Лекция

Введение

Задачи линейного программирования

Домашова Д.В.

Содержание

Задача принятия решения (ЗПР)

Системное описание ЗПР

Управляющая подсистема может воздействовать на объект управления с помощью альтернативных управляющих воздействий: УВ1, УВ2, УВn

Состояние Объекта управления определяется двумя факторами: выбранным УВ; состоянием среды

Принципиальным является следующее обстоятельство: Управляющая подсистема не может воздействовать на среду, более того, она, как правило, не имеет полной информации о состоянии среды

Цель Управляющей подсистемы: перевести объект управления в наиболее предпочтительное для себя состояние, используя для этого любое УВ, имеющееся в ее распоряжении

Выбор Управляющей подсистемой конкретного УВ (допустимой альтернативы, допустимого решения) называется принятием решения

При принятии решения основной задачей является нахождение оптимального решения.
На содержательном уровне оптимальное решение можно определить как наилучшее в следующем смысле: оптимальное решение в наибольшей степени соответствует цели Управляющей подсистемы в рамках имеющейся у нее информации о состоянии среды

Задача принятия решения (ЗПР)

В зависимости от информации, которую имеет при принятии решения управляющая подсистема относительно состояния среды, различают несколько основных типов задач принятия решения.

1 *Принятие решения в условиях определенности* характеризуется тем, что состояние среды является фиксированным (неизменным), причем управляющая система «знает», в каком состоянии находится среда.

2. *Принятие решения в условиях риска* означает, что управляющая подсистема имеет информацию стохастического характера о поведении среды (например, ей известно распределение вероятностей на множестве состояний среды)

3 *Принятие решения происходит в условиях неопределенности*, если никакой дополнительной информации (кроме знания самого множества возможных состояний среды) управляющая подсистема не имеет.

4. *Принятие решения в теоретико-игровых условиях* имеет место тогда, когда среду можно трактовать как одну или несколько целенаправленных управляющих подсистем. В этом случае математическая модель принятия решения называется ***теоретико-игровой моделью*** (*игрой*).

Постановка задачи оптимизации

$$f(x) \to \min$$

$$D = \begin{cases} g_i(x) \le 0,\ i = \overline{1,m} \\ x \in X \subseteq R^n \end{cases} \qquad (1)$$

где $f(x),\ g_i(x),\ i = \overline{1,m}$ - нелинейные функции.

Решение: $x^* = \arg\min\limits_{x \in D} f(x)$

Разрешимость задачи оптимизации

$x^* \in R^n$

Опр.: Точка $x^* \in D$ называется точкой глобального минимума функции f на множестве D, или глобальным решением задачи (1), если

$$f(x^*) \leq f(x) \qquad (2)$$

для любого x принадлежащего множеству D

Опр.: Точка x^* принадлежащая множеству D называется точкой локального минимума f на D, или локальным решением задачи (1), если

$$\exists \varepsilon > 0 : f(x^*) \leq f(x) \qquad (3)$$

для любого $x \in D \cap U_\varepsilon(x^*)$,

где $U_\varepsilon(x^*) = \{x \in R^n : \|x - x^*\| < \varepsilon\}$ – шар радиуса $\varepsilon > 0$ с центром в x^*.

Замечание: Если неравенство (2) или (3) выполняется как строгое при $x \neq x^*$, то говорят, что x^* - точка строгого минимума в глобальном или локальном смысле.

Обозначение : $x^* \in X$ – точка глобального минимума:

$f(x^*) = \min f(x) \qquad x \in D$

или $x^* = \arg\min f(x) \qquad x \in D$

Разрешимость задачи оптимизации

$f(x) \to \max$ (4)
x ЄD

Задача (4) эквивалентна задаче

$- f(x) \to \min$
x Є D

Отсюда следует, что можно переносить результаты, полученные для задачи min на задачи max и наоборот.

Разрешимость задачи оптимизации

В пространстве R^n множество D – компактно <=> когда оно замкнуто и ограничено (содержится в шаре радиуса $M < +\infty$)

Следующая теорема дает ответ на вопрос о существовании оптимального решения для задачи оптимизации.

Теорема Вейерштрасса. Если f – непрерывная действительная функция на компактном множестве $D \subset R^n$ (D – замкнуто и ограничено) => задача (1) имеет оптимальное решение $x^* \in D$.

Классификация задач оптимизации

1. Задачи безусловной оптимизации.

$f(x) \to \min,\ x \in R^n$ (5)

Условия оптимальности (экстремума)

Необходимое:

Теорема: Пусть функция f дифференцируема в т. $x^* \in R^n$. Если x^* - локальное решение задачи (5), то $f'(x^*) = \left(\frac{\partial f}{\partial x_1}(x^*), \dots, \frac{\partial f}{\partial x_n}(x^*) \right) = 0$

Точка x^* - стационарная точка.

Достаточное:

Теорема: Пусть функция f дважды дифференцируема в т. $x^* \in R^n$. Пусть $f'(x^*) = 0$, а матрица $f''(x^*)$ положительно определена { $(f''(x^*)h, h) > 0$ для любого $h \in R^n$, $h \neq 0$}. Таким образом, x^*-строгое локальное решение задачи (5).

Классификация задач оптимизации

2. Задачи условной оптимизации.

2.1Задачи линейного программирования.

Целевая функция – линейна. D задается системой линейных равенств и неравенств.

2.2 Задачи нелинейного программирования.

Целевая функция – нелинейная.

В классе задач нелинейного программирования можно выделить:

- задачи выпуклого программирования.

Выпуклая целевая функция и выпуклое D.

- квадратичного программирования.

Целевая функция имеет квадратичную форму, ограничения – линейные неравенства и равенства.

Задача об оптимальном плане производства продукции

n – видов продукции, $j=\overline{1,n}$;

m – видов ресурсов (сырья), $i=\overline{1,m}$;

a_{ij} – количество ресурса i-го вида, требующегося для производства единицы продукции j-го вида;

b_i – запасы ресурса i-го вида $i=\overline{1,m}$;

c_j – доход (прибыль) от реализации единицы продукции i-го вида.

Необходимо найти такой план производства продукции, при котором достигается максимальная прибыль, для реализации которого достаточно имеющихся ресурсов.

$$f=c_1x_1+\ldots+c_nx_n\to\max$$

$$\begin{cases} a_{11}x_1+\ldots+a_{1n}x_n\le b_1 \\ \ldots \\ a_{m1}x_1+\ldots+a_{mn}x_n\le b_m \end{cases}$$

$$x_j\ge 0,\ j=\overline{1,n}$$

Задача о диете
(исторически одна из самых первых)

n – видов кормов, $j = \overline{1, n}$;

m – видов питательных веществ, $i = \overline{1, m}$;

a_{ij} – содержание *i*-го вида питательного вещества в единице j-го вида корма;

b_i – необходимый минимум *i*-го питательного вещества в день $i = \overline{1, m}$;

c_j – стоимость единицы *i*-го вида корма.

Необходимо составить рацион, имеющий минимальную стоимость, в котором содержание каждого вида питательных веществ было бы не менее установленного предела.

$$f = c_1 x_1 + \ldots + c_n x_n \to \min$$

$$\begin{cases} a_{11} x_1 + \ldots + a_{1n} x_n \geq b_1 \\ \ldots \\ a_{m1} x_1 + \ldots + a_{mn} x_n \geq b_m \end{cases}$$

$$x_j \geq 0, \; j = \overline{1, n}$$

Линейное программирование

Определение 1. Задача, состоящая в нахождении наибольшего (наименьшего) значения (1) на множестве точек $x=(x_1,\dots,x_n)$, удовлетворяющих системе ограничений вида (2) называется задачей линейного программирования общего вида.

$$f = c_1x_1 + c_2x_2 + \dots + c_nx_n \xrightarrow{?} max(min) \qquad (1)$$

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n \; R_1 \; b_1 \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n \; R_2 \; b_2 \\ \dots \quad \dots \quad \dots \quad \dots \quad \dots \quad \dots \\ a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n \; R_n \; b_m \end{cases} \qquad (2)$$

Здесь $f = c_1x_1 + c_2x_2 + \dots + c_nx_n \xrightarrow{?} \max$ – целевая функция;

R_i , $i=\overline{1,m}$ · один из знаков =, $\geq$, $\leq$;

C_j $j=\overline{1,n}$ и a_{ij} , $i=\overline{1,m}$, $j=\overline{1,n}$ - заданные константы.

Линейное программирование

Определение 2. Всякую точку $\overline{X} = (x_1, ..., x_n)$, компонента которой удовлетворяет всем ограничениям системы (2), будем называть допустимой точкой или допустимым решением задачи, или допустимым планом задачи.

Задача линейного программирования состоит, таким образом, в нахождении такой допустимой точки $\overline{x}^0$ (такого допустимого плана) среди множества допустимых точек, при которой целевая функция принимает ***max(min)*** значение.

Допустимое решение $\overline{x}^{(0)} = (x_1^{(0)}, ..., x_1^{(0)})$, доставляющее целевой функции оптимальное значение (оптимум), будем называть оптимальным решением или оптимальным планом задачи.

