#### Математическое программирование

(постановка задачи и основные определения)

*Основная* задача математического программирования состоит в минимизации вещественной функции на множестве, определенном системой ограничений типа равенства и/или неравенства.

Записывать задачу будем в следующем виде:

, где

**Определение.** *ϕ* называется *целевой функцией*.

*Ограничения или условия* записываются в виде:

, где

**Определение.** *X* называется *допустимым множеством*.

*Размерность задачи*: *n* – число переменных; *m+l* – число ограничений.

Запись min*ϕ*(*x*) означает:

1. либо найти оптимальную точку

.

**Определение.** Всякая допустимая *точка* *x*∈*X* называется *планом*; *x\** – *оптимальный план*.

1. если *x\** не существует, то найти , например, .
2. либо показать, что *X* = ∅ (допустимое множество – пусто).

**Классификация задач математического программирования**

1. Если целевая функция линейна, т.е. , где *c*∈*Rn* и ограничения линейны, т.е. имеют вид:

*Ax* ≤ *b*, где *A* – *m*×*n* – матрица, *b*∈*Rm*

*Gx*=*h*, где *G* – *l*×*n* – матрица, *h*∈*Rl*,

то это *задача линейного программирования* (иначе, нелинейного, например, квадратичного).

1. Если целевая функция *ϕ* – выпукла и допустимое множество *X* – выпуклое (*fi*, *gk* – выпуклые функции), то это *задача выпуклого программирования*.
2. Если по условию переменные – целые числа, т.е. , то это – *задача целочисленного программирования* (в данном курсе не рассматривается).

**Спецификация задач математического программирования**

* как правило, методы классического анализа для отыскания условных экстремумов неприменимы (экстремум достигается в угловых точках допустимого множества).
* большое количество переменных и ограничений в практических задачах, так что задача перебора точек, подозреваемых в экстремальности, может оказаться нетривиальной.

⇒ целью математического программирования является *создание*, где это возможно, *аналитических методов* определения решения, а при отсутствии таких методов – *создание эффективных вычислительных способов* получения *приближенного решения*.

Наименование предмета – *математическое программирование* – связано с тем, что целью решения задач является *выбор программы действий*.

**Cведения о выпуклых множествах**

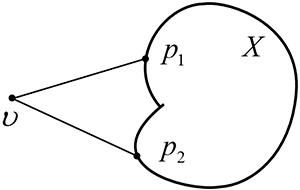
**Определение 1.**Множество  называется *выпуклым*, если   (выпуклое множество содержит отрезок, соединяющий две любые его точки).

**Определение 2.**Точка *x*∈*X* называется *внутренней*, если существует , где  – *ε*-окрестность точки *x*, т.е. точка *x*∈*X* называется *внутренней*, если существует такая её окрестность, все точки которой принадлежат *X*. И наоборот, если найдется такая *ε*-окрестность точки *υ*, которая не содержит ни одной точки множества *X* – такая точка называется *внешней* по отношению к множеству *X*.

**Определение 3.** Точка *x*∈*X* называется *граничной*, если ∀*ε* >0 существует и существует , т.е. в любой окрестности точки *x* содержатся как точки, принадлежащие множеству *X*, так и точки, не предлежащие этому множеству.

**Определение 4.** *Проекцией* точки *υ* на множество *X* называют такую точку *p*∈*X*, что

.

При этом,  называют "расстояние" от точки *υ* до множества *X*. Ясно, что если *υ*∈*X*, то *p*= *υ*. Если же *υ*∉*X*, и множество *X* – открыто, то проекция *p* не существует. Если множество *X* – не выпукло, то проекция может быть не единственной.

Верно следующее утверждение:

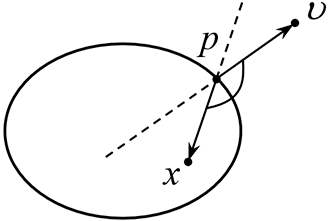
**Лемма 1.** Пусть X – выпуклое замкнутое множество из Rn, , тогда:

1) Любая точка *υ*∈*Rn* имеет и притом единственную проекцию на это множество;

2) Для того чтобы точка *p*∈*X* была проекцией точки *υ* на множество *X*, необходимо и достаточно выполнения неравенства  для ∀*x*∈*X*.

**Доказательство.**

тупой угол (*x*–*p*,*υ*–*p*) ≤ 0)



*Докажем первое утверждение леммы.*

Рассмотрим функцию *g*(*x*) вида

.

Поскольку *g*(*x*) сильно выпукла, то по следствию из теоремы об ограниченности множеств Лебега для сильно выпуклой функции можно утверждать, что *g*(*x*) достигает своей нижней грани на Х в единственной точке *p*′∈*X*.

Это означает, что

,

Причем равенство здесь возможно, только когда *x*=*p*′ (т.к. *p*′ единственная точка), а тогда *p*=*p*′, что и требовалось доказать.

*Докажем второе утверждение леммы.*

*Необходимость*. Пусть *p* – проекция точки *υ* на *X*.

Возьмем произвольную точку *x*∈*X,* отличную от *р,* рассмотрим точку . Ввиду выпуклости множества Х для ∀*α*∈[0,1] точка *z*∈*X.*

Так как , и из определения проекции следует, что , то .

Поскольку это неравенство справедливо для ∀*α*∈[0,1], то .

Переходя к пределу при *a->0,* имеем , что и требовалось доказать.

**Достаточность.** Пусть верно  ∀*x*∈*X*, тогда ∀*x*∈*X* верно

,

т.е. точка *p* является проекцией точки *υ* на *X*.

**Определение 5.** Гиперплоскостью в *Rn* называется множество вида ,

где  – вектор ∈ *Rn,* .

**Свойства гиперплоскостей**

1. Это множество всегда *не пусто*: если, например *ci*≠ 0, то точка *x*0 с координатами ,  удовлетворяет равенству , т.е. .

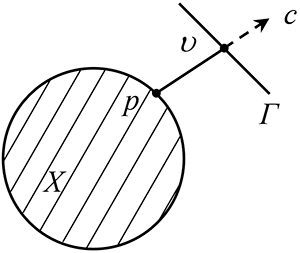
2. Пусть *x*0 – ∀(⋅) из *Гc,λ* , т.е. , тогда .

Известно, что два вектора *a*,*b*∈*Rn* – ортогональны, если (*a*,*b*)=0 ⇒ гиперплоскость *Гc,λ* состоит из тех и только тех точек *x*, для которых вектор *x*–*x*0 ортогонален вектору *с*. Вектор *с* называют *нормальным* вектором гиперплоскости *Гc,λ* .

3. В пространстве *Rn* гиперплоскость определяет два полупространства:

 и .

