**Введение в математическое программирование**

Задачи математического программирования находят применения в различных областях деятельности, где необходим выбор одного из возможных образов действий, т.е. программ действий. Например, при решении проблем управления и планирования, проектирования в военном деле и т.д..

Содержание математического программирования составляет теорию и методы задач о нахождении экстремумов функций на множествах, определяемых линейными и нелинейными ограничениями (равенствами и неравенствами).

Выделяют два класса задач математического программирования:

* Детерминированные задачи – когда вся исходная информация является полностью определенной;
* Задачи стохастического программирования – в этих задачах исходная информация содержит элементы неопределенности, либо некоторые параметры задачи носят случайный характер с известными вероятностными характеристиками.

***В математическом программировании выделяют следующие основные разделы:***

**Линейное программирование**.

Особенности – целевая функция линейна, множества, на котором ищется экстремум целевой функции задается системой линейных равенств и неравенств.

**Нелинейное программирование**

Особенности - Нелинейная целевая функция и ограничения.

**Целочисленное программирование**

Особенности – на переменные накладывается условие целочисленности.

Специфика задач математического программирования состоит в том, что:

1. К задачам математического программирования, как правило, не применимы методы классического анализа для отыскания условных экстремумов
2. В практических задачах число ограничений и переменных столь велико, что если просто перебирать точки, которые возможно являются экстремумами, то это потребует крайне больших вычислительных затрат.

**Целью математического программирования** является создание, где это возможно, аналитических методов определения оптимального решения. При отсутствии таких методов, создание эффективных вычислительных способов получения приближенного решения.

**Различные формы моделей задач линейного программирования и их эквивалентность**

*Задача использования ресурсов*

Для производства двух видов продукции Р1, Р2 используют три вида ресурсов R1, R2, R3 . Запасы ресурсов и расход ресурсов на изготовление единицы продукции приведены ниже.

|  |  |  |  |
| --- | --- | --- | --- |
| Виды ресурсов | Расход ресурсов на 1 ед продукции | | Запасы ресурсов, ед |
| Р1 | Р2 |
| R1 | 2 | 5 | 20 |
| R2 | 8 | 5 | 40 |
| R3 | 5 | 6 | 30 |

Сколько продукции первого и второго вида будут выпускаться. (Составить мат модель)

X1 – кол-во единиц первой продукции

X2 – кол-во единиц второй продукции

Ограничения для R1: { 2\*X1 + 5\*X2 <= 20

Ограничения для R2: { 8\*X1 + 5\*X2 <=40

Ограничения для R3: { 5\*X1 + 6\*X2 <=30

{ X1 , X2>=0

Z= 50\*X1 + 40\*X2 🡪 max

*Задача составления рациона (задача о диете)*

При откорме каждое животное ежедневно должно получать не менее 9 ед питательного вещества S1, не менее 8 ед питательного вещества S2, и не менее 12 ед питательного вещества S3. Содержание кол-ва единиц питательных веществ в 1 кг каждого вида корма и стоимость 1 кг корма ниже.

|  |  |  |
| --- | --- | --- |
| Питательное вещество | Кол-во единиц питательного вещества в 1 кг корма | |
| Корм 1 | Корм 2 |
| S1 | 3 | 1,2 |
| S2 | 0,5 | 2 |
| S3 | 1 | 6,1 |
| Стоимость 1 кг корма, ден. Ед. | 4 | 6 |

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

X1 – кол-во кг 1-го корма

X2 – кол-во кг 2-го корма

{ 3\*X1 + 1.2\*X2 >=9

{ 0.5\*X1 + 2\*X2 >=8

{ 1\*X1 + 6.1\*X2 >=12

{ X1, X2 >=0

Z= 4\*X1 + 6\*X2 🡪min

*Транспортная задача*

Необходимо перегнать в пунктах A и B соответственно 6 и 4 машины в пункты назначения C (3 машины) и D (7 машин). Ниже приведены расстояния между каждым из пунктов отправления и назначения.

|  |  |  |
| --- | --- | --- |
| Пункты | C | D |
| A | 80 | 30 |
| B | 60 | 90 |

Требуется спланировать перевозки так, чтобы суммарный пробег в машино-километрах оказался **наименьшим**.

