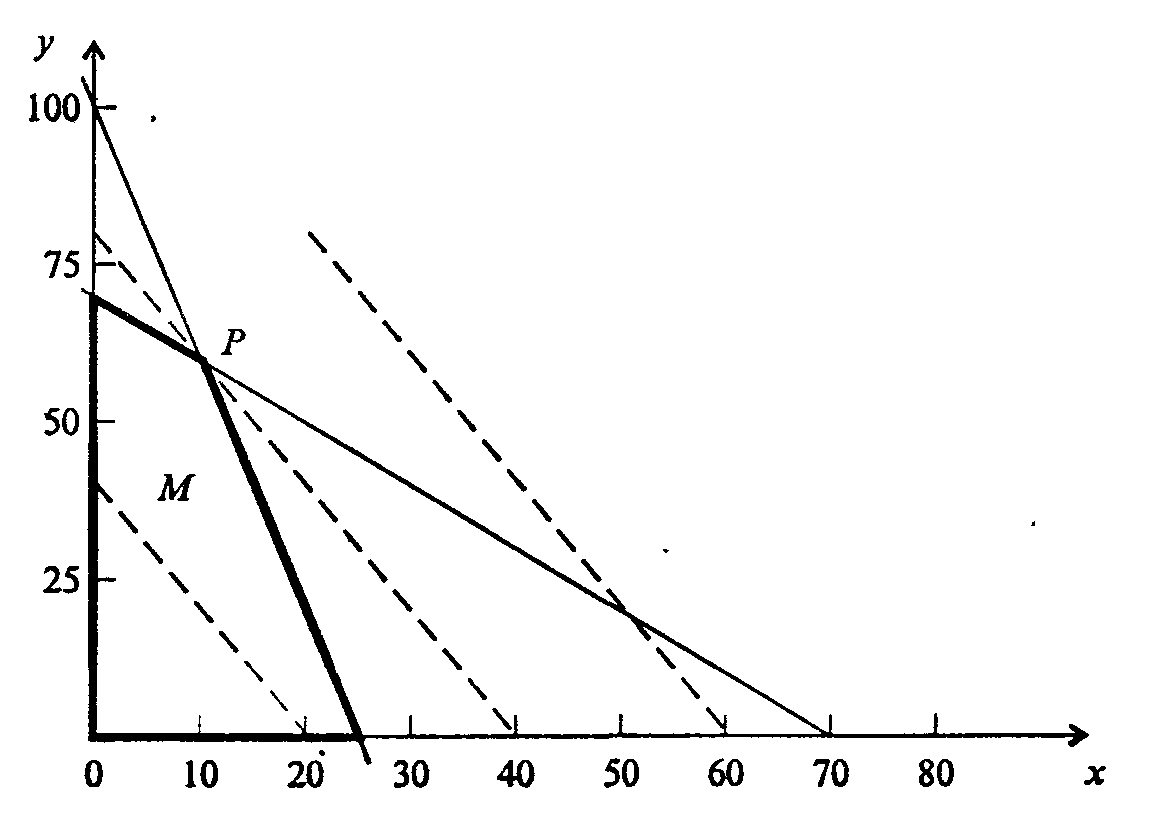
Итак, учитывая все условия задачи, приходим к ее математической модели: среди неотрицательных целочисленных решений системы линейных неравенств

(7.71)

найти такое, которое соответствует максимуму линейной функции

(7.72)

Проще всего решить эту задачу чисто геометрически. Построим на плоскости *(х, у)* область, соответствующую неравенствам (7.71) и условию неотрицательности *x* и *у.* Эта область выделена на рис. 7.62 жирной линией. Всякая ее точка удовлетворяет неравенствам (7.71) и неотрицательности переменных. Пунктирные линии на рисунке - семейство прямых, удовлетворяющих уравнению *f* *= 2х + у* = *с* (с разными значениями константы *с*)*.* Вполне очевидно, что наибольшему возможному значению *f*, совместному с предыдущими условиями, соответствует жирная пунктирная линия, соприкасающаяся с областью *М* в точке *Р.*



*Рис. 7.62.* Графическое решение задачи об оптимальном плане производства (к примеру 1)

Этой линии соответствует значение *f* = 80. Пунктирная линия правее хоть и соответствует большему значению *f*, но не имеет общих точек с *М,* левее - меньшим значениям *f*. Координаты точки *Р*(10*,* 60) - искомый оптимальный план производства.