**Лемма 2** (Теорема отделимости).Для любого выпуклого и замкнутого множества *X* и любой точки *υ*, не принадлежащей множеству *X*, существует такая гиперплоскость *Г*, что  и  для ∀*x*∈*X*.

Очевиден геометрический смысл теоремы: существует проходящая через точку *υ* гиперплоскость *Г* такая, что *X* лежит в одном из полупространств, определенных *Г*.

*Доказательство.* Пусть *p* – проекция *υ* на *X*.

Определим , и рассмотрим

гиперплоскость ,

для которой выполняется первое утверждение леммы 2.

По лемме 1, если *p* – проекция, то ∀*x*∈*X* справедливо .

Поскольку точка *υ*∉*X*, то расстояние .

Итак, имеем для ∀*x*∈*X* : 

, *ч.т.д.*

**Теорема** (об опорной гиперплоскости).

В любой граничной точке *x*0 выпуклого множества существует *опорная гиперплоскость*, т.е. существует *c*≠0 и λ :

, и для всех *x*∈*X* .

|  |  |
| --- | --- |
| 33  Опорная гиперплоскость единственна (если существует касательная гиперплоскость, то она совпадает с опорной). И в этом случае опорная гиперплоскость единственная. | 34  Понятие опорной гиперплоскости – шире касательной. В точке *x*0 не существует касательной, но существуют опорные гиперплоскости, причем в качестве вектора *c* можно выбрать любой, лежащий между *c*1,*c*2. |

**Доказательство.**Рассмотрим последовательность {*υk*} – внешних точек относительно  (т.е. по определению сходимости ).

По лемме 2 (теорема отделимости) существует последовательность гиперплоскостей

, где  и .

Т.к. длину *ck* можно выбирать произвольно, то, не умаляя общности, можно считать, что . Не меняя обозначений, считаем, что .

Далее воспользуемся леммой Больцано-Вейерштрасса.

**Лемма**(Больцано-Вейерштрасса). Из любой ограниченной последовательности всегда можно извлечь такую частичную последовательность, которая сходилась бы к конечному пределу.

Рассмотрим .

Переходя к пределу в соотношении, определяющем *Гk*, получим гиперплоскость:

, где *λ*=(*с*, *x*0).

А, переходя к пределу в соотношении  ∀*x*∈*X*, получим:

 для ∀*x*∈*X* , *ч.т.д.* (равенство возникает, поскольку *x*0∈*X*, а (*c*, *x*0) = *λ*) .

**Теорема** (о разделяющей гиперплоскости).

Пусть *X*0 – множество внутренних точек выпуклого множества *X*; *Y* – выпуклое множество.

Если  (множество *Х*0 – не пусто) и  (не пересекается с другим множеством), то для множеств *X* и *Y* существует *разделяющая гиперплоскость*, т.е.

существует *c* ≠ 0 : ∀*x*∈*X*, ∀*y*∈*Y* справедливо соотношение (*c*,*y*) ≤ (*c*,*x*).

**Доказательство.**Рассмотрим множество .Это множество выпукло. Действительно,

.

Точка *z*=0 не является внутренней точкой множества *Z* (т.к. ).

Поэтому существует *с* ≠ 0: . Это неравенство справедливо:

- по теореме об опорной гиперплоскости, если точка *z*= 0 – граничная для множества *Z;*

- или по теореме о разделяющей гиперплоскости, если точка *z*= 0 – внешняя (тогда неравенство строгое).

⇒ .

Последнее неравенство остается справедливым и для ∀*y*∈*Y* и ∀*x*∈*X*, поскольку предельный переход не нарушает нестрогих неравенств, *ч.т.д.*

Введем два определения:

**Определение 1.** Точка *x* множества *X* (*x*∈*X*) называется *угловой* (*или крайней*) точкой, если в *X* не существует таких точек *x*′ и *x*″, *x*′ ≠ *x*″, что , при некотором *α*∈(0,1).

*Геометрически*: точка *x* – крайняя в *X*, если её нельзя поместить внутрь отрезка, концы которого лежат в *X*.

Например,

* у треугольника крайние точки – вершины;
* у луча – начало;
* у круга – все точки окружности;
* прямая, гиперплоскость – крайних точек не имеют.

**Определение 2.** Точка *x*0∈*X* называется *выпуклой комбинацией* точек *x*1,…, *xn*∈*X*, если существуют , существуют *xi*∈*X, i=1,…N,* такие, что:



**Теорема** Крейна–Мильмана (о представлении).

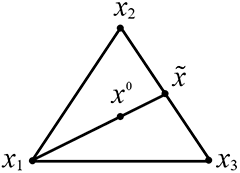
Пусть *X* – выпуклое, замкнутое, ограниченное множество, тогда ∀*x*0∈*X* может быть представлена в виде выпуклой комбинации конечного числа угловых точек множества, т.е. ∀*x*0∈*X* существуют *αi* ≥ 0, существуют *xi*∈*X* – угловые точки:



**Доказательство.** Индукция по размерности пространства *n*.

При *n*= 1 *X* – отрезок ⇒ утверждение теоремы очевидно.





Предположим, что для *n*= *k* – 1 теорема справедлива.

Пусть *X*∈*Rk*. Возможны два случая:

1) *x*0 – граничная точка *X*.

Построим в этой точке гиперплоскость, опорную к *X* (существует по теореме об опорной гиперплоскости):

, где *λ*= (*c*, *x*0) – опорная гиперплоскость.

Рассмотрим множество *X*0 = *X*∩ *Гc*,*λ..* Оно, как пересечение выпуклого, замкнутого, ограниченного множества *X* с выпуклым замкнутым множеством *Гc*,*λ*, само выпукло, замкнуто и ограничено.

Кроме этого, .

По индукционному предположению существуют  – угловые точки *X*0:



Покажем, что  являются угловыми точками и для множества *X*.

Предположим противное, т.е. что некоторая точка *xi* не является угловой для множества *X*. Это означает, что существует  и .

Т.к. , то



и т.к. *Гc,λ* – опорная к *X*, то

 (\*)

Поскольку 0 < *α*< 1, можно записать:

⇒.

Из последнего соотношения следует, что (*c*,*x*′) ≥ (*c*,*x*0), но

,

Поскольку .

Аналогично можно показать, что *x*″∈*X*0 ⇒ противоречие с тем, что *xi* – угловая точка *X*0  ⇒ *xi* – угловая точка *X*, *ч.т.д*

2) Пусть теперь *x*0 – внутренняя точка множества *X*. Проведем через *x*0 прямую *l*. Пересечение *l*∩ *X* является отрезком с концами *x͂* и *x͌*, принадлежащими границе множества *X*, и, поскольку *x*0 – внутренняя точка *X* ⇒ существует *α*∈(0,1):

.