X1 – сколько машин перегнали из A в C

X2 – сколько машин перегнали из A в D

Y1 – сколько машин перегнали из B в C

Y2 – сколько машин перегнали из B в D

{ X1 + X2 = 6

{ Y1 + Y2

{ X1 + Y1 = 3

{ X2 + Y2 = 7

{ X1, X2 ,Y1, Y2 >=0

Z= 80\*X1 + 60\*Y1 + 30\*X2 + 90\*Y2 🡪min

Требуется найти X1 … Xn удовлетворяющих ограничения

**(1.1)**

{ A11 \* X1 + a12 \* X2 +… + A1n \* Xn =B1

{ Am1 \* X1 + Am2 \* X2 + … + Amn \* Xn = Bm

{ Am+k 1 \* X1 + Am+k 2 \* X2 + … + Am+k n \* Xn = Bm+k

{ X1, … , Xn >=0 **(1.2)**

Z= C1 \* X1 + C2 \* X2 +…+Cn \* Xn =max (min) **(1.3)**

Без ограничения общности можно считать, что в системе (1.1) первые m условий являются уравнениями, а последующие k – неравенствами. Очевидно этого всегда можно добиться, за счет переупорядочивания условий. Неравенства данной системы можно полагать одного знака, т.к. неравенства противоположного смысла можно всегда привести к данным, путем умножения обеих частей на -1.

Функцию (1.3) называют *целевой функцией*.

Совокупность значений X1, …, Xn, удовлетворяющих ограничениям (1.1) и (1.2) называют **допустимым решением**. Совокупность всевозможных допустимых решений называют **областью допустимых решений задач**.

Допустимые решения, при которых целевая функция принимает свое максимальное (минимальное) значение, называют **оптимальным решением**.

Рассмотрим частные случаи линейного программирования.

**Постановка задачи линейного программирования в симметричной форме.**

Требуется найти значения неизвестных x1, …, xn.

Удовлетворяющих ограничениям

{ a11 \*x1 + a1 2 \*x2 + … + a1 n \*xn<=b1

{ a21 \*x1 + a2 2 \*x2 + … + a2 n \*xn<=b2

…………………

{ an1 \*x1 + an 2 \*x2 + … + an n \*xn<=bn

Xj >= 0 (j=1..n)

Z= C1 \* X1 + C2 \* X2 +…+Cn \* Xn =min

**Постановка задачи в канонической форме.**

X1, …, Xn – неизвестны.

Ограничения:

{ a1 1 \*x1 + a1 2 \*x2 + … + a1 n \*xn=b1

{ a2 1 \*x1 + a2 2 \*x2 + … + a2 n \*xn=b2

…………

{ am 1 \*x1 + am 2 \*x2 + … + am n \*xn=bm

Xi>=0 (i=1..n)

Z= C1 \* X1 + C2 \* X2 +…+Cn \* Xn =max

Указанные три формы задач линейного программирования эквивалентны в том смысле, что каждая из них с помощью некоторых преобразований может быть переписана в форме другой задачи. Это означает, что если имеется способ нахождения решения одной из указанных задач, то тем самым можно определить оптимальное решение любой их трех задач.

Правила перехода от одной формы к другой состоят в следующем:

1. Для перехода от минимума целевой функции к максимуму, следует изменить знак этой функции (F= -Z)
2. Если в ограничении правая часть неотрицательна, то следует умножить это ограничение на -1
3. Ограничения неравенства преобразуются в ограничения равенства путем введения фиктивных неотрицательных переменных
4. Если некоторая неизвестная не подчинена условию неотрицательности, то она заменяется в целевой функции и во всех ограничениях разностью между двумя новыми неотрицательными неизвестными

Пример.

Привести задачу линейного программирования к канонической форме.

Z= 6X1 + 4X2 –X3 🡪min

{ 4X1 – X3 <=2

{ -X1 – X2 +X3 <=1

{ -2X1 +2X2 >=3

X1, X3>= 0

Приводим:

{ 4X1 –X3 + X4 = 2

{ -X1 – X2 +X3 + X5 =1

{ -2X1 +2X2 – X6 =3

F= -Z = -6X1 - 4X2 +X3 🡪max

X2 = X7 – X8

X7, X8 >=0

{ 4X1 –X3 + X4 = 2

{ -X1 – X7 + X8 +X3 + X5 =1

{ -2X1 +2X7 – 2X8 – X6 =3

X1, X3, X4, X5, X6, X7, X8 >=0

F= -Z = -6X1 - 4X7 +4X8 +X3 🡪max

**Базисное решение задач линейного программирования**

X1, X2,…, Xn – неизвестны

(\*)

{ A11 \* X1 + A12 X2 + …+ A1n \* Xn = b1

{ A21 \* X2 + A22 X2 + …+ A2n \* Xn = b2

{ …

{ Am1 \* X1 + Am2 X2 + …+ Amn \* Xn = bm

{Xj >= 0 , j=1..n

Z=C1 X1 + C2 X2 + … + Cn Xn 🡪 max

Усл (\*) представляют собой систему линейных уравнений.