Отметим, что нам «повезло» - решение *(х, у)* оказалось целочисленным. Если бы прямые

*4x + y = 100*

*х + у = 70*

пересеклись в точке с нецелочисленными координатами, мы бы столкнулись со значительными проблемами. Еще больше их было бы, если бы наш завод выпускал три и более видов продукции.

Прежде чем обсуждать возникающие при этом математические проблемы, дадим формулировки нескольких классических задач линейного программирования в общем виде.

*Пример 2. Транспортная задача.* Некий продукт (например, сталь) вырабатывается на *т* заводах *P1,* *P2*, ..., *Рm*, причем ежемесячная выработка составляетдь *а1, a2, …, аm* тонн, соответственно. Пусть эту сталь надо доставить на предприятия *Q1, Q2,..., Qk* (всего *k),* причем *b1, b2, ..., bk -* ежемесячная потребность этих предприятий. Наконец, пусть задана стоимость *cij* перевозки одной тонны стали с завода *Рi* на предприятие *Qj,.* Естественно считать, что общее производство стали равно суммарной потребности вней:

*a1 + a2 +…+am = b1 + b2 +…+bk* (7.73)

Необходимо составить план перевозок, при котором

1) была бы точно удовлетворена потребность в стали предприятий*Q1, Q2,..., Qk,*

*2)* была бы вывезена вся сталь с заводов *Р1, Р2,....,Pm*;

3) общая стоимость перевозок была бы наименьшей.

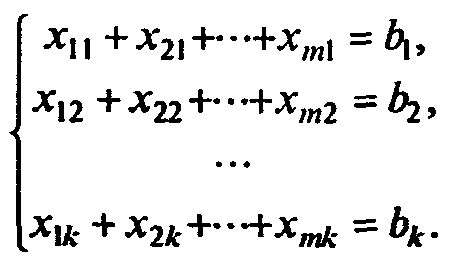
Обозначим через *xij* количество стали (в тоннах), предназначенной к отправке с завода *Рi* на предприятие *Qj.* План перевозок состоит из *(m∙k)* неотрицательных чисел *xij (i=* 1, 2,..., *m;j =* 1,2,..., *k).*

Таблица 7.10

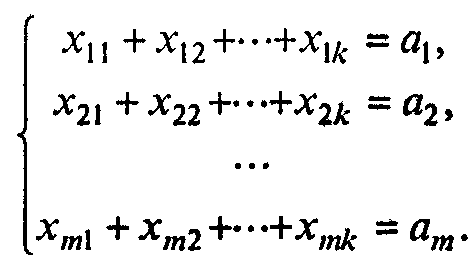
**Схема перевозок стали**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | в *Q1* | в *Q2* | в *Q3* | … | в *Qk* | Отправлено |
| *из P1* | *x11* | *x12* | *x13* | *…* | *x1k* | *a1* |
| *из Р2* | *x21* | *x22* | *x23* | *…* | *x2k* | *a2* |
| *…* | *…* | *…* | *…* | *…* | *…* | *…* |
| *из Pm* | *xm1* | *xm2* | *xm3* | *…* | *xmk* | *am* |
| *Привезено* | *b1* | *b2* | *b3* | *…* | *bk* |  |

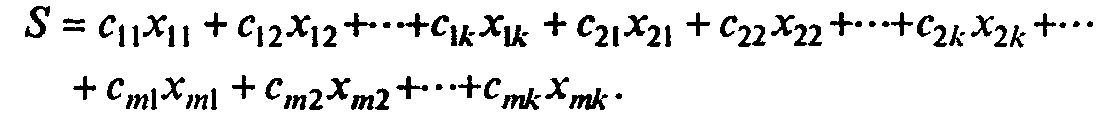
Первое условие примет вид

(7.74)

Второе условие примет вид

(7.75)

Раз стоимость перевозки одной тонны из *Рi* в *Qj* равна *cij,* то общая стоимость *S* всех перевозок равна

(7.76)

Таким образом, мы приходим к следующей чисто математической задаче: дана система *m+k* линейных алгебраических уравнений (7.74) и (7.75) c *m∙k* неизвестными (обычно *т∙k >> m+k)* и линейная функция *S.* Требуется среди всех неотрицательных решений данной системы найти такое, при котором функция *S* достигает наименьшего значения (минимизируется).