Геометрическая интерпретация решения двумерной задачи линейного программирования

Рассмотрим двумерную задачу:

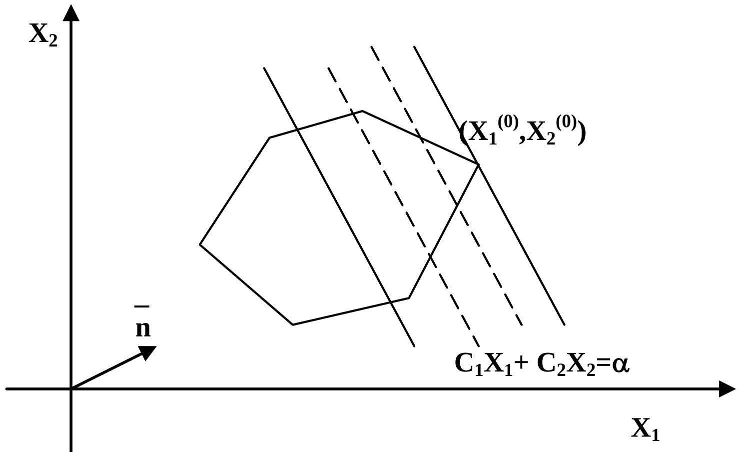
$$F = c_1 x_1 + c_2 x_2 \to max$$

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

Каждое из ограничений $a_{i1} x_1 + a_{i2} x_2 \le b_i$ определяет в плоскости, с системой координат $X_1 0 X_2$, множество точек лежащих по одну сторону от прямой $a_{i1} x_1 + a_{i2} x_2 \le b_i$ (т.е. полуплоскость).

Множество всех точек плоскости, координаты которых удовлетворяют всем ограничениям, т.е. принадлежат сразу всем полуплоскостям, определяемым отдельными ограничениями, будет представлять собой допустимое множество.

Геометрическая интерпретация



Пусть допустимая область задачи оказалась непустой.
Мы хотим найти те точки допустимой области, координаты которых дают целевой функции наибольшее значение.

Построим линию уровня целевой функции $F = c_1 x_1 + c_2 x_2 = \alpha$.

Перемещая линию уровня в направлении вектора $\mathbf{grad}\ \mathbf{G} = (c_1, c_2) = \bar{n}$, нормального к линии уровня, будем получать в пересечении этой линии с допустимой областью точки, в которых целевая функция принимает новое значение, большее, чем на предшествующих линиях уровня.

Пересечение допустимой области с линией уровня в том ее положении, когда дальнейшее перемещение дает пустое множество, и будет множеством оптимальных точек задачи линейного программирования.

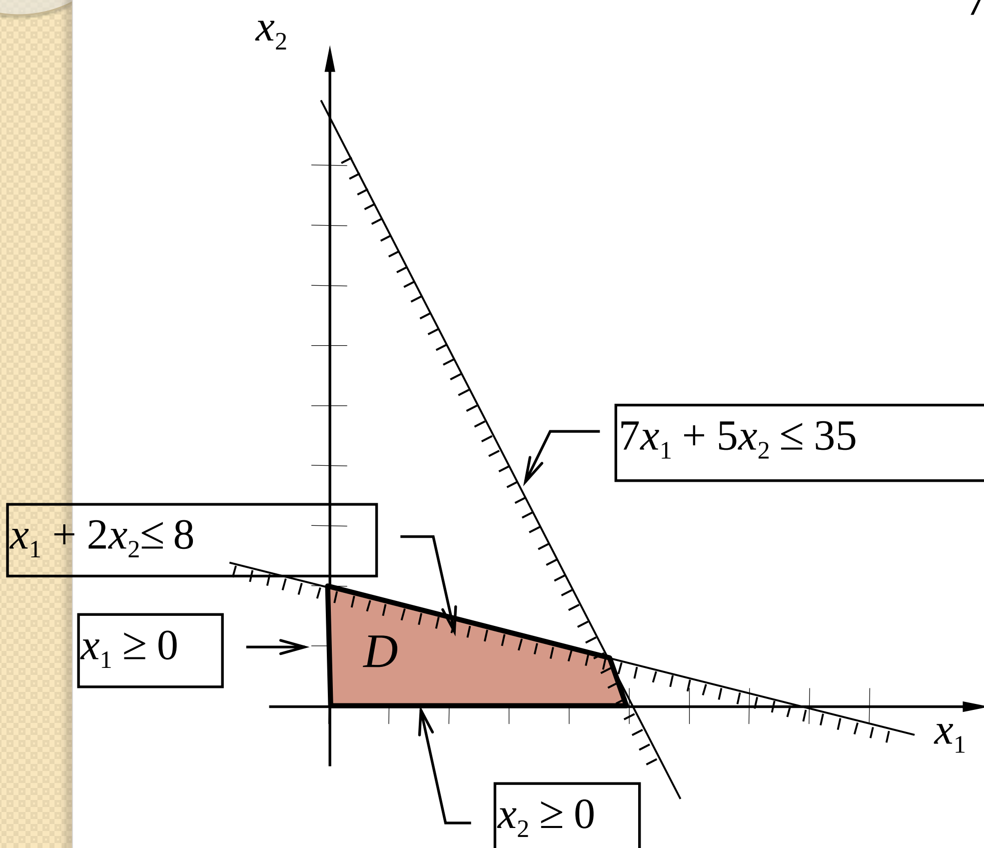
Геометрическая интерпретация ЗЛП

$$f(x_{1,}\ x_2) = 4x_1 + 5x_2 \rightarrow max$$

$$7x_1 + 5x_2 \leq 35$$

$$x_1 + 2x_2 \leq 8$$

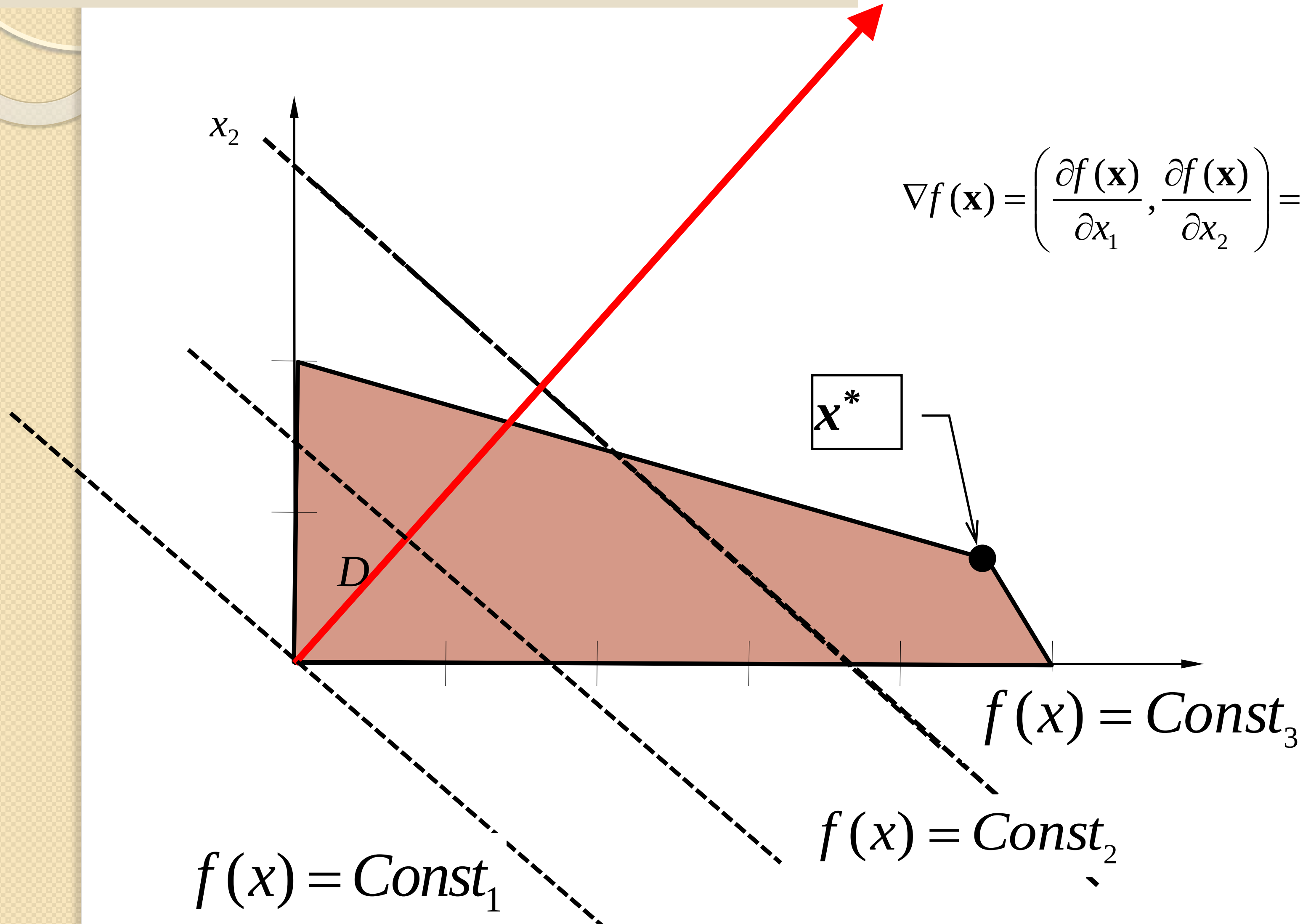
$$x_1 \geq 0 \qquad x_2 \geq 0$$



Геометрическая интерпретация ЗЛП

$$\nabla f(\mathbf{x}) = \left(\frac{\partial f(\mathbf{x})}{\partial x_1}, \frac{\partial f(\mathbf{x})}{\partial x_2}, \ldots, \frac{\partial f(\mathbf{x})}{\partial x_n} \right) = (c_1, c_2, \ldots, c_n) = \mathbf{c}$$