1. Если система (\*) несовместна, то задача линейного программирования не имеет решение.
2. Если существует единственное решение системы (\*), то оно очевидно и является одновременно решением задачи линейного программирования поскольку другого решения нет.
3. Если система (\*) имеет бесконечное множество решений, то задача линейного программирования становится объектом исследования.

Пусть (\*) имеет бесконечное множество решений. Тогда используя метод Жардана-Гаусса, ее можно привести к ступенчатому виду.

(\*\*)

{ X1 + … + A1,r+1 \* Xr+1 + A1n \* Xn =B1

{ X2 + … + A2,r+1 \* Xr+1 + A2n \* Xn =B2

{ …

{ Xr + A1,r+1 \* Xr+1 + A1n \* Xn =Br

{ R<n

Система (\*) и (\*\*) эквивалентны, то есть имеют одно и тоже множество решений. Неизвестные, входящие с единичными коэффициентами только в одно уравнение системы и с нулевыми в остальные, называют **базисными** или **базисом**.

Б= {X1, X2,…, Xr } – базис

Xr+1, … Xn

Все остальные переменные – **свободные**.

**Общим решением** называют совокупность выражений базисных неизвестных через свободные неизвестные и свободные члены.

**Общее решение (\*\*):**

{ X1 = B1 – (A1,r+1 Xr+1 + … A1n Xn)

{ X2 = B2 – (A2,r+1 Xr+1 + … A2n Xn)

…

{ Xr = Br – (Ar,r+1 Xr+1 + … Arn Xn)

Частным решением называют решение, получающееся из общего при конкретных значениях свободных неизвестных.

**Базисным решением** называют частное решение, получающееся из общего при нулевых значениях свободных неизвестных.

(\*\*\*)

{ X1 = B1

{ X2 = B2

{ …

{ Xr = Br

{ Xr+1 = 0

{ Xr+2 = 0

{ …

{ Xn = 0

Если B1, B2, …, Br >=0 (\*\*\*) – допустимо.

Базисное решение называют **не вырожденным**, если все значения базисных неизвестных положительны и **вырожденным** в противном случае.

**Теоретические основы методов решения задачи линейного программирования**

Множество точек называется выпуклым, если оно вместе с любыми двумя своими точками содержит весь отрезок, соединяющий эти точки.

Среди точек выпуклого множества выделяют **внутренние, граничные и угловые точки**.

Точка выпуклого множества называется **внутренней**, если в некоторой ее окрестности содержатся точки только данного множества.

Точка выпуклого множества называется **граничной**, если в любой ее окрестности содержатся как точки принадлежащей данному множеству, так и точки, не принадлежащие ему.

Точка выпуклого множества называется **угловой**, если она не является внутренней и граничной ни для какого отрезка, целиком принадлежащего данному множеству.

Множество точек называется **замкнутым**, если оно включает все свои граничные точки.

Множество точек называется **ограниченным**, если существует шар (круг) радиуса конечной длины с центром в любой точке множества, который полностью содержит в себе данное множество.

Выпуклое, замкнутое множество точек пространства (плоскости) имеющее конечное число угловых точек, называется **выпуклым многогранником** (**выпуклым многоугольником**), если оно *ограниченно* и **выпуклой многогранной (многоугольной) областью**, если оно *не ограничено*.

**Свойства задач линейного программирования.**

1. Множество всех допустимых решений задач линейного программирования представляют собой выпуклый многогранник или выпуклую многогранную область (выпуклый многоугольник или выпуклую многоугольную область), которую называют **многогранником решений (многоугольником решений)**.
2. Число угловых точек многогранника (многоугольника) решений конечно.
3. Экстремум целевой функции в задаче линейного программирования, если он существует, достигается хотя бы в одной угловой точке многогранника (многоугольника) решений. Следовательно, для нахождения оптимального решения, необходимо исследовать лишь конечное число вершин многогранника (многоугольника) решений.
4. Каждому допустимому базисному решению задач линейного программирования соответствует угловая точка многогранника (многоугольника) решений. И наоборот, каждой угловой точке многогранника (многоугольника) решений соответствует допустимое базисное решение. Таким образом экстремум целевой функции следует искать среди конечного числа допустимых базисных решений, совпадающих с угловыми точками многогранника решений.