Практическое значение этой задачи огромно, ее умелое решение в масштабах нашей страны могло бы экономить ежегодно огромные средства.

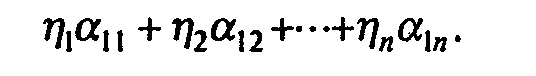
*Пример 3. Задача о диете.* Пусть у врача-диетолога имеется *n* различных продуктов *F1, F2, ..., Fn,* из которых надо составить диету с учетом их питательности. Пусть для нормального питания человеку необходимо *т* веществ *N1, N2, ...,* *Nm*. Предположим, что за месяц каждому человеку необходимо *γ1* кг вещества *N1, γ2* кг вещества *N2*, ..., *γm* кг вещества *Nm.* Для составления диеты необходимо знать содержание питательных веществ в каждом продукте. Обозначим через *αij* количество *i*-го питательного вещества, содержащегося в одном килограмме *j*-го продукта. Всю эту информацию представляют в виде, так называемой, матрицы питательности (табл. 7.11).

Таблица 7.11

**Матрица питательности**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Питательное вещество | Продукт | | | |
| *F1* | *F2* | … | *Fn* |
| *N1* | *α11* | *α12* | *…* | *α1n* |
| *N2* | *α21* | *α22* | *…* | *α2n* |
| … | *…* | *…* | *…* | *…* |
| *Nm* | *αm1* | *αm2* | *…* | *αmn* |

Предположим, что диетолог уже выбрал диету, т.е. определил, что человек должен за месяц потреблять *η1* кг продукта *F1,*..., *ηn* кг продукта *Fn.* Полное количество питательного вещества *N1* будет



По условию требуется, чтобы его, по крайней мере, хватило

***η1α11 + η2 α12 +…+ ηn α1n ≥ γ1***

Точно то же и для остальных веществ. В целом

***η1αi1 + η2 αi2 +…+ ηn αin ≥ γi (i =* 1, 2,… *m)***

Эти условия определяют наличие минимума необходимых питательных веществ. Диета, для которой выполнены условия (7.78) - допустимая диета. Предположим, что из всех допустимых диет должна быть выбрана самая дешевая. Пусть *πi -* цена 1 кг продукта *Fi.* Полная стоимость диеты, очевидно,

***S =* *π1η1 + π2η2 +… + πnηn*.** (7.79)

Таким образом, мы пришли к задаче: найти неотрицательное решение *η1, …, ηn* системы неравенств (7.78), минимизирующее выражение (7.79).

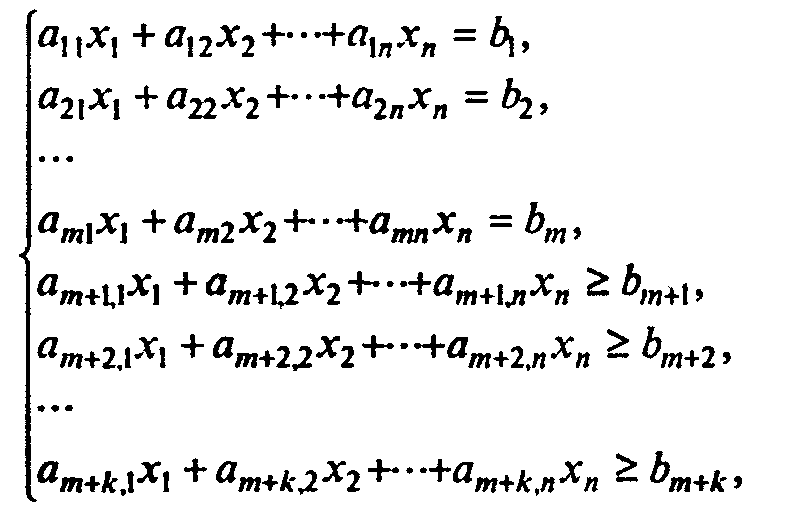
В примерах, приведенных выше. имеется нечто общее. Каждый из них требует нахождения наиболее выгодного варианта в определенной экономической ситуации. С чисто математической стороны в каждой задаче требуется найти значение нескольких неизвестных так, чтобы

1) все эти значения были неотрицательны;

2) удовлетворяли системе линейных уравнений или линейных неравенств;

3) при этих значениях некоторая линейная функция имела бы минимум (или максимум). Таким образом, *линейное программирование -* это математическая дисциплина, изучающая методы нахождения экстремального значения линейной функции нескольких переменных при условии, что последние удовлетворяют конечному числу линейных уравнений и неравенств. Запишем это с помощью формул: дана система линейных уравнений и неравенств.