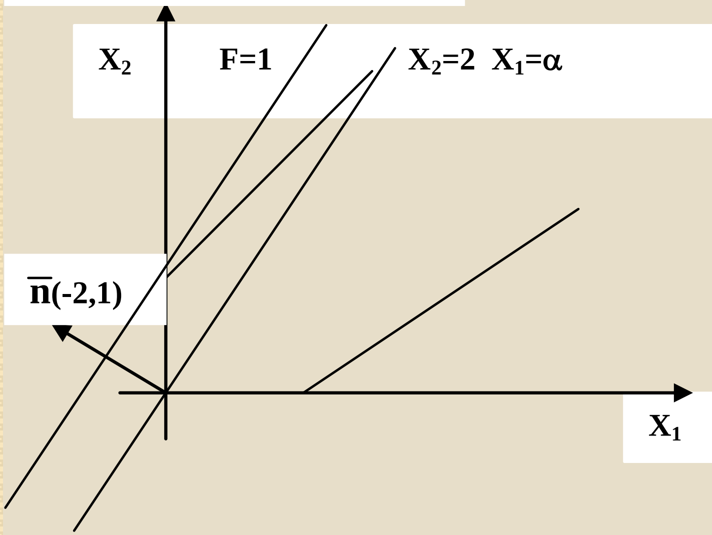
$$\nabla f(\mathbf{x}) = \left(\frac{\partial f(\mathbf{x})}{\partial x_1}, \frac{\partial f(\mathbf{x})}{\partial x_2} \right) = (4,5)$$



Геометрическая интерпретация ЗЛП

$F = -2x_1 + x_2$

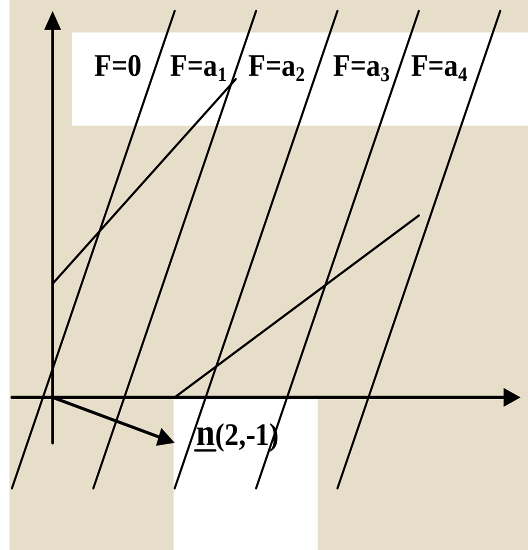
$$\begin{cases} -x_1 + x_2 \le 1 \\ x_1 - 2x_2 \le 1 \\ x_1 \ge 0,\ x_2 \ge 0 \end{cases}$$



$$F_{max} = (-2x_1 + x_2)\Big|_{\substack{x_1=0 \\ x_2=1}} = 1$$

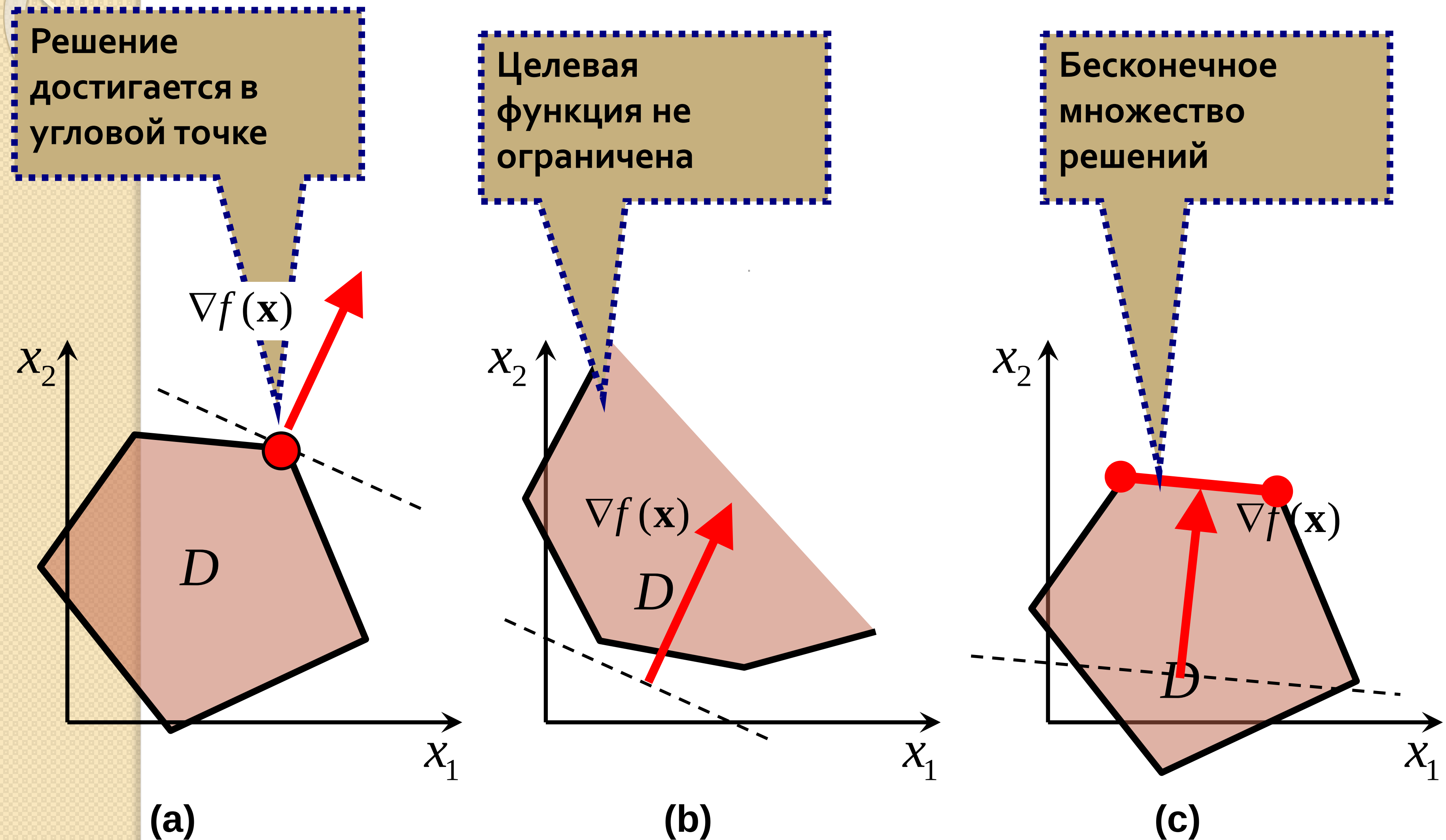
$F = 2x_1 - x_2 \to max$

$$\begin{cases} -x_1 + x_2 \le 1 \\ x_1 - 2x_2 \le 1 \\ x_1 \ge 0,\ x_2 \ge 0 \end{cases}$$



Функция ***F*** не достигает оптимального значения в допустимой области (она не ограничена).

Принципиальные ситуации, возможные при решении задачи линейного программирования



Опр. Множество ***D*** – точек ***n***-мерного евклидова пространства будем называть выпуклым, если для любых $x^{(1)} = \left(x_1^{(1)},\dots,x_n^{(1)}\right)$ и $x^{(2)} = \left(x_1^{(2)},\dots,x_n^{(2)}\right)$ и любых $\boldsymbol{\alpha \geq 0, \beta \geq 0}$, таких, что $\boldsymbol{\alpha+\beta=1}$, точка $x = \alpha x^{(1)} + \beta x^{(2)}$ также принадлежит ***D***.

Опр. Вершиной выпуклого множества в $\boldsymbol{R_n}$ назовем такую точку, которую нельзя представить в виде $x = \alpha x^{(1)} + \beta x^{(2)}$, $\boldsymbol{\alpha>0}$, $\boldsymbol{\beta>0}$, $\boldsymbol{\alpha+\beta=1}$, ни при каких $x^{(1)}, x^{(2)}$.

Свойства задачи линейного программирования

- **Свойство 1.** Допустимая область задачи линейного программирования выпукла, если она не пуста.
- **Свойство 2.** Если допустимая область имеет вершины и задача линейного программирования имеет решение, то оно достигается по крайней мере в одной из вершин.
- **Свойство 3.** Множество решений задачи линейного программирования выпукло.
- **Свойство 4.** Если допустимая область ограничена, то задача линейного программирования имеет оптимальное решение.
- **Свойство 5.** Необходимым и достаточным условием существования решения задачи линейного программирования на максимум (минимум) является ограниченность целевой функции сверху (соответственно снизу) в допустимой области.