Поскольку для граничных точек *x͂* и *x͌* теорема верна, то верна она и для *x*0. Действительно, для граничных точек имеют место соотношения:



,

где все *yi* и *zi* – угловые точки множества *X*.

А тогда , *ч.т.д.*

*Замечание*: Можно доказать, что в указанном представлении число угловых точек не превосходит величины *n*–размерности пространства.

**Линейное программирование**

Задача линейного программирования:

(1)

Здесь *cj*, *aij*, *bi* – заданные числа, причем не все *cj* и *aij*= 0.

Это – задача линейного программирования (ЗЛП) со смешанными ограничениями. К задачам такого вида (1) сводятся многие прикладные задачи технико-экономического содержания. Из общей задачи линейного программирования обычно выделяют и исследуют два её подкласса – *основную* задачу и *каноническую*.

Если *k*= *m* (только ограничение неравенства) и *s*= *n* (прямые ограничения) накладываются на все элементы вектора, то это *основная (стандартная) форма ЗЛП*.

(2)

*A*– *m*× *n* – матрица условий; *b*– *m* ­– вектор ограничений.

Если *k*= 0 (только ограничения равенства) и *s*= *n*, то это – *каноническая форма ЗЛП*.

(3)

Задачи в формах (1), (2), (3) могут быть сведены друг к другу, т.е. приведены к эквивалентной задаче (с тем же множеством решений).

*Сведение* (1) *к* (2):

Обозначим  – множество всех ограничений;

 – множество ограничений неравенств;

 – множество ограничений равенств.

Аналогично:

Обозначим ;

 – множество прямых ограничений;



Заметим, что задача (1) приводится к виду:



Здесь:





*Идея*: Нужно (*m* – *k*)–равенств заменить неравенствами:



Ввести (*n*– *s*)–прямых ограничений:



Т.о., делаем замену переменных , где элементы вектора



Тогда задача (1) приводится к виду:

(2)

Это задача в основной форме (2) с размерностью  и числом ограничений

.

*Сведение* (2) *к* (3)*:*

Введем *m*–дополнительных переменных и рассмотрим задачу в пространстве *Rn*+*m*:

;

Тогда (2) можно записать в виде:

(3′)

Эта задача в канонической форме (3) с размерностью  и *m*-ограничениями.

Нетрудно убедиться в том, что множество решений, рассмотренных выше задач совпадают, либо пусты одновременно.

**Геометрическая интерпретация основной задачи линейного программирования**

Рассмотрим задачу ЛП в форме (2), т.е. основную ЗЛП:



Пусть ⇒ задача сводится к виду:

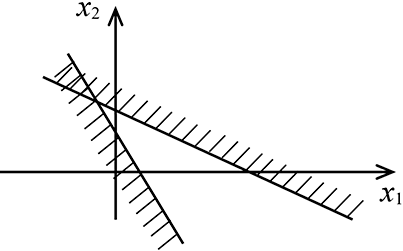


Введем множества:

 – положительные квадрант плоскости.

 – полуплоскость, образованная прямой .

Ясно, что множество *X* является пересечением множеств  и возможны следующие случаи:



1) Может случиться, что это пересечение пусто, тогда задача теряет смысл (а).

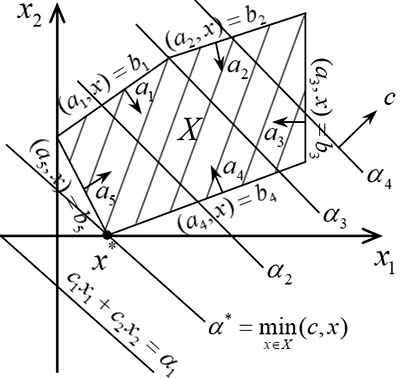
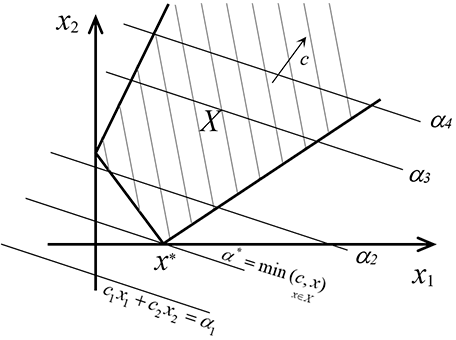
(a)

2) Если множество *X* не пусто, то оно образовано пересечением конечного числа полуплоскостей ⇒ множество *X* представляет собой *выпуклое многоугольное множество*, границей которого является ломаная, составленная *из отрезков каких-либо координатных осей* и *прямых* . Это многоугольное множество может быть ограниченным (выпуклый многоугольник) (б) и неограниченным (в).

Рассмотрим уровни минимизируемой функции, т.е. .

При изменении *α* от –∞ до +∞ прямая, перемещаясь параллельно самой себе, "зачертит" всю плоскость. При этом направление вектора *с* задает движение линии уровня по направлению возрастания функции (*c, x*).

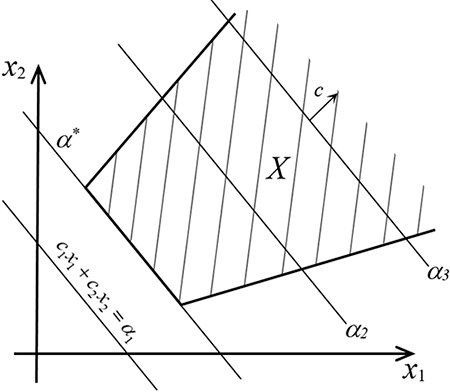
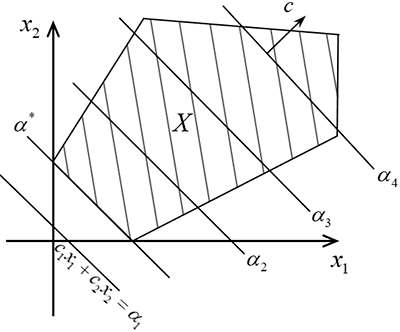
Если *X* – многоугольник (б), то при изменении *α* от –∞ до +∞ прямая, соответствующая линии уровня, при некотором значении *α*\* впервые коснется *X* (выпуклого многоугольника) и будет иметь с этим множеством *X* общую точку *x*\*, т.е. *x*\* – решение задачи.



(в)

(б)

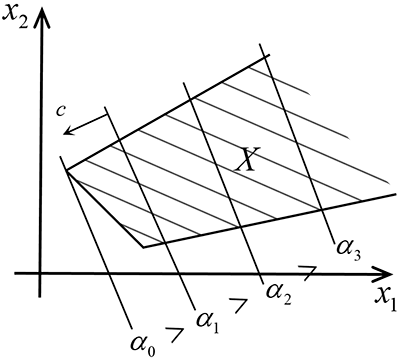
Возможен случай, когда при первом касании линией уровня множества *X*, общей окажется целая сторона многоугольника, тогда решением будет целая прямая. Это может случиться, когда множество *X* имеет сторону, перпендикулярную вектору *c* ((г) и (д)).