Запишем это с помощью формул: дана система линейных уравнений и неравенств

(7.80)

и линейная функция

***f = c1x1 + c2x2 + … + cnxn***(7.81)

Требуется найти такое неотрицательное решение

*x1 ≥ 0, x2 ≥ 0, …, xn ≥ 0*

системы (7.80), чтобы функция *f* принимала наименьшее (или наибольшее) значение.

Условия (7.80) называют ограничениями данной задачи, а функцию *f* - целевой функцией (или линейной формой). В приведенных выше примерах ограничения имели вид не уравнений, а неравенств. Заметим, что ограничения в виде неравенств всегда можно свести к системе в виде равенств (способом введения добавочных неизвестных).

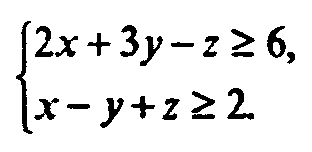
Так, для неравенства

***ai1x1 + ai2x2 + … + ainxn ≥ bi***

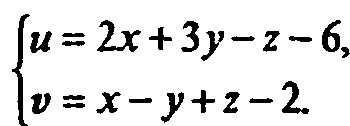
вводя добавочное неизвестное *xn* +1, получаем

***xn*+1 =** ***ai1x1 + ai2x2 + … + ainxn - bi***

Потребовав его неотрицательности наряду с остальными неизвестными, получим, что условие *xn + 1* *≥* 0 превращает (7.84) в (7.83). Введя по отдельному дополнительному неизвестному для каждого из неравенств, получим систему уравнений, равносильную исходной системе неравенств. *Пример.* Дана система неравенств



Сведем ее к системе уравнений. Получим

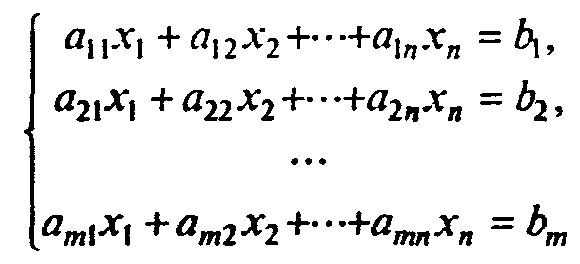


После оптимизации значениями дополнительных неизвестных следует пренебречь.

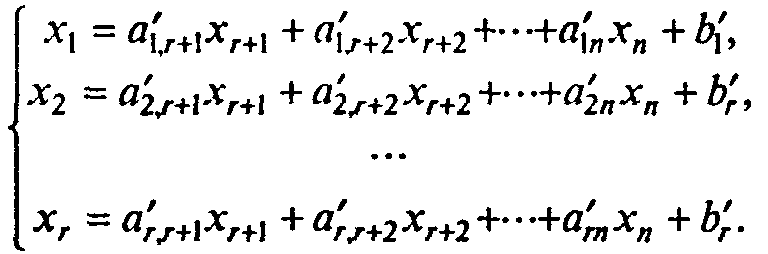
### 7.2. СИМПЛЕКС-МЕТОД

Для решения ряда задач линейного программирования существуют специальные методы. Есть, однако, общий метод решения всех таких задач. Он носит название симплекс-метода и состоит из алгоритма отыскания какого-нибудь произвольного допустимого решения и алгоритма последовательного перехода от этого решения к новому допустимому решению, для которого функция *f* изменяется в нужном направлении (для получения оптимального решения).

Пусть система ограничений состоит лишь из уравнений



и требуется отыскать минимум линейной функции (7.81). Для отыскания произвольного опорного решения приведем (7.85) к виду, в котором некоторые *r* неизвестных выражены через остальные, а свободные члены неотрицательны (как это сделать - обсудим позднее):



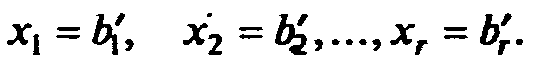
Неизвестные *x1*, *x2*, ..., *xr* - *базисные неизвестные*, набор {*x1*, *x2*, ..., *xr*} называется базисом, а остальные неизвестные {*xr+1*, *xr+2*, ..., *xn*} - *свободные.* Подставляя (7.86) в (7.81), выразим функцию