Ошибки

1. Слайд 2: классификация и разрешимость ЗО в презентации идут в другом порядке
2. Слайд 12: пропущено многоточие между первым и m-ым ограничениями
3. Слайд 13: индекс при R должен принимать значения от 1 до m, индекс при C должен быть j.

исправлено

Ошибки нашли:

Ефимов

Атаханова

Лекция
Симплекс-метод

Домашова Д.В.

Идея симплекс-метода решения задачи ЛП

Свойства задачи линейного программирования наталкивают на следующую схему решения задачи линейного программирования, известную, как симплекс-метод.

Пусть рассматриваемая задача имеет непустое допустимое множество с вершинами.

Тогда:

- Тем или иным способом находим какую-нибудь вершину допустимого множества и по определенным критериям определяем, не является ли она оптимальной.

Если она оптимальна, то задача решена. Если нет, то

- используя определенные правила, проверяем, нельзя ли утверждать, что задача не имеет оптимального решения (целевая функция не ограничена сверху или, соответственно, снизу на допустимом множестве).

Если утверждать это можно, то задача неразрешима. Если нельзя, то

- по определенному правилу ищем новую, лучшую (в смысле значения целевой функции) вершину и переходим к пункту 1).

Идея симплекс-метода решения задачи ЛП

Для реализации предложенной схемы необходимо:

- указать способ нахождения вершины допустимого множества,
- критерии оптимальности, неразрешимости,
- способ перехода от одной вершины к другой, лучшей в смысле значения целевой функции.

Линейное программирование

Каноническая форма

Задачу линейного программирования представленную в виде:

$$f = c_1 x_1 + c_2 x_2 + \ldots + c_n x_n \to max(min)$$

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n = b_2 \\ \ldots \quad \ldots \quad \ldots \quad \ldots \quad \ldots \quad \ldots \\ a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n = b_m \end{cases} \qquad (3)$$

$$x_i \geq 0, \; i = \overline{1,n}$$

будем называть канонической задачей линейного программирования.

Линейное программирование
Приведение к канонической форме

Введем две дополнительные неотрицательные переменные x_4 и x_5 в первом и третьем ограничениях, так, чтобы получились ограничения-неравенства.

Переменную x_3 представим в виде: $x_3 = x_3' - x_3''$, где $x_3' \geq 0$, $x_3'' \geq 0$.

Получим: $F = x_1 - 2x_1 + x_3' - x_3'' \to max$

$$\begin{cases} x_1 + 2x_2 + 3x_3 + x_4 = 5 \\ 2x_1 + 3x_2 - 4(x_3' - 3x_3'') = 3 \\ -x_1 + x_2 - (x_3' - x_3'') - x_5 = 2 \\ x_1 \geq 0,\ x_2 \geq 0,\ x_3' \geq 0,\ x_3'' \geq 0,\ x_4 \geq 0,\ x_5 \geq 0, \end{cases}$$

Получили задачу в канонической форме.

Линейное программирование

Задача ЛП в векторном виде

Введем в рассмотрение:

$$\overline{C} = (c_1, c_2, ..., c_n), \quad \overline{X} = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}, \quad \overline{b} = \begin{pmatrix} b_1 \\ \vdots \\ b_n \end{pmatrix}$$

$$A = \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ a_{21} & \cdots & a_{2n} \\ \cdots & \cdots & \cdots \\ a_{m1} & \cdots & a_{mn} \end{pmatrix}$$

Перепишем задачу (3) линейного программирования в векторной форме:

$$f = (c, x) \to \max$$

$$Ax = b$$

$$x \geq 0$$

Линейное программирование

$$f = c_1 x_1 + c_2 x_2 + \dots + c_n x_n \to max$$

$$\begin{array}{l} a_{11}x_1 + \dots + a_{1n}x_n = b_1 \\ \dots \quad \dots \quad \dots \quad \dots \\ a_{m1}x_1 + \dots + a_{mn}x_n = b_m \\ x_1 \geq 0, \dots, x_m \geq 0 \end{array}$$

Введем в рассмотрение векторы

$$A_j = \begin{pmatrix} a_{1j} \\ a_{2j} \\ \vdots \\ a_{mj} \end{pmatrix}, \quad j = \overline{1,n}$$

$$f = (c, x) \to max$$

$$A_1 x_1 + A_2 x_2 + \dots + A_n x_n = b$$

$$x \geq 0$$

Задачу ЛП можно трактовать следующим образом: из всех разложений вектора ***b*** по векторам ***A1…An*** с неотрицательными коэффициентами требуется выбрать хотя бы одно такое, коэффициенты ***xj, j=I,n***, которого доставляют целевой функции ***f*** оптимальное значение.

Не ограничивая общности, считаем ранг матрицы ***A*** равным *m* и *n>m* (случай *n=m* – тривиален).

Опорные точки допустимого множества

Определение. Ненулевое допустимое решение $x = (x_1, ..., x_n)$ называется опорным, если векторы $\boldsymbol{A_j}$, соответствующие отличным от нуля координатам вектора x, линейно - независимы.

Определение. Ненулевое опорное решение назовем невырожденным, если оно имеет точно $\boldsymbol{m}$ положительных координат.

Определение. Если число положительных координат опорного решения меньше $\boldsymbol{m}$, то оно называется вырожденным.

Определение. Упорядоченный набор из $\boldsymbol{m}$ линейно-независимых векторов $\boldsymbol{A_j}$, соответствующих положительным координатам опорного решения назовем базисом.

Опорные точки допустимого множества

Пример. Дана система ограничений задачи:

$$2x_1 + x_2 + 3x_3 = 2$$
$$x_1 - 2x_3 + x_4 = 1$$
$$x_1 \ge 0,\ x_2 \ge 0,\ x_3 \ge 0,\ x_4 \ge 0$$

Здесь ***m=2***;

$$A_1 = \begin{pmatrix} 2 \\ 1 \end{pmatrix}, A_2 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, A_3 = \begin{pmatrix} 3 \\ -2 \end{pmatrix}, A_4 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

Классифицировать точки:

$$\overline{x}^{(1)} = (0,2,0,1),\quad \overline{x}^{(2)} = (1,0,0,0),\quad \overline{x}^{(3)} = (1/2,1,0,1/2)$$

Опорные точки допустимого множества
Теорема о связи опорного решения и вершины допустимого множества

Теорема.
Вектор $x = (x_1, ..., x_n)$ тогда и только тогда является опорным решением задачи, когда точка $x = (x_1, ..., x_n)$ является вершиной допустимого множества.

Таким образом, задача нахождения вершины допустимого множества свелась к задаче нахождения опорного решения, а, следовательно, к нахождению базиса.

Будем считать, что исходный базис $\boldsymbol{A_1}$, $\boldsymbol{A_2}$, … , $\boldsymbol{A_m}$ дан.
Отправляясь от него, покажем, как найти опорное решение.
Сформулируем условие оптимальности решения, условие отсутствия решения.
Покажем, как перейти к базису, дающему лучшее решение.

Симплекс-метод

$$f = (c, x) \to \max$$
$$Ax = b$$
$$x \geq 0$$
$$rg\ A = m$$

$A_1, \ldots, A_m$ - базис, $\Rightarrow$

$$A_B = \left[A_1 \ldots A_m\right],\ A_s = \left[A_{m+1} \ldots A_n\right]$$
$$A = \left[A_B, A_s\right]$$
$$x = \left[x_B, x_s\right], c = \left[c_B, c_s\right]$$

Симплекс-метод

$$A_x = b \Leftrightarrow$$

$$A_B x_B + A_s x_s = b$$

$$\Rightarrow$$

$$x_B + A_B^{-1} A_s x_s = A_B^{-1} b$$

$$x_B = A_B^{-1}\left(b - A_s x_s\right) \Rightarrow$$

$$x^0 = \left[A_B^{-1}\left(b - A_s x_s\right), 0\right] = \left[A_B^{-1} b, 0\right]$$

Симплекс-метод

$$f(x) = c_B x_B + c_s x_s =$$

Симплекс-метод

$$f(x) = c_B x_B + c_s x_s =$$

$$= c_B\left(A_B^{-1}b - A_B^{-1}A_s x_s\right) + c_s x_s =$$

Симплекс-метод

$$f(x) = c_B x_B + c_s x_s =$$
$$= c_B \left(A_B^{-1} b - A_B^{-1} A_s x_s \right) + c_s x_s =$$
$$= c_B A_B^{-1} b + \left(c_s - c_B A_B^{-1} A_s \right) x_s$$

Симплекс-метод

$$f(x) = c_B x_B + c_s x_s =$$
$$= c_B\left(A_B^{-1}b - A_B^{-1}A_s x_s\right) + c_s x_s =$$
$$= c_B A_B^{-1}b + \left(c_s - c_B A_B^{-1}A_s\right)x_s$$

$$f\left(x^{on}\right) = c_B A_B^{-1}b, \qquad \text{т.к. } x_s = 0 \;\Rightarrow$$

$$f(x) = f\left(x^{on}\right) + \left(c_s - c_B A_B^{-1}A_s\right)x_s$$

$$\Delta_s = c_s - c_B A_B^{-1}A_s, \qquad s = \overline{m+1,n} \;\Rightarrow$$

$$f(x) = f\left(x^{on}\right) + \Delta_s x_s = f\left(x^{on}\right) + \sum_{j=m+1}^{n}\Delta_j x_j$$

Симплекс-метод

Идея симплекс-метода состоит в том, чтобы исходя из начального опорного решения найти новое опорное решение, исключая для этого некоторый вектор A_s, $s \in \{1,\ldots,m\}$ из начального базиса и заменяя его одним из небазисных векторов A_r, $r \in \{m+1,\ldots n\}$ таким образом, чтобы новое опорное решение не ухудшало значения целевой функции.