(г)

(д)

Если многоугольное множество *X* не ограничено, то возможна ситуация, когда прямая линия уровня при всех  имеет общую точку с множеством *X* (е), тогда . В этом случае первого касания с прямой нет – задача не имеет решения.



(е)

Из рассмотренных случаев ясно, что ЗЛП может не иметь ни одного решения ((а) и (е)), может иметь единственное решение ((б) и (в)) и, наконец, может иметь бесконечное множество решений (линия уровня параллельна одной из граней допустимого множества).

На примере рассмотренной выше ЗЛП нетрудно увидеть, что, *если задача имеет решение*, то среди решений найдется хотя бы одна угловая точка многоугольного множества . Ниже мы увидим, что это не случайно ­– и в более общей ЗЛП, оказывается, нижняя грань минимизируемой функции достигается на *X* в угловой точке множества.

Итак, рассмотрим основную задачу линейного программирования, т.е. задачу в форме (2):



**Теорема.**Допустимое множество в задаче (2) – выпукло и замкнуто.

**Доказательство.**

*Замкнутость.* Рассмотрим последовательность

, т.е.  для ∀*k*.

Пусть (непрерывность линейной функции).

Переходя к пределу в неравенстве: , получим , и соответственно,

,

т.е. множество содержит свои предельные точки, следовательно, оно замкнуто.

*Выпуклость.* Пусть . Это означает, что



Рассмотрим  и покажем, что  , т.е. множество Х - выпукло.

Действительно:

.

Аналогично:

, *ч.т.д.*

*Теорема доказана*.

**Теоремы об оптимальных точках основной задачи линейного программирования**

**Теорема 1*.*** Если допустимое множество задачи (2) не пусто (*X* ≠ ∅) и целевая функция ограничена снизу на *X*, то существует *x*\*– оптимальная точка, причем *x*\* лежит на границе множества *X*.

**Доказательство.** (индукция по *n*)

а) Пусть *n*= 1. В этом случае утверждение теоремы очевидно:

допустимое множество – отрезок  или луч, а целевая функция - *ϕ*(*x*)­ = *cx*.

Если *X* – отрезок, то min достигается и находится на границе.

Если *X* – луч, то *c*> 0 (в силу ограниченности *ϕ*(*x*) снизу на *X)* ⇒ min достигается на границе луча.

б) Пусть теорема верна для *n* – 1, и докажем её для *n*.

Рассмотрим грани допустимого множества *X*:

 – ограничения - неравенства;

и "координатные" грани:

 – прямые ограничения.

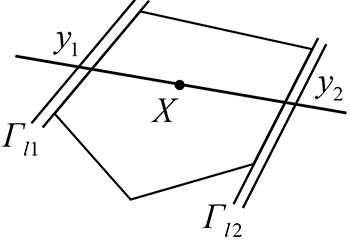
Т.к. , *Гl* – линейное пространство размерности (*n*– 1), выпуклое замкнутое множество, то по индукционному предположению имеем (*m*+ *n*) - точек (на ∀ грани – по одной), где *ϕ*(*x*) достигает min.

Пусть *x*\* – одна из них, где функция минимальна. Докажем, что *x*\* – оптимальная точка на множестве *X*.

Предположим *противное*, пусть существует .

Можно утверждать, что точка *x* – внутренняя точка множества *X*. Действительно, если бы она была граничной, то для нее не могло бы выполняться .

А тогда через точку *x* можно провести прямую, пересекающую две граничные гиперплоскости ⇒ существуют  такие, что:



*y*1,*y*2 – граничные и 

⇒ существует и существует  ⇒** – по выбору точки *x*\*.

Получаем:

 – противоречие!

*Теорема доказана*.

**Теорема 2.**Если допустимое множество задачи (2) не пусто и целевая функция задачи (2) ограничена снизу на допустимом множестве *X*, то среди оптимальных точек задачи линейного программирования есть крайняя (угловая).

**Доказательство.**

Пусть *x*\* – оптимальная точка ЗЛП (она существует по теореме 1).

1) Предположим, что *X* – ограниченное множество. Но *X* – выпукло и замкнуто (по доказанной выше теореме). Тогда по теореме Крейна-Мильмана (о представлении), точка *x*\* может быть представлена в виде выпуклой комбинации конечного числа угловых точек множества *X*, т.е. существует *y*1,…, *yk* – крайние точки в *X* и существуют *λ*1,…, *λk*, такие, что:



Поскольку *ϕ*  - линейная функция, можно записать:

.

Докажем, что в этом случае , т.е. тем самым мы докажем, что среди оптимальный точек *x*\* есть крайние - 

Предположим, что  (\*)

Тогда: 

⇒ с учетом (\*) имеем .

Далее будет доказывать по индукции.

Пусть для некоторого , справедливо: .

Докажем, что .

Действительно, 

Min Min среди суммируемых по (\*)

Тогда, перенеся в левую часть неравенства, получим

,

а тогда, с учетом (\*), имеем

, *ч.т.д.*

2) Предположим, что *X* – неограниченное множество, а *x*\* – оптимальная точка ЗЛП.

Рассмотрим вектор

.

Пусть . Введем новое ограничение .

Тогда точка *x*\* лежит в новом допустимом множестве, при этом, т.к. , то новое допустимое множество ограниченно. ⇒ По доказанному выше существует :

.

Осталось доказать, что существует , т.е. крайняя оптимальная точка не лежит на новой гиперплоскости, т.е. *zj* – крайняя точка множества *X* и оптимальна.

Имеем:

, *ч.т.д.*

(Условие того, что точка *x*\* лежит на новой гиперплоскости (*d*,*x\**) = *μ*+ 1). *Теорема доказана*.

Итак, *для решения задачи линейного программирования надо искать крайние точки*. Ранее было введено геометрическое определение крайней точки. Для того чтобы уметь находить её, следует ввести её *алгебраическое её определение*.

**Характеристика крайних точек**

Рассмотрим основную форму ЗЛП, когда допустимое множество *X* имеет вид:

 – всего (*m* + *n*) ограничений.

**Определение.**Если в точке *x* для некоторых ограничений выполняются равенства {(*Ai*, *x*) = *bi* или *xj* = 0 }, то они называются *активными*, остальные ограничения называются *пассивными*.