**Теоретические основы симплекс-метода.**

Симплекс-метод представляет собой вычислительную процедуру, основанную на последовательном переборе допустимых базисных решений. При этом значение целевой функции улучшается или, по крайней мере, не ухудшается.

Основная идея симплекс метода заключается в следующем:

1. Находится одно из допустимых базисных решений, которое называют **начальным.**
2. Осуществляется его пошаговое улучшение, пока не будет получено оптимальное решение, либо будет установлено, что целевая функция не ограничена на множестве допустимых решений.

Симплекс-методом решаются задачи линейного программирования, приведенные к канонической форме.

X1, X2, … ,Xn , которые уд.

{ A11 X1 + A12 X2 +…+ A1n Xn <=B1

{ A21 X1 + A22 X2 +…+ A2n Xn <=B2

…

{ Am1 X1 + Am2 X2 +…+ Amn Xn =Bm

{ Xj >=0,j=1..n

Z=C1 X1 + C2 X2 + … + Cn Xn 🡪 max

X1, X2, … ,Xn , Xn+1, Xn+2, …, Xn+m –уд.

{ A11 X1 + A12 X2 +…+ A1n Xn + Xn+1 = B1

{ A21 X1 + A22 X2 +…+ A2n Xn + Xn+2 = B2

…

{ Am1 X1 + Am2 X2 +…+ Amn Xn + Xn+m = Bm

{ Xj >=0,j=1..n+m

Z=C1 X1 + C2 X2 + … + Cn Xn + 0 Xn+1 + … + 0 Xn+m🡪 max

Перепишем целевую функцию Z в виде уравнения

-C1 X1 - C2 X2 - … - Cn Xn + Z = 0 – приведенная запись

Запись целевой функции в виде уравнения называется **приведенной**. Такая запись позволяет записать задачи линейного программирования в виде системы линейных уравнений.

(\*)

Ci = am+I,i

{ A11 X1 + A12 X2 +…+ A1n Xn + Xn+1 = B1

{ A21 X1 + A22 X2 +…+ A2n Xn + Xn+2 = B2

…

{ Am1 X1 + Am2 X2 +…+ Amn Xn + Xn+m = Bm

{ am+1,1 X1 + am+1,2 X2+ … + am+1,n Xn + Z= 0

{ Xj >=0,j=1..n+m

Система (\*) приведена к **ступенчатому виду**. Если в системе (\*) все свободные члены неотрицательны ( т.е. Bi >=0), тогда имеется начальное допустимое базисное решение.

Начальное допустимое базисное решение:

X1 =0, X2 =0, …, Xn =0, Xn+1 =B1, Xn+2 = B2, …, Xn+m = Bm ,

Z =0 – начальное значение целевой функции.

Коэффициенты перед свободными неизвестными в m+1 строке называют **оценками.** Можно показать, что если эти оценки больше 0 (am+1,j >0), то при введении переменной Xj в базис целевая функция увеличится.

Если am+1,j <= при введении Xj в базис, Z уменьшится.

Таким образом оценки определяют приращение целевой функции при введении в базис соответствующей свободной неизвестной. Отсюда следует критерий оптимальности симплекс-метода: если в приведенной записи целевой функции все оценки окажутся неотрицательными, то достигнуто оптимальное решение и расчет прекращается. В случае неоптимальности текущего базиса осуществляется переход к следующему. Это делается за счет вывода одной переменной из базиса и ввода другой. При этом размерность базиса сохраняется.

**Основные теоремы двойственности**

**Первая теорема двойственности:**

1. Если одна из двойственных задач имеет оптимальное решение, то другая так же имеет оптимально решение, причем для экстремальных значений целевых функций выполняется Zmax=Tmin.
2. Если одна из двойственных задач неразрешима, то есть Zmax 🡪infin или Tmin 🡪infin, то и другая задача неразрешима.

Согласно данной теореме план производства X\*=(x1\*,…) и план стоимости U\*=(u1\*,…) являются оптимальными тогда и только тогда, когда цена произведенной продукции равна суммарной стоимости ресурсов.

Таким образом двойственные оценки позволяют сопоставить и сбалансировать затраты и результаты решения.

**Вторая теорема двойственности (теорема о дополняющей нежёсткости)**

Пусть X\* =(x1\*,…,xn\*) – оптимальное решение исходной задачи, U\*=(u1\*,…,un\*) – оптимальное решение двойственной задачи, тогда:

1. Если
2. Если U\*I >0, тогда

Таким образом если в некотором оптимальном решении исходной задачи расход i-го ресурса строго меньше его запаса bi, то в оптимальном решении двойственной задачи соответствующая двойственная оценка этого ресурса рвана 0. Если же в некотором оптимальном решении двойственной задачи его i-я компонента >0, то в оптимальном решении расход соответствующего ресурса bi равен его запросу.