Симплекс-метод

Выводы:

1) небазисный A_r имеет смысл вводить в новый базис, если $\Delta_r > 0$;

2) причем лучше всего такой A_r, у которого $\Delta_r > 0$ и самая большая из всех $r = \overline{m+1, n}$;

3) если все $\Delta_j \le 0,\ j = \overline{1, n}$, то значение целевой функции улучшить нельзя, следовательно, рассматриваемое опорное решение является оптимальным.

Симплекс-метод

Определим, какой вектор должен быть исключен из базиса.

Так как A_B – базис в R^m $\Rightarrow$ все столбцы матрицы A и b можно представить в виде линейных комбинаций данных столбцов

$$b = \sum_{j=1}^{m} A_j x_{j0} = [A_1, \ldots A_m] \begin{pmatrix} x_{10} \\ \ldots \\ x_{m0} \end{pmatrix} = A_B x^0$$

$$A_k = \sum_{j=1}^{m} A_j x_{jk} = [A_1, \ldots, A_m] \begin{pmatrix} x_{1k} \\ \ldots \\ x_{mk} \end{pmatrix} = A_B x^k \Rightarrow$$

Симплекс-метод

$$b = \sum_{j=1}^{m} A_j x_{j0} = [A_1, \ldots A_m] \begin{pmatrix} x_{10} \\ \ldots \\ x_{m0} \end{pmatrix} = A_B x^0$$

$$A_k = \sum_{j=1}^{m} A_j x_{jk} = [A_1, \ldots, A_m] \begin{pmatrix} x_{1k} \\ \ldots \\ x_{mk} \end{pmatrix} = A_B x^k \Rightarrow$$

$$\begin{cases} x^0 = A_B^{-1} b - \text{ненулевые координаты опорной точки} \\ x^k = A_B^{-1} A_k, \ k = \overline{1,n} - \text{координаты разложения вектора } A_k \text{ по базису} \end{cases}$$

$$\Delta_k = c_k - c_B A_B^{-1} A_k = c_k - c_B x^k = c_k - \sum_{j=1}^{m} c_{j+m} x_{jk}$$

Симплекс-метод

Новый базис получим из старого путем замены в нем некоторого базисного вектора A_s таким вектором A_r, который должен обеспечивать получение нового допустимого базисного решения, т.е. координаты вектора x_B должны быть ≥ 0.

$$x_B = A_B^{-1}b - A_B^{-1}A_s x_s = A_B^{-1}b - A_B^{-1}\sum_{j=m+1}^{n} A_j x_j = A_B^{-1}b - A_B^{-1}A_r x_r,$$

т.к. только новая базисная переменная $x_r>0$, все остальные небазисные переменные $x_j = 0,\ j = \overline{m+1,n} \setminus r$

$$x_B = A_B^{-1}b - A_B^{-1}A_r x_r$$

$A_B^{-1}b$ – ненулевые координаты опорной точки

$A_B^{-1}A_r$ – координаты разложения вектора A_r по базису

Симплекс-метод

правило выбора вектора, выводимого из базиса

Запишем в покоординатной форме.

$$x_{Bi} = x_{i0} - x_{ir}x_r \,, \; i = \overline{1,m}$$

Все $x_{Bi} \geq 0$, следовательно, $x_{i0} - x_{ir}x_r \geq 0 \; i = \overline{1,m}$.

Пусть $I \subset \{1,\ldots,m\}$ - множество индексов, для которых $x_{ir} > 0$

$$I = \{i : x_{ir} > 0\}$$

Найдем максимальное значение, до которого можно увеличивать значение переменной x_r:

$$x_r^{\max} = \min_{i \in I} \frac{x_{i0}}{x_{ir}}$$

Если $x_r := x_r^{\max}$, то, по крайней мере, одна базисная переменная x_s, где $s \in \{1,\ldots,m\}$ обратится в ноль и, следовательно, соответствующий ей столбец A_s матрицы A можно вывести из старого базиса.

Симплекс-метод

правило выбора вектора, выводимого из базиса

Выводится тот вектор As, для которого отношение координат опорного решения к положительным координатам разложения вектора Ar по базису является минимальным

$$\frac{x_{s0}}{x_{sr}} = \min_{i \in I} \frac{x_{i0}}{x_{ir}}$$

Симплекс-метод
Критерий отсутствия решения

Существует $\Delta_r > 0$, такая, что если все $x_{ir} \leq 0, i = \overline{1, m}$, то задача не имеет решения, т.е. существуют допустимые решения со сколь угодно большим значением целевой функции.

В этих случаях говорят, что целевая функция не ограничена сверху в допустимой области

Алгоритм Симплекс-метода

1. Для известного начального базиса находят координаты разложения векторов *b* и A_k $k=\overline{1,n}$ по базису:

$$\begin{cases} x^0 = A_B^{-1}b - \text{ненулевые координаты опорной точки} \\ x^k = A_B^{-1}A_k,\ k=\overline{1,n} - \text{координаты разложения вектора } A_k \text{ по базису} \end{cases}$$

2. Вычисляют симплекс-разности:

$$\Delta_k = c_k - c_B A_B^{-1} A_k,\ k=\overline{1,n}$$

3. Проверяют план на оптимальность:

Если все $\Delta_k \le 0,\ k=\overline{1,n},$

то решение оптимально.

4. Проверяется критерий отсутствия решения:

Если $\exists\, \Delta_r > 0: \text{все } x_{ir} \le 0,\ i=\overline{1,m}$, то целевая функция не ограничена сверху в допустимой области.

5. Определяют вектор A_r, вводимый в базис: $\Delta_r > 0$ и максимальная среди всех положительных $\Delta_k,\ k=\overline{1,n}$

6. Определяют вектор A_s, выводимый из базиса.

$$A_s : \frac{x_{s0}}{x_{sr}} = \min_{i \in I} \frac{x_{i0}}{x_{ir}} \quad (x_{ir} > 0) \Rightarrow \text{строим новый базис и переходим в п.1}$$

Алгоритм Симплекс-метода

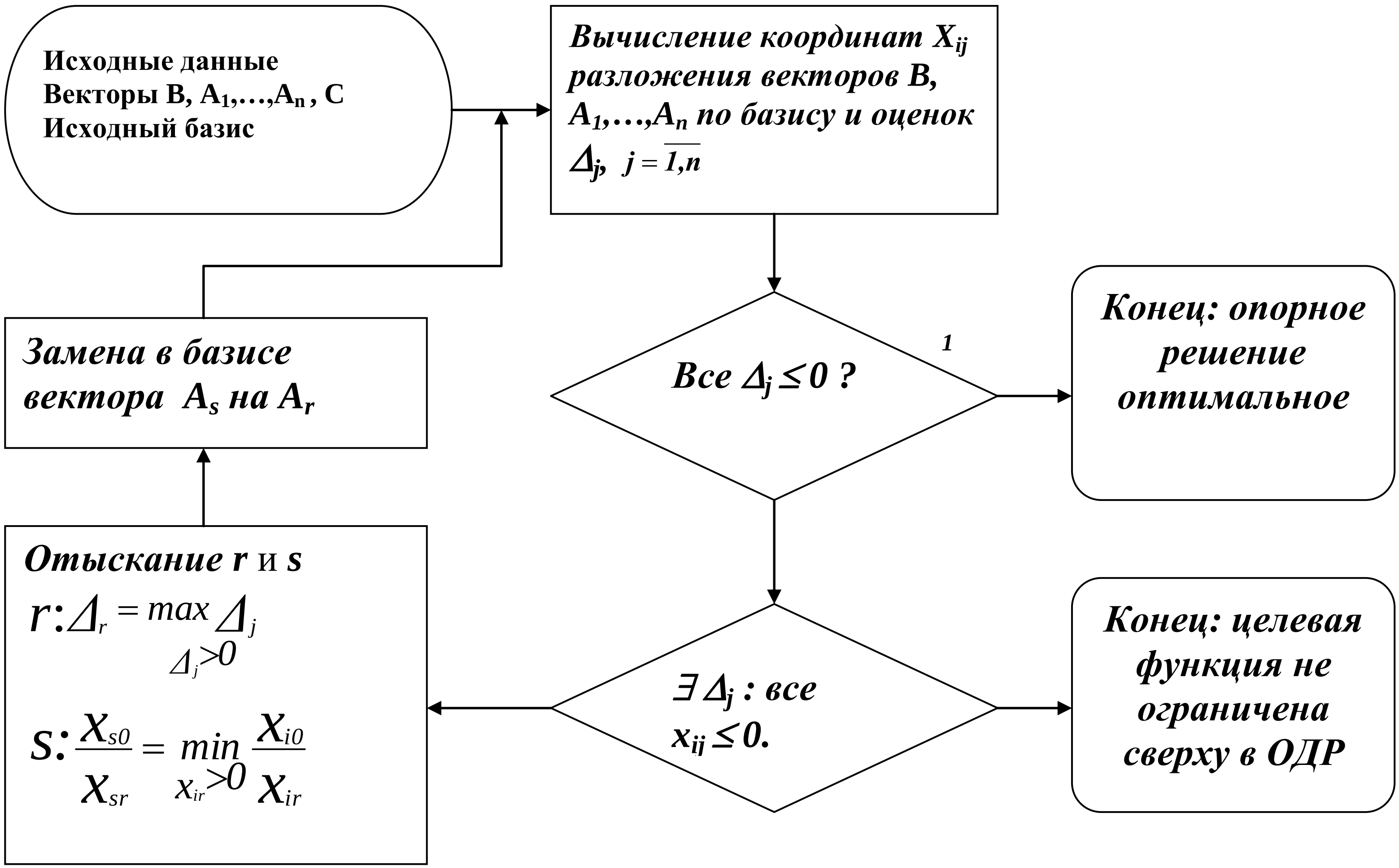


Рис. 3.