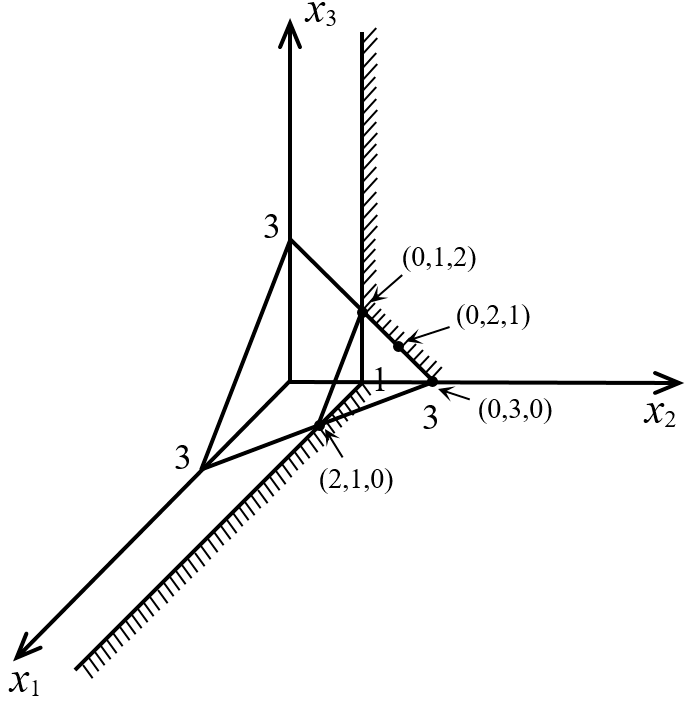
Рассмотрим матрицу ограничений: 

Для ∀*x*∈*X* определим множество индексов активных ограничений:



Обозначим через *Bx* матрицу, составленную из векторов-строк матрицы *B*, соответствующую активным ограничениям точки *x*∈*X*.

**Пример.** Пусть , где *m* = *2*, *n* = *3*;



Матрица *B* имеет вид: 

Единичная матрица, соответствующая прямым ограничениям

(0, 1, 2) – крайняя точка, для неё:



(2, 1, 0) – крайняя точка:



(0, 3, 0) – крайняя точка:



(0, 2, 1) – не крайняя точка:



Для внутренних точек *x* матрицы *Bx* будет состоять из пустых строк.

**Теорема*.*** Для того чтобы допустимая точка *x* задачи (2) была крайней, необходимо и достаточно, чтобы ранг матрицы *Bx* был максимальным, т.е. равен *n*, т.е. *x*∈*X* – крайняя точка ⇔ *rang Bx* = *n*.

**Доказательство.**

*Достаточность.* Пусть . Это означает, что из соотношения , следует, что *z*= 0. Докажем, что тогда точка *x*∈*X* – крайняя.

Предположим обратное, т.е. пусть существует , где *λ*∈(0,1)

Обозначим  (т.е. вектор *bx* состоит из компонент вектора *b* и нулей).

Т.к. *x*1,*x*2 – допустимые точки, то  и сложим



По определению *bx* в этом соотношении должно быть равенство

⇒ 

Аналогично можно показать, что  ⇒ *x*=*x*2.

Значит наше предположение не верно, точка *x* крайняя, *ч.т.д.*

*Необходимость.* Пусть точка *x* – крайняя. Предположим, что , т.е. существует .

Зафиксируем такую точку *z*∈*X* и при малом *ε* > 0 рассмотрим точки: 

|  |  |
| --- | --- |
| Для *x*1 и *x*2 выполняются активные ограничения | По определению точки *z* имеем:    ( т.к. для крайней точки справедливо  и по предположению ) Отсюда следует, что активные ограничения для точки *x*1 и точки *x*2 верны. |

|  |  |
| --- | --- |
| Для *x*1 и *x*2 выполняются пассивные ограничения | *ε* можно выбрать таким образом, чтобы и пассивные ограничения сохранялись: ,  , значит можно выбрать *ε* такимобразом,  чтобы выполнялось неравенство  Аналогично можно показать, что и прямые пассивные ограничения сохраняются: |
|  |  |

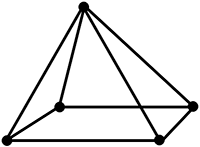
Это означает, что *x*1,*x*2 – допустимые точки. Но  – противоречит тому, что *x* – крайняя точка. Т.е. не существует такого вектора , *ч.т.д*.

**Замечания.**

1. В крайней точке пересекаются не менее чем *n* гиперплоскостей (граней допустимого множества). Для *n* = 2 – две прямых, для *n* = 3– три плоскости и т.д.

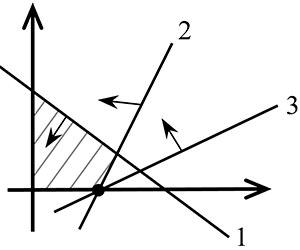
**Определение.**Крайняя точка называется *невырожденной*, если она лежит на пересечении ровно  гиперплоскостей. В противном случае она называется *вырожденной*. Задача линейного программирования, у которой существует вырожденная точка, также называется вырожденной.

*Примеры вырожденных точек*:



вершина 4-х

угольной пирамиды



вырожденная точка

Далее, если не оговорено особо, будем рассматривать невырожденные задачи ЛП.

2. Т.к. ограничений (*m*+ *n*), а в крайних точках сходится *n*-гиперплоскостей, то всего *крайних точек* может быть не более  – *конечное число*. (Число сочетаний из n+m по n).

Доказанные выше теоремы (о существовании решения ЗЛП и алгебраической характеристике её крайних точек) означают, что для поиска решения основной задачи ЛП, *достаточно перебрать лишь крайние точки допустимого множества X*, число которых *конечно*. Крайние точки могут быть найдены с использованием теоремы о характеристике крайних точек за конечное число арифметических операций.

*Выводы из 3-х теорем.*

1. *Справедлива следующая альтернатива:*

*- либо целевая функция на допустимом множестве не ограничена снизу;*

*- либо существует крайняя оптимальная точка.*

*2. Количество крайних точек допустимого множества конечно.*

Таким образом, вышедоказанные теоремы обосновывают принципиальную возможность решения задачи ЛП за конечное число шагов методом полного перебора крайних точек.

Однако "конечное" – не значит "малое". При сколько-нибудь больших *m* и *n* этот простой метод требует огромной вычислительной работы.