*f* через свободные неизвестные:

***f* = *c0* + *c'r*+1*xr*+1 + *c'хr*+2 +…+ *с'nxn*.**

Положим все свободные неизвестные равными нулю:





Полученное таким образом допустимое решение



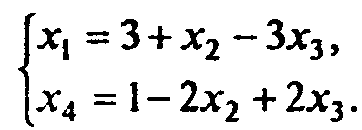
отвечает базису *x1, х2, ..., xr,* т.е. является базисным решением. Допустим для определенности, что мы ищем минимум *f*. Теперь нужно отданного базиса перейти к другому с таким расчетом, чтобы значение линейной функции *f* при этом уменьшилось. Проследим идею симплекс-метода на примере.

*Пример* 1. Дана система ограничений

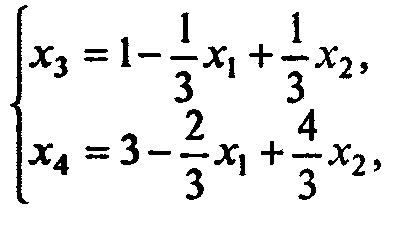
*x1 –* 3*x2 +* 5*x3 – x4 =* 2

*x1 + x2 + x3 + x4 =* 4

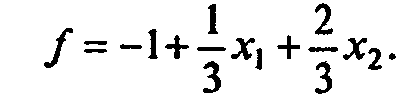
Требуется минимизировать линейную функцию *f* *= х2 – x3.* В качестве свободных переменных выберем *х2* и *х3.* Тогда данная система ограничений преобразуется к виду



Таким образом, базисное решение (3, 0, 0, 1). Так как линейная функция уже записана в свободных неизвестных, то ее значение для данного базисного решения *f* *= 0.* Для уменьшения этого значения можно уменьшить *x2* или увеличить *x3*. Но *x2* в данном базисе равно нулю и потому его уменьшать нельзя. Попробуем увеличить *x3.* Первое из уравнений имеет ограничение *x3* = 1 (из условия *x1* ≥ 0), второе - не дает ограничений. Далее, берем *x3* = 1, *х2* не меняем и получаем новое допустимое решение (0, 0, 1, 3), для которого *f =* -1 - уменьшилось. Найдем базис, которому соответствует это решение (он состоит, очевидно, из переменных *x3, x4*). От предыдущей системы ограничений переходим к новой:



а форма в новых свободных переменных имеет вид

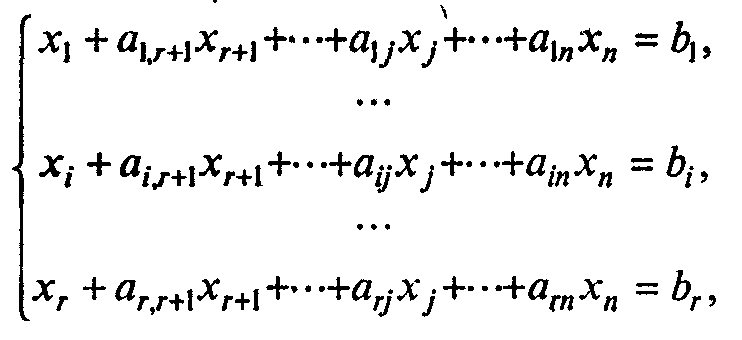


Теперь попробуем повторить предыдущую процедуру. Для уменьшения *f* надо уменьшить либо *x1,* либо *x2*, но это невозможно, так как в этом базисе *x1* *=* 0, *x2* = 0.

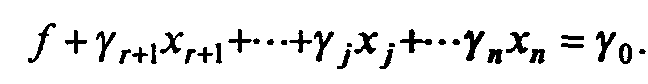
Таким образом, данное базисное решение является оптимальным, и min*f*= -1 при *x1 = 0, x2* = 0, *x3* = 1, *x4* = 3.

Приведем алгоритм симплекс-метода в общем виде. Обычно все вычисления по симплекс-методу сводят в стандартные таблицы.