Симплекс-метод

Формулы пересчета координат разложения векторов по новому базису

$$x'_{jk} = \begin{cases} x_{jk} - \dfrac{x_{jr}}{x_{sr}} x_{sk}, & \text{если } j \in I \setminus s \\ \dfrac{x_{sk}}{x_{sr}}, & \text{если } j = r \end{cases}$$

Симплекс-метод

Формулы пересчета координат разложения векторов по новому базису

Доказательство

1) Вектор A_k через старый базис:

$$A_k = \sum_{j \in I} A_j x_{jk}, \quad k = \overline{1, n}$$

2) Вектор A_k через новый базис:

$$A_k = \sum_{j \in I \setminus s} A_j x'_{jk} = A_r x'_{rk}, \quad k = \overline{1, n}$$

Вектор

$$A_r = \sum_{j \in I} A_j x_{jr} = \sum_{j \in I \setminus s} A_j x_{jr} + A_s x_{sr} \Rightarrow$$

$$\Rightarrow A_s = \frac{A_r}{x_{sr}} - \sum_{j \in I \setminus s} A_j \frac{x_{jr}}{x_{sr}}, \quad \text{т.к. } x_{sr} \neq 0$$

Симплекс-метод

Формулы пересчета координат разложения векторов по новому базису
Доказательство

3) Подставим A_s в (1) и получим

$$A_k = \sum_{j \in I \setminus s} A_j x_{jk} + A_s x_{sk} =$$

$$= \sum_{j \in I \setminus s} A_j x_{jk} + \left(A_r \frac{x_{sk}}{x_{sr}} - \sum_{j \in I \setminus s} A_j \frac{x_{jr} x_{sk}}{x_{sr}} \right) =$$

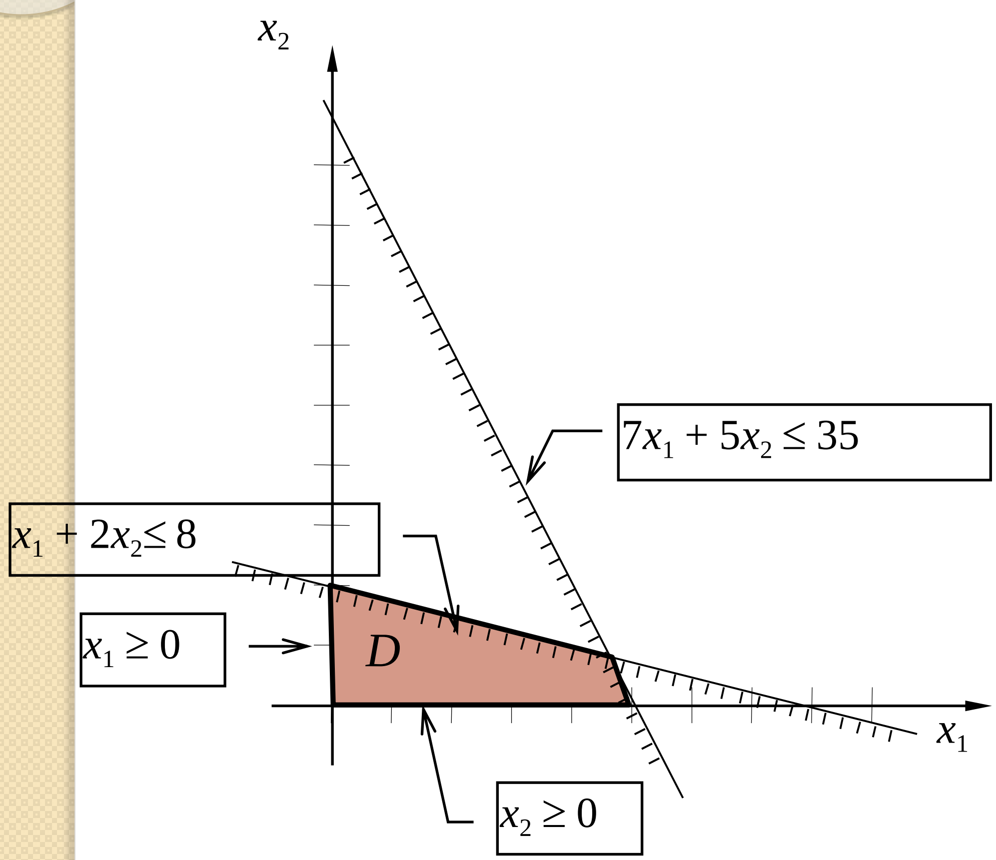
$$= \sum_{j \in I \setminus s} A_j \left(x_{jk} - \frac{x_{jr} x_{sk}}{x_{sr}} \right) + A_r \frac{x_{sk}}{x_{sr}}$$

$$x'_{jk} = \begin{cases} x_{jk} - \dfrac{x_{jr}}{x_{sr}} x_{sk}, \text{ если } j \in I \setminus s \\ \dfrac{x_{sk}}{x_{sr}}, \text{ если } j = r \end{cases}$$

Симплекс-метод
Симплекс-таблицы

базис	$c_{баз.}$	B	c_1		c_r		c_n
			A_1		A_r		A_n
A_{i1}	c_{i1}	x_{10}	x_{11}		x_{1r}		x_{1n}
…	…	…	…		…		…
A_{is}	c_{is}	x_{s0}	x_{s1}		x_{sr}		x_{sn}
…	…	…	…		…		…
A_{im}	c_{im}	x_{m0}	x_{mr}		x_{mr}		x_{mn}
		$f(x^{on})$	Δ_1		Δ_r		Δ_n

Симплекс-метод
Пример
Геометрическая интерпретация ЗЛП



$$f = 4x1 + 5x2 \to \max (\min)$$

$$\begin{cases} 7x_1 + 5x_2 \le 35 \\ x_1 + 2x_2 \le 8 \end{cases}$$

$$x_1 \ge 0, x_2 \ge 0$$

$$x_{\max} = \left(\frac{10}{3}, \frac{7}{3} \right)$$

$$f_{\max} = 25$$

Симплекс-метод
Пример

Базис	B опор.	C(базис)	A1	A2	A3	A4
A3	35	0	7	5	1	0
A4	8	0	1	2	0	1

Симплекс-метод
Пример

Базис	B опор.	C(базис)	A1	A2	A3	A4
A3	35	0	7	5	1	0
A4	8	0	1	2	0	1
	F=0		d1=-4	d2=-5	d3=0	d4=0

Симплекс-метод
Пример

Базис	В опор.	С(базис)	А1	А2	А3	А4
А3	35	0	7	5	1	0
А4	8	0	1	2	0	1
	F=0		d1=-4	d2=-5	d3=0	d4=0

Базис	В опор.	С(базис)	А1	А2	А3	А4
А3	15	0	4,5	0	1	-2,5
А2	4	5	0,5	1	0	0,5

Симплекс-метод
Пример

Базис	B опор.	C(базис)	A1	A2	A3	A4
A3	35	0	7	5	1	0
A4	8	0	1	2	0	1
	F=0		d1=-4	d2=-5	d3=0	d4=0

Базис	B опор.	C(базис)	A1	A2	A3	A4
A3	15	0	4,5	0	1	-2,5
A2	4	5	0,5	1	0	0,5
	F=20		d1=-1,5	d2=0	d3=0	d4=2,5

Симплекс-метод
Пример

Базис	В опор.	С(базис)	A1	A2	A3	A4
A3	35	0	7	5	1	0
A4	8	0	1	2	0	1
	F=0		d1=-4	d2=-5	d3=0	d4=0

Базис	В опор.	С(базис)	A1	A2	A3	A4
A3	15	0	4,5	0	1	-2,5
A2	4	5	0,5	1	0	0,5
	F=20		d1=-1,5	d2=0	d3=0	d4=2,5

N	Базис	С(базис)	В опор.	A1	A2	A3	A4
1	A1	4	3,33333333	1	0	0,22222222	-0,5555555
2	A2	5	2,33333333	0	1	-0,1111111	0,7777777