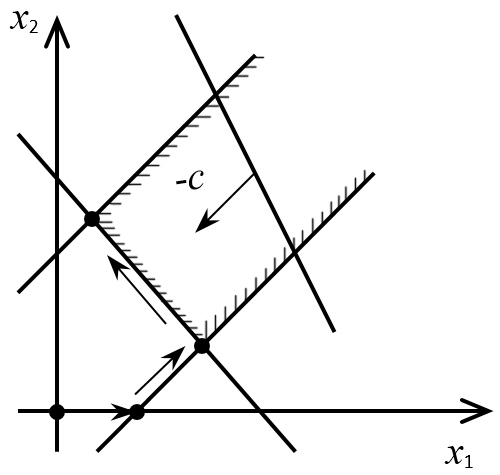
Следовательно, естественным образом подходим к основной идее *симплекс-метода* – полный перебор следует заменить упорядоченным, разумным.

Название метода связано с тем, что он впервые разрабатывался применительно к ЗЛП, в которых множество *X* представляло собой симплекс в . Затем метод был обобщен на случай более общих множеств *X*, но первоначальное название так и сохранилось. В литературе его ещё называют *методом последовательного улучшения плана*.

Итак, *симплекс метод* – это алгоритм преобразования таблицы (состоящей из коэффициентов *aij*, *bj*, *c­j*), основанных на *методе Жордановых исключений* при решении системы линейных уравнений.

Метод, состоит из 2-х этапов:

1. *поиск крайней точки*, в результате которой могут быть три ситуации:



* крайней точки нет (т.е. *Х* = ∅);
* крайняя точка не найдена;
* крайняя точка найдена, в этом случае начинается второй этап:

1. *перебор крайних точек и поиск оптимальной*:

* оптимальной точки нет (т.е. *ϕ*(*x*) не ограничена снизу на *X*);
* оптимальная точка не найдена;
* оптимальная точка найдена, на этом алгоритм симплекс-метода кончается.

*Переход к следующей* крайней точке в поисках оптимальной осуществляется *исходя из предположения, что значение целевой функции уменьшается*.

Поэтому, поскольку:

* число крайних точек конечно и среди них обязательно есть решение задачи (если оно существует),
* возврат к уже просмотренным точкам невозможен, то за *конечное число* итераций эта процедура приведет к решению, либо к выводу о том, что *X* = ∅.

Теоретически не исключается ситуация, когда метод пройдется по всем крайним точкам множества *X* (и такие патологические примеры построены). Однако, как показывает практика, для большинства задач количество итераций симплекс-метода находится в пределах от *m* до *2m*.

**Алгоритм симплекс-метода решения основной ЗЛП**

Рассмотрим основную ЗЛП:

,

*A*–(*m*× *n*) – матрица, *b*–(*m* × 1) – вектор

,

*Ai*–(1 × *n*) – строка, *bi* – число, *i*∈1,…, *m*.

Введем *m*-переменных: .

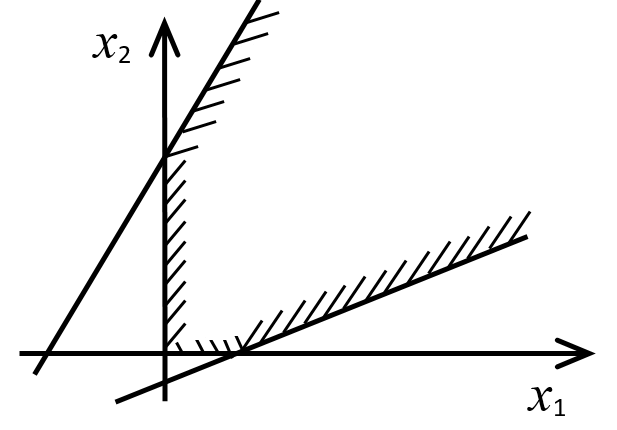
Итак, допустимое множество *X* ограниченно (*m* + *n*) – гиперплоскостями: .

1. **Алгоритм поиска крайней точки**

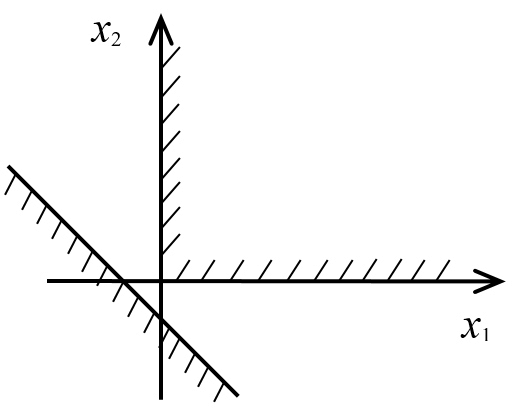
Имеем систему:



Т.к. в крайней точке неравенство заменяется равенством, то используем метод Жордановых исключений для решения системы линейных уравнений.



1. Если ∀*i* –*bi* > 0, то крайняя точка найдена, это точка *х* = 0.
2. Если существуют *s*: –*bs*< 0, торассмотрим коэффициенты *as*1,…, *asn*. Если все они ≤ 0, то





⇒ допустимое множество пусто, крайних точек нет.

1. Иначе: при  существует . Тогда делается один шаг Жордановых преобразований, который состоит в замене координат . Из s-го уравнения:

.

Подставим *xr* в другие уравнения системы:

,

где

 (1)

В новом *s*-м уравнении:

 (2)

В табличной форме:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | *Внебазисные*  *переменные* | | |  |
|  | *x*1 | *xr* | *xn* | –*b* | → |  | *x*1 | *ys* | *xn* | –*b* |
| *Базисные переменные* | *y*1 | *a*11 | *a*1*r* | *a*1*n* | –*b*1 | *y*1 | *ã*11 | *ã*1*r* | *ã*1*n* | –*b̃*1 |
| *ys* | *as*1 | *asr* | *asn* | –*bs* | *xr* | *ãs*1 | *ãsr* | *ãsn* | –*b̃s* |
| *ym* | *am*1 | *amr* | *amn* | –*bm* | *ym* | *ãm*1 | *ãmr* | *ãmn* | –*b̃m* |
|  |  |  |  |  |  |  |  |  |  |  |  |

1) Обратим внимание на то, что теперь

.

**Определение.** Элемент *asr* называется *разрешающим*.

Зависимые координаты  (в левом столбце), называются *базисными*.

Независимые координаты  (в верхней строке), называются *внебазисными*.

Один шаг Жордановых исключений – это замена базиса. *Крайняя точка найдена: все независимые координаты равны* 0, *все* .

2) Для того чтобы последовательно приближаться к крайней точке, необходимо чтобы ∀ шаг увеличивал (не уменьшал!) число положительных компонент вектора *b*.

Для этого, в пункте в) фиксируем столбец *r* (для которого при существует ) и в нём выбирается такая строка *s*, т.е. такой разрешающий элемент *asr*, чтобы

,

или *отрицательное отношение было бы максимальным среди всех отрицательных отношений.*

Покажем, что при таком выборе разрешающего элемента, число положительных компонент вектора *b* не уменьшится.