**Транспортные задачи. Постановка транспортной задачи**

Транспортными задачами называют задачи определения оптимального плана перевоза груза из данных пунктов отправления в заданные пункты назначения. **В общем случае транспортная задача может быть сформулирована следующим образом:**

Однородный груз сосредоточен у M поставщиков в объемах a1, …, am­. Данный груз необходимо доставить N потребителям. В объемах b1,…,bn. Известны Сij – стоимости перевозок единицы груза от каждого i-го поставщика каждому j-му потребителю. Требуется составить план перевозок, позволяющий:

1. Вывести все грузы от поставщиков
2. Полностью удовлетворить запросы потребителей
3. Минимизировать суммарную стоимость перевозок

Xij – кол-во груза, планируемого к перевозке от i-го поставщика каждому j-му потребителю.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Ai\Bj | B1 | B2 | … | Bn |
| A1 | X11 C11 | X12 C12 | … | X1n C1n |
| A2 | X21 C21 | X22 C22 | … | X2n C2n |
| … |  |  |  |  |
| Am | Xm1 Cm1 | Xm2 Cm2 | … | Xmn Cmn |

{

2\*

{

4\*

Особенности транспортной задачи:

1. Распределению подлежат однородные ресурсы
2. Условия задачи описываются только уравнениями
3. Все неизвестные выражаются в одинаковых единицах измерения
4. Во всех уравнениях коэффициентах при неизвестных равны 1
5. Каждая неизвестная встречается только в 2-х уравнениях системы ограничений

**Необходимые и достаточные условия разрешимости транспортных задач**

Всякое решение ограничений 1\* - 3\* определяемое матрицей X=(xij)m&n называется **допустимым решением транспортной задачи**.

Допустимое решение, при котором целевая функция принимает минимальное значение, называется **оптимальным решением транспортной задачи**.

Если транспортная задачи имеет оптимальное решение, то оно совпадает по крайней мере с одним из ее допустимых базисных решений.

Теорема:

Для того, чтобы транспортная задача имела решение, необходимо и достаточно, чтобы суммарные запасы груза была равна суммарным потребностям. . **Это уравнение баланса**.

Модель транспортной задачи, в которой соблюдается уравнение баланса, называется закрытой. Если уравнение баланса не соблюдается, то модель называется открытой.

**Правила приведения открытой модели ТЗ к закрытой**

1. Если запасы превышают потребность.

Тогда вводится фиктивный n+1 пункт потребления с потребностью а Ci,n+1 =0

1. Если потребности превышают запас.

Тогда вводится фиктивный m+1 пункт производства с запасом груза , а Cm+1,j=0.

Система ограничений транспортной задачи содержит m\*n неизвестных, и m+n уравнений, связанных между собой уравнением баланса. Тогда если сложить почленно уравнения отдельно подсистема 1\* и уравнения отдельно подсистемы 2\*, то получим 2 одинаковых уравнения.

Наличие в системе уравнений 2-х одинаковых уравнений говорит об ее линейной зависимости. Число линейно независимых уравнений в системе ограничений будет равняться m+n-1. Следовательно число базисных неизвестных в транспортной задаче так же равно m+n-1. Таким образом если получено невырожденное допустимое базисное решение транспортной задачи, то в матрице X положительными являются только m+n-1компонентов, а остальные 0.

**Нахождение начального допустимого базисного решения методом «Северо-западного угла»**

Суть метода:

Последовательно распределяются все запасы, имеющиеся в 1, 2,… пунктах поставок по 1, 2, … пунктам потребления. Построение начального допустимого базисного решения начинается с левого верхнего угла транспортной таблицы. А именно с клетки с координатами (1,1). Каждый шаг распределения сводится к попытке полного исчерпания запасов в очередном пункте поставок или к попытке полного удовлетворения потребности в очередном пункте потребления. Нахождение начального допустимого базисного решения продолжается до тех пор, пока не будут исключены все поставщики и потребители. На каждом шаге, кроме последнего из рассмотрения удаляется только одна строка или только один столбец, поэтому в результате получается m+n-1 базисная неизвестная. Если на некотором шаге ai = bj , то начальное допустимое базисное решение будет вырожденным, т.е. в базисе будет переменная =0.