Симплекс-метод
Пример

Базис	В опор.	С(базис)	A1	A2	A3	A4
A3	35	0	7	5	1	0
A4	8	0	1	2	0	1
	F=0		d1=-4	d2=-5	d3=0	d4=0

Базис	В опор.	С(базис)	A1	A2	A3	A4
A3	15	0	4,5	0	1	-2,5
A2	4	5	0,5	1	0	0,5
	F=20		d1=-1,5	d2=0	d3=0	d4=2,5

N	Базис	С(базис)	В опор.	A1	A2	A3	A4
1	A1	4	3,33333333	1	0	0,22222222	-0,5555555
2	A2	5	2,33333333	0	1	-0,1111111	0,7777777
				d1=0	d2=0	d3=0,33333	d4=1,66666

Симплекс-метод
Пример

Базис	В опор.	С(базис)	A1	A2	A3	A4
A3	35	0	7	5	1	0
A4	8	0	1	2	0	1
	F=0		d1=-4	d2=-5	d3=0	d4=0

Базис	В опор.	С(базис)	A1	A2	A3	A4
A3	15	0	4,5	0	1	-2,5
A2	4	5	0,5	1	0	0,5
	F=20		d1=-1,5	d2=0	d3=0	d4=2,5

N	Базис	С(базис)	В опор.	A1	A2	A3	A4
1	A1	4	3,333333333	1	0	0,222222222	-0,55555555
2	A2	5	2,333333333	0	1	-0,11111111	0,77777777
				d1=0	d2=0	d3=0,333333	d4=1,666666

Симплекс-метод
Пример

Базис	В опор.	С(базис)	А1	А2	А3	А4
А3	35	0	7	5	1	0
А4	8	0	1	2	0	1
	F=0		d1=-4	d2=-5	d3=0	d4=0

Базис	В опор.	С(базис)	А1	А2	А3	А4
А3	15	0	4,5	0	1	-2,5
А2	4	5	0,5	1	0	0,5
	F=20		d1=-1,5	d2=0	d3=0	d4=2,5

N	Базис	С(базис)	В опор.	А1	А2	А3	А4
1	А1	4	3,33333333	1	0	0,22222222	-0,5555555
2	А2	5	2,33333333	0	1	-0,1111111	0,7777777
				d1=0	d2=0	d3=0,33333	d4=1,66666

Симплекс-метод

Поиск начальной опорной точки методом искусственного базиса

Будем считать, что для задачи (1)

$$< c, x > \to \max$$

$$D: Ax = b, \ \ x \geq 0 \qquad (1)$$

выполнено условие $b \geq 0$

Рассмотрим вспомогательную задачу:

$$-\sum_{i=1}^{m} y_i \to \max \qquad (2)$$

$$\tilde{D}: Ax + y = b, \ \ x \geq 0, \ y \geq 0$$

$$\tilde{D} = \left\{ (x, y) \in R_{n+m} : \sum_{j=1}^{n} A_j x_j + \sum_{i=1}^{m} e_i y_i = b, \ x \geq 0, \ y \geq 0 \right\} \Rightarrow$$

$(\overline{0}, b)$ - опорная точка в $\overline{D}$.

Целевая функция вспомогательной задачи ограничена сверху нулем (0). Следовательно, эта задача имеет оптимальное решение.

Симплекс-метод

Поиск начальной опорной точки методом искусственного базиса

$$G = -y_1 - y_2 - \dots - y_m \to max$$

$$\begin{cases} a_{11}x_1 + \dots + a_{1n}x_n + y_1 = b_1 \\ a_{21}x_1 + \dots + a_{2n}x_n + 0y_1 + y_2 = b_2 \\ \dots \quad \dots \quad \dots \quad \dots \quad \dots \quad \dots \\ a_{m1}x_1 + \dots + a_{mn}x_n + 0y_1 + \dots + y_m = b_m \end{cases} \qquad \begin{matrix} x_j \ge 0, \quad j = \overline{1,n} \\ y_i \ge 0, \quad i = \overline{1,m} \end{matrix}$$

Переменные $y_1, y_2, \dots, y_m$ – называют искусственными переменными.

Очевидно, что векторы $A_{n+1} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \ A_{n+2} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \ \dots, \ A_{n+m} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 1 \end{pmatrix}$

образуют базис для опорного решения $x = (\underbrace{0, 0, \dots, 0}_{n}, b_1, b_2, \dots, b_m)$,

который называют искусственным базисом.

Симплекс-метод
Поиск начальной опорной точки методом искусственного базиса

Утверждение.

Пусть (x^*, y^*) – решение задачи (2) и $f^* = -\sum_{i=1}^{m} y_i^*$

Если $f^*=0$, то x^* - опорная точка множества D.
Если $f^*<0$, то задача (1) не имеет допустимых точек: множество D - *пусто*.

Доказательство:

1) Если $f^*=0$, $y^*=0$, то $(x^*,0)$ – опорная точка множества $\tilde{D}$ и D, следовательно, оптимальный базис вспомогательной задачи можно взять в качестве начального для задачи (1).

2) Если $f^*<0$, то, если $\exists x \in D \Rightarrow \exists (x,0) \in \tilde{D}$, что несовместимо с условием $f^*<0 \Rightarrow$ задача (1) не имеет область допустимых решений.

Симплекс-метод

Поиск начальной опорной точки методом искусственного базиса

Решая вспомогательную задачу симплекс-методом, мы найдем оптимальное решение $x^{(0)} = (x_1^{(0)},\ldots,x_n^{(0)}, y_1^{(0)},\ldots,y_m^{(0)})$.

Если в этом решении среди искусственных переменных есть положительные, то исходная задача линейного программирования неразрешима, так ее ОДР пуста.

Если же $y_i^{(0)} = 0,\ i = \overline{1,m}$, то базис, соответствующий оптимальному решению вспомогательной задачи, можно взять в качестве исходного базиса основной задачи.

Поиск начальной опорной точки методом искусственного базиса
Пример

$F = 5x_1 + 2x_2 \to \max(\min)$

$$\begin{cases} 5x_1 - 6x_2 \le 30 \\ x_1 + 2x_2 \ge 4 \end{cases}$$

$x_1 \ge 0 \quad x_2 \ge 0$

$$\begin{cases} 5x_1 - 6x_2 = 30 \\ x_1 + 2x_2 = 4 \\ \end{cases}$$

$\overline{n} = \overline{grad} F = (5,2)$

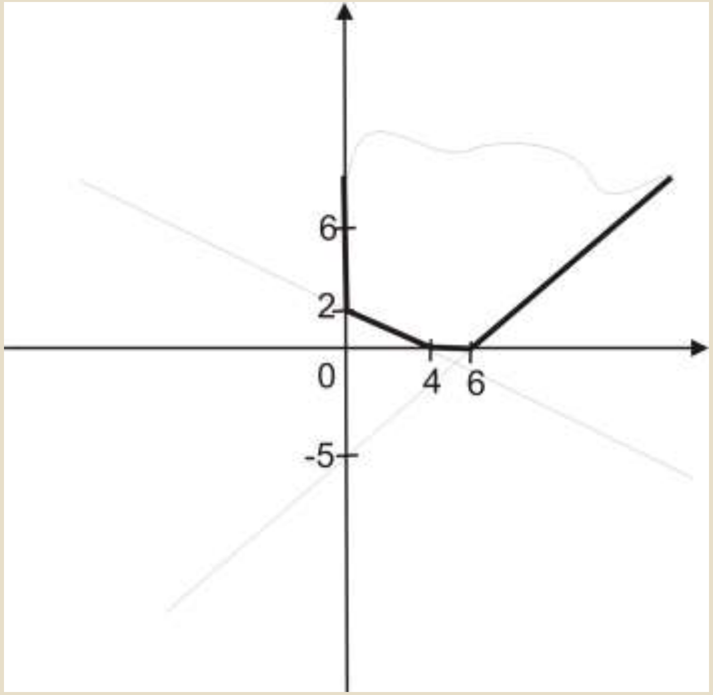
$$\begin{cases} x_1 + 2x_2 = 4 \\ x_1 = 0 \end{cases}$$

$x(0;2)$

$F = 4$

$x_{\min} = (0, 2)$

$F(x_{\min}) = 4$



Поиск начальной опорной точки методом искусственного базиса
Пример

$$F = 5x_1 + 2x_2 \to \max(\min)$$

$$\begin{cases} 5x_1 - 6x_2 + x_3 = 30 \\ x_1 + 2x_2 - x_4 = 4 \end{cases}$$

$$x_1 \geq 0 \quad x_2 \geq 0 x_3 \geq 0 x_4 \geq 0$$

$$G = -y_1 \to \max(\min)$$

$$\begin{cases} 5x_1 - 6x_2 + x_3 = 30 \\ x_1 + 2x_2 - x_4 + y_1 = 4 \end{cases}$$

$$x_1 \geq 0 \quad x_2 \geq 0 x_3 \geq 0 x_4 \geq 0 y_1 \geq 0$$

Базис	$C_{баз}$	$\overline{B}$ (опорное решение)	0	0	0	0	-1
			$\overline{A_1}$	$\overline{A_2}$	$\overline{A_3}$	$\overline{A_4}$	$\overline{A_5}$
$\overline{A_3}$	0	30	5	-6	1	0	0
$\overline{A5}$	-1	4	1	2	0	-1	1
Оценки		F=0	$\Delta_1 = 1$	$\Delta_2 = 2$	$\Delta_3 = 0$	$\Delta_4 = -1$	$\Delta_5 = 0$

Замечания

- Проблема зацикливания.
  - Вырожденные планы могут привести к зацикливанию, т.е. к многократному повторению процесса вычислений, не позволяющему получить оптимальный план.
  - Можно использовать метод Крено: Элементы строк, имеющие одинаковые наименьшие симплексные отклонения, делятся на предполагаемые разрешающие элементы. За ведущую выбирается та сторона, в которой раньше встретится наименьшее частное при просмотре слева направо по столбцам.
- Бесчисленное множество решений
  - Если в строке Δ оптимального плана находится нуль, принадлежащий свободной переменной, вектор которой не входит в базис, а в столбце этого вектора имеется хотя бы один положительный элемент, то задача имеет бесчисленное множество решений.
  - Свободные переменные можно ввести в базис, в результате будет получен новый оптимальный план с другим набором базисных переменных.