1. Если компонента корректируется по формуле (2), то, т.к.  имеем .

(показали ранее).

1. Пусть компонента вектора *b* корректируется по формуле (1). Тогда, нас интересует случай, когда . Хочется, чтобы .
   * пусть  и 

, *ч.т.д.*

* + пусть  и 

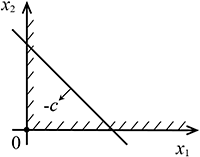
, *ч.т.д.*

Таким образом, алгоритм сходится к крайней точке за конечное число шагов в предположении, что среди крайних точек нет вырожденных. На практике это означает, что ∀*i* *bi*≠ 0.

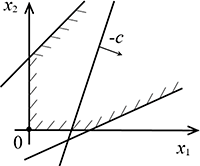
1. **Алгоритм поиска оптимальной точки**

Продолжаем преобразование таблицы. Т.к. от одного базиса всегда можно перейти к другому, то считаем, что в начале этого этапа (в верхней строке) находятся независимые координаты *x*1,…, *xn*, при этом, точка *x* = 0 (т.е. ) – крайняя.

1. Если ∀*j* *cj* ≥ 0, то *оптимальная точка найдена*. Это точка *x*= 0. Действительно  достигает min в нуле.



1. Если существует , то рассмотрим коэффициенты *alr*,…, *amr*. Если все они ≥ 0, то оптимальной точки нет. *Целевая функция не ограничена снизу на* *X*. Действительно, для ∀*i* в неравенстве , *xr* может неограниченно расти. С другой стороны, при росте *xr*, *ϕ*(*x*) – убывает .



1. Иначе: при  существует . Делается один шаг Жордановых исключений, т.е. меняется базис относительно разрешающего элемента . Предположим, что целевая функция имела вид:

,

подставим в нее *xr,* выраженный из *s*-го уровня , где

 (3)

Остальные элементы вычисляются по формулам (1) и (2).

В табличной форме:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | *x*1 | *xr* | *xn* | –*b* | → |  | *x*1 | *ys* | *xn* | –*b* |
| *y*1 | *a*11 | *a*1*r* | *a*1*n* | –*b*1 | *y*1 | *ã*11 | *ã*1s | *ã*1*n* | –*b̃*1 |
| *ys* | *as*1 | *asr* | *asn* | –*bs* | *xr* | *ãs*1 | *ãsr* | *ãsn* | –*b̃s* |
| *ym* | *am*1 | *amr* | *amn* | –*bm* | *ym* | *ãm*1 | *ãmr* | *ãmn* | –*b̃m* |
| *ϕ* | *c*1 | *cr* | *cn* | *α* | *ϕ* | *с̃*1 | *с̃r* | *с̃n* | *α̃* |

Необходимо обеспечить, чтобы ∀ следующий шаг преобразований не ухудшал достигнутого, т.е. чтобы правый столбец таблицы оставался положительным (–*bi* > 0) и чтобы функция *ϕ* – уменьшалась (количество положительных компонент вектора "*c*" может меняться).

Для того, чтобы двигаться по крайним точкам к точке min функции *ϕ*, аналогично случаю поиска крайней точки, в пункте в) фиксируют столбец *r* и в нем выбирают такую строку *s*, т.е. такой разрешающий элемент *asr*, чтобы

.

Выше было показано, что при таком выборе *asr* (выше) количество положительных компонент вектора *b* не уменьшается.

При этом, элемент , а значение функции 

, *ч.т.д.*

Таким образом, алгоритм выбора оптимальной точки также *сходится к оптимальной точке за конечное число итераций*, для невырожденной ЗЛП (среди компонент вектора "*b*" не должно быть нулевых).

Координаты оптимальной точки определяются следующим образом:

* если *xj* находится на *i*-ом месте левого столбца, то его значение равно *bi*;
* если *xi* находится на *j*-ом месте верхней строки, то его значение равно 0.

Далее рассмотрим несколько примеров:

Представим формулы для пересчета таблицы в более компактном виде. Итак,

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | *x*1 | *xr* | *xn* | –*b* | → |  | *x*1 | *ys* | *xn* | –*b* |
| *y*1 | *a*11 | *a*1*r* | *a*1*n* | –*b*1 | *y*1 | *ã*11 | *ã*1*r* | *ã*1*n* | –*b̃*1 |
| *ys* | *as*1 | *asr* | *asn* | –*bs* | *xr* | *ãs*1 | *ãsr* | *ãsn* | –*b̃s* |
| *ym* | *am*1 | *amr* | *amn* | –*bm* | *ym* | *ãm*1 | *ãmr* | *ãmn* | –*b̃m* |
| *ϕ* | *c*1 | *cr* | *cn* | *α* | *ϕ* | *с̃*1 | *с̃r* | *с̃n* | *α̃* |

Для элементов "разрешающей" строки:



Для элементов "разрешающего" столбца:



Для остальных элементов:



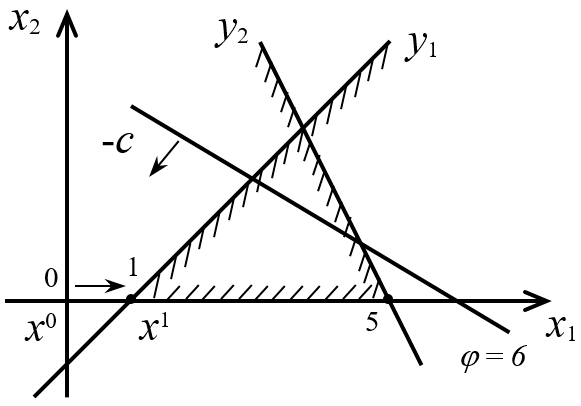
**Примеры.**

**Пример 1**

1. *Целевая функция имеет вид:* 



Приводим к основному виду задачи линейного программирования (ЗЛП) 



⇒

Рис.1

На рис.1 представлено графическое решение задачи с определением допустимого множества и линии уровня целевой функции. Как видно из рис 1, решение задачи достигается в точке (1,0). Решим задачу с помощью алгоритма симплекс-метода.