Запишем систему ограничений в виде

(7.90)

а функцию *f*

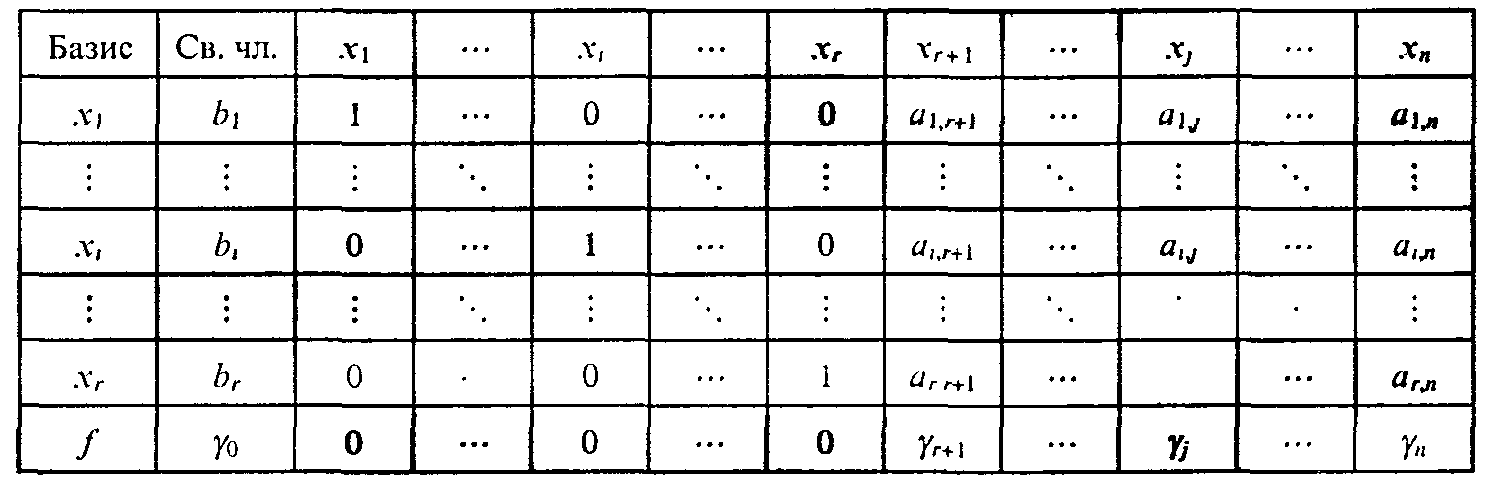
(7.91)

Тогда очередной шаг симплекс-процесса будет состоять в переходе от старого базиса к новому таким образом, чтобы значение линейной функции, по крайней мере, не увеличивалось.

Данные о коэффициентах уравнений и линейной функции занесем в табл. 7.12.

Таблица 7.12

**Симплекс-таблица**



Сформулируем алгоритм симплекс-метода применительно к данным, внесенным в табл. 7.12.

1. Выяснить, имеются ли в последней строке таблицы положительные числа *(γ0* не принимается во внимание). Если все числа отрицательны, то процесс закончен; базисное решение (*b1*, *b2*, .... *br,* 0, ..., 0) является оптимальным; соответствующее значение целевой функции *f* = *γ0*. Если в последней строке имеются положительные числа, перейти к п.2.

2. Просмотреть столбец, соответствующий положительному числу из последней строки, и выяснить, имеются ли в нем положительные числа. Если ни в одном из таких столбцов положительных чисел нет, то оптимального решения не существует. Если найден столбец, содержащий хотя бы один положительный элемент (если таких столбцов несколько, взять любой из них), пометить этот столбец и перейти к п. 3.

3. Разделить свободные члены на соответствующие положительные числа из выделенного столбца и выбрать наименьшее частное. Отметить строку таблицы, соответствующую наименьшему частному. Выделить разрешающий элемент, стоящий на пересечении отмеченных строки и столбца. Перейти к п. 4.

4. Разделить элементы выделенной строки исходной таблицы на разрешающий элемент (на месте разрешающего элемента появится единица). Полученная таким образом новая строка пишется на месте прежней в новой таблице. Перейти к п. 5.

5. Каждая следующая строка новой таблицы образуется сложением соответствующей строки исходной таблицы и строки, записанной в п. 4. которая предварительно умножается на такое число, чтобы в клетках выделенного столбца при сложении появились нули. На этом процесс заполнения новой таблицы заканчивается, и происходит переход к п. 1.