<u>Проверяли</u>: **Абаева Зарина и Бакшеева Татьяна (С18-702)**

Список опечаток :

№ слайда	Ошибка	Исправление
5	Опечатка в целевой функции	$F = x_1 - 2x_2 + x'_3$ -x''_3
5	Не расписано x_3 в первом ограничении	$x_1 + 2x_2 + 3\left(x'_3 - x''_3\right) + x_4 = 5$
5	Опечатка во втором ограничении	$2x_1 + 3x_2 - 4\left(x'_3 - x''_3\right) = 3$
12	Опечатка в 1ой строчке	Ax=b
30	Опечатка в индексе последней координаты вектора А1	x_{m1}
45	Последняя строка таблицы, целевая функция	G=-4

Лекция
Двойственные задачи ЛП

Домашова Д.В.

Определение двойственной задачи

Рассмотрим задачу ЛП в общей форме

$$F = c_1x_1 + \ldots + c_nx_n \to \max$$

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n \le b_1 \\ . \\ a_{k1}x_1 + a_{k2}x_2 + \ldots + a_{kn}x_n \le b_k \\ . \\ a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n = b_m \end{cases}$$

(1)

$$x_j \ge 0,\ j = \overline{1,l},\ l \le n$$

Определение: Задача

$$F^* = b_1y_1 + \ldots + b_my_m \to \min$$

$$\begin{cases} a_{11}y_1 + a_{21}y_2 + \ldots + a_{m1}y_m \ge c_1 \\ . \\ a_{1l}y_1 + a_{2l}y_2 + \ldots + a_{ml}y_m \ge c_l \\ . \\ a_{1n}y_1 + a_{2n}y_2 + \ldots + a_{mn}y_m = c_n \end{cases}$$

$$y_i \ge 0, i = \overline{1,k}, k \le m$$

называется двойственной к задаче (1).

Определение двойственной задачи

	Прямая задача	Двойственная задача
1	n - переменных	m - переменных
2	m - ограничений	n - ограничений
3	Целевая функция – ищется max	Целевая функция – ищется min
4	c – вектор коэффициентов целевой функции	b – вектор коэффициентов целевой функции
5	b – вектор свободных членов системы ограничений	c – вектор свободных членов системы ограничений
6	A – матрица коэффициентов системы ограничений	$A^т$ – матрица коэффициентов системы ограничений
7	$x_j >=0$, j=1,k	j-ое ограничение «>=»,j=1,k
8	x_j – не ограничена в знаке, j=k+1,n	j-ое ограничение «=»,j=k+1,n
9	i-ое ограничение «<=»,i=1,l	$y_i >=0$, i=1,l
10	i-ое ограничение «=»,i=l+1,m	y_i, – не ограничена в знаке, i=l+1,m

Пример постановки двойственной задачи

$$F = x_1 - 4x_2 - 3x_3 \rightarrow \min$$
$$\begin{cases} 3x_1 + 4x_2 + x_3 \le 7 \\ x_1 + 2x_2 + x_3 = 6 \\ x_3 \ge 4 \end{cases}$$
$$x_j \ge 0,\ \ j = \overline{1,3}$$

$$F = -\ x_1 + 4x_2 + 3x_3 \rightarrow \max$$
$$\begin{cases} 3x_1 + 4x_2 + x_3 \le 7 \\ -x_3 \le -4 \\ x_1 + 2x_2 + x_3 = 6 \end{cases}$$
$$x_j \ge 0,\ \ j = \overline{1,3}$$

$$F^* = 7y_1 - 4y_2 + 6y_3 \rightarrow \min$$
$$\begin{cases} 3y_1 - y_3 \le -1 \\ 4y_1 - 2y_3 \le 4 \\ y_1 - y_2 + y_3 = 3 \end{cases}$$
$$y_{1,2} \ge 0$$

Симметричная пара двойственных задач

$$F = \langle c, x \rangle \to \max$$

$$Ax \le b$$
$$x \ge 0$$

$$F^* = \langle b, y \rangle \to \min$$

$$A^T y \ge C$$
$$y \ge 0$$

Основные теоремы двойственности

Теорема 1. Если одна из пары двойственных задач имеет оптимальное решение, то и другая имеет оптимальное решение, причем значения целевых функций задач при их оптимальных планах равны между собой: F(x*) = F*(y*). Если же целевая функция одной из пары двойственных задач не ограничена, то другая задача вообще не имеет планов (ОДР пуста).

Теорема2: $x^* = (x_1^*, \ldots, x_n^*)$ и $y^* = (y_1^*, \ldots., y_m^*)$ - оптимальные решения прямой и двойственной задач $\Leftrightarrow$

$$(\sum_{j=1}^{n} a_{ij} x_j^* - b_i) y_i^* = 0, i = \overline{1, m}$$

$$(\sum_{i=1}^{m} a_{ij} y_i^* - c_j) x_j^* = 0, \ j = \overline{1, n}$$

Основные теоремы двойственности

Теорема 3. $y^* = C_b A_B^{-1}$

Доказательство.
Пусть прямая задача:

$$F = \langle c, x \rangle \to \max_{\substack{Ax=b \\ x \ge 0}}$$

Тогда двойственная:

$$F^* = \langle y, b \rangle \to \min_{A^T y \ge C}$$

Пусть x^* - оптимальное решение прямой.
Тогда $A_B x^*_b = b$, $A_B^{-1} A_B x_b^* = A_B^{-1} b$, $x_b^* = A_B^{-1} b$.
Подставим x^* в целевую функцию:
$F = \langle c, x^* \rangle = c_b x^*_b = C_b A_B^{-1} b$,
$C_b A_B^{-1} b = y^* b$ тогда $y^* = C_b A_B^{-1}$,
где C_b – коэффициенты при базисных переменных;
A_B^{-1} - обратная матрица к матрице, составленной из компонент векторов, вошедших в оптимальных базис (расположена в первых m строках последней (оптимальной) симплекс-таблицы, в столбцах векторов, представляющих начальный базис.
При этом $y^* = C_b A_B^{-1}$ - находится в строке Δ

Основные теоремы двойственности

Установим соответствие между переменными прямой и двойственной задач в симплекс-таблице:

$$\underbrace{x_1 \ldots x_n}_{\text{основные}} \qquad \underbrace{x_{n+1} \ldots x_{n+m}}_{\text{дополнительные}}$$

$$\underbrace{y_{m+1} \ldots y_{m+n}}_{\text{дополнительные}} \qquad \underbrace{y_1 \ldots y_m}_{\text{основные}}$$

Пример

$$f = 4x1 + 5x2 \rightarrow \max\ (\min)$$

$$\begin{cases} 7x_1 + 5x_2 \le 35 \\ x_1 + 2x_2 \le 8 \end{cases}$$

$$x_1 \ge 0, x_2 \ge 0$$

$$x_{\max} = \left(\frac{10}{3}, \frac{7}{3}\right)$$

$$f_{\max} = 25$$

$$f^* = 35y1 + 8y2 \rightarrow \min$$

$$\begin{cases} 7y_1 + y_2 \ge 4 \\ 5x_1 + 2x_2 \ge 5 \end{cases}$$

$$y_1 \ge 0, y_2 \ge 0$$

Пример

$$f = 4x1 + 5x2 \to \max\ (\min)$$

$$\begin{cases} 7x_1 + 5x_2 \le 35 \\ x_1 + 2x_2 \le 8 \end{cases}$$

$$x_1 \ge 0, x_2 \ge 0$$

$$x_{\max} = \left(\frac{10}{3}, \frac{7}{3}\right)$$

$$f_{\max} = 25$$

$$f^* = 35y1 + 8y2 \to \min$$

$$\begin{cases} 7y_1 + y_2 \ge 4 \\ 5x_1 + 2x_2 \ge 5 \end{cases}$$

$$y_1 \ge 0, y_2 \ge 0$$

$$\begin{cases} 7 \cdot \frac{10}{3} + 5 \cdot \frac{7}{3} = 35 \Rightarrow y_