Cоставляем таблицу: Начинаем решение с точки х1= 0, х2=0.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | *x*1 | *x*2 | –*b* | Точка (0,0) – не крайняя ⇒ ищем разрешающий элемент.  Для этой точки существует элемент а11 ⇒ фиксируем первый столбец и рассмотрим отрицательные величины  и выберем среди них максимальное  и рассмотрим  и .  Разрешающий элемент - а11. |
| *y*1 | 1 | –1 | –1 |
| *y*2 | –2 | –1 | 10 |
| *ϕ* | 1 | 2 | 0 |
|  |  |  |  |

⇒ Делаем первый шаг преобразований:

1. Для 1-ой строки имеем:







1. Для 2-ой строки имеем:







1. Для коэффициентов ϕ при целевой функции имеем:







⇒ получаем таблицу вида:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | *y*1 | *x*2 | –*b* | Пришли в точку х1=1, х2=0.  ⇒ поскольку ∀*i* *bi* > 0 точка (1,0) – крайняя,  и поскольку ∀*r*  точка (1,0) – оптимальная, при этом  *ϕ*min =1. |
| *x*1 | 1 | 1 | 1 |
| *y*2 | –2 | –3 | 8 |
| *ϕ* | 1 | 3 | 1 |

1. **Пример 2**

*Целевая функция имеет вид:* 



Приводим к основному виду задачи линейного программирования:



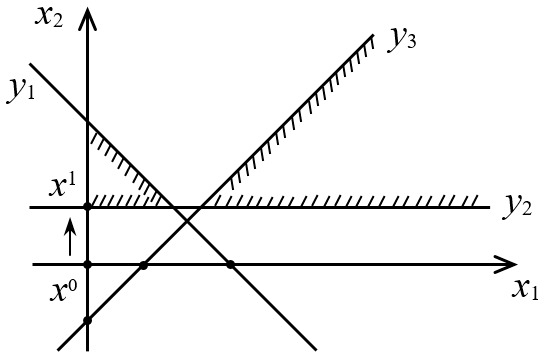


Рис.2

На рис.2 представлено графическое решение задачи с определением допустимого множества и линии уровня целевой функции. Как видно из рис 2, допустимое множество - пусто. Решим задачу с помощью алгоритма симплекс-метода.

Составим таблицу:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | *x*1 | *x*2 | –*b* | Начинаем решение с точки х1= 0, х2=0.  Ищем крайнюю точку. Начинаем решение с точки х1= 0, х2=0.  Можно продолжать её искать, поскольку для  существует  и для  существует .  Выбираем ∀ из *b*2 или *b*3. |
| *y*1 | –1 | –1 | 2 |
| *y*2 | 0 | 1 | –1 |
| *y*3 | 1 | –1 | –1 |
| *ϕ* | 1 | 0 | 0 |

Допустим, выбрали *b*2, тогда в столбце 2 рассмотрим соотношения:

,  – max среди отрицательных ⇒ разрешающий элемент *a*22.

⇒ Делаем первый шаг преобразований:

1. Для 2-ой строки:







1. Для 1-ой строки:







1. Для 3-ей строки:







1. Для коэффициентов *ϕ* имеем:







⇒ приходим к таблице:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | *x*1 | *y*2 | –*b* | Пришли в точку (х1=0,х2=1)  Продолжаем искать крайнюю точку.  Её можно продолжать искать, т.к. существует для .  фиксируем 1-й столбец и рассмотрим в нем отношения:  , ­– min ⇒ *a*11 – разрешающий элемент |
| *y*1 | –1 | –1 | 1 |
| *х*2 | 0 | 1 | 1 |
| *y*3 | 1 | –1 | –2 |
| *ϕ* | 1 | 0 | 0 |

Делаем второй шаг преобразований:

1. Для 1-ой строки имеем:







1. Для 2-ой строки имеем:







1. Для 3-ей строки имеем:







1. Для коэффициентов *ϕ* имеем:







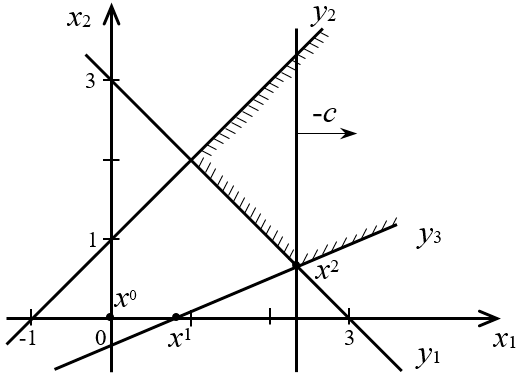
⇒ приходим к таблице:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | *y*1 | *y*2 | –*b* | Пришли в точку х1=1, х2=1.  В этой строке все элементы  при *bi*<0  ⇒ допустимое множество пусто. |
| *x*1 | –1 | –1 | 1 |
| *x*2 | 0 | 1 | 1 |
| *y*3 | –1 | –2 | –1 |
| *ϕ* | –1 | –1 | 1 |

1. **Пример 3**

*Целевая функция имеет вид:* 





Приводим к основному виду задачи линейного программирования:

 Рис.3

На рис.3 представлено графическое решение задачи с определением допустимого множества и линии уровня целевой функции. Как видно из рис 3, целевая функция неограничена на допустимом множестве. Решим задачу с помощью алгоритма симплекс-метода.

Составим таблицу:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | *x*1 | *x*2 | –*b* | Начинаем решение с точки х1= 0, х2=0.  Точка (0,0) – не крайняя и можно продолжать искать крайнюю точку, т.к. для –*b*1 = –3 существует *a*11 и *a*12 > 0 ⇒ выбираем ∀ из них, например 1-й столбец.  Рассмотрим отношения  ⇒ *a*31– разрешающий элемент |
| *y*1 | 1 | 1 | –3 |
| *y*2 | 1 | –1 | 1 |
| *y*3 | –1 | 2 | 1 |
| *ϕ* | –1 | 0 | 0 |

⇒ Делаем первый шаг преобразований:

1. Для 3-ей строки имеем:







1. Для 1-ой строки имеем:







1. Для 2-ой строки имеем:







1. Для коэффициентов *ϕ* имеем:







⇒ приходим к таблице:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | *y*3 | *x*2 | –*b* | Пришли в точку х1=1, х2=0  Точка (1,0) – не крайняя.  Крайнюю точку можно искать, т.к. для –*b*1 = –2 существует .  ⇒ *a*12 – разрешающий элемент. |
| *y*1 | –1 | 3 | –2 |
| *y*2 | –1 | 1 | 2 |
| *x*1 | –1 | 2 | 1 |
| *ϕ* | 1 | –2 | –1 |

⇒ Делаем второй шаг преобразований:

1. Для 1-ой строки имеем:







1. Для 2-ой строки имеем:







1. Для 3-ей строки имеем:







1. Для коэффициентов *ϕ* имеем:







 приходим к таблице:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | *y*3 | *y*1 | –*b* | Пришли в точку х1=  , х2=  ⇒ Точка  – крайняя, но  в этом столбце при  все *aij* > 0 ⇒ оптимальной точки нет, inf*ϕ* = –∞ (целевая функция не ограничена на допустимом множестве) |
| *x*2 |  |  |  |
| *y*2 |  |  |  |
| *x*1 |  |  |  |
| *ϕ* |  |  |  |