Таким образом, используя алгоритм симплекс-метода применительно к симплекс-таблице, мы можем найти оптимальное решение или показать, что его не существует. Результативность симплекс-метода гарантируется следующей теоремой (приведем ее без доказательства): *если существует оптимальное решение задачи линейного программирования, то существует и базисное оптимальное решение. Это решение может быть получено через конечное число шагов симплекс-методом, причем начинать можно с любого исходного базиса.*

Ранее мы предполагали, что если система ограничений задана в виде (7.85),топеред первым шагом она уже приведена к виду (7.86), где *bi ≥* 0 (*i* = 1, 2, ..., *r).* Последнее условие необходимо для использования симплекс-метода. Рассмотрим вопрос об отыскании начального базиса.

Один из методов его получения - метод симплексного преобразования.

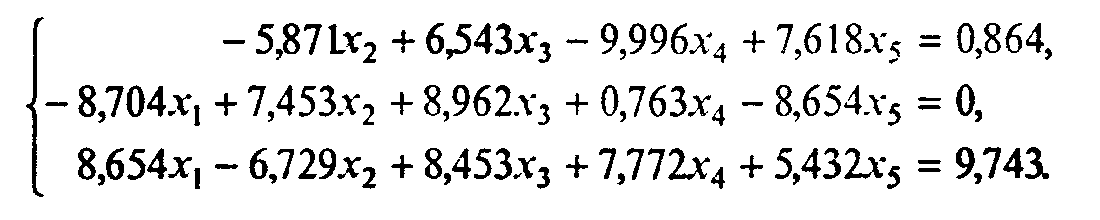
Прежде всего проверяем, есть ли среди свободных членов отрицательные. Если свободные члены не являются числами неотрицательными, то добитьсяих неотрицательности можно несколькими способами:

1) умножить уравнения, содержащие отрицательные свободные члены, на-1;

2) найти среди уравнений, содержащих отрицательные свободные члены, уравнение с максимальным по абсолютной величине отрицательным свободным членом и затем сложить это уравнение со всеми остальными, содержащими отрицательные свободные члены, предварительно умножив его на-1.

Затем, используя действия, аналогичные указанным в пп. 3 - 5 алгоритма симплекс-метода, совершаем преобразования исходной таблицы до тех пор, пока не получим неотрицательное базисное решение.

*Пример 2.* Найти исходное неотрицательное базисное решение системы ограничений



Так как условие неотрицательности свободных членов соблюдается, приступим к преобразованиям исходной системы, записывая результаты в таблицу. Согласно алгоритму просматриваем первый столбец. В этом столбце имеется единственный положительный элемент *a31*. Делим на 8,654 все коэффициенты и свободный член третьей строки, после чего умножаем каждый коэффициент на 8,704 и складываем с соответствующими коэффициентами второй строки. Первая строка преобразований не требует, так как коэффициент при неизвестном *x1* равен нулю. В результате получаем

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 0,00000  0,00000  1,00000 | -5,87100  0,68512  -0,77756 | 6,54300  17,46384  0,97677 | -9,99600  8,57990  0,89808 | 7,61800  -3,19062  0,62769 | 0,86400  9,79929  1,11584 |

Продолжая просматривать второй столбец и совершая аналогичные преобразования, имеем

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 0,00000  0,00000  1,00000 | 0,00000  1,00000  0,00000 | 156,19554  25,49013  20,79687 | 63,52761  12,52318  10,63560 | -19,72328  -4,65701  -2,99341 | 84,83688  14,30299  12,24727 |

И, наконец, на третьем шаге находим исходный базис. Его образуют неизвестные *x1*, *x2,* *x3*. Неизвестные *x4, х5* являются свободными:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 0,00000  0,00000  1,00000 | 0,00000  1,00000  0,00000 | 1,00000  0,00000  0,00000 | 0,40672  2,15588  2,17713 | -0,12627  -1,43829  -0,36733 | 0,54315  0,45815  0,95155 |

При решении задачи линейного программирования целесообразно использование компьютера. В этом случае можно составить программу, решающую задачу. Учитывая,что программирование довольно трудоемко, можно посоветовать воспользоваться для оформления результатов расчетов табличным процессором. Кроме того, если получившаяся модель задачи слишком громоздка, можно воспользоваться математическими пакетами, которые позволяют получить решение задачи линейного программирования. И, наконец, еще один возможный вариант применения компьютеров - комбинирование всех вышеуказанных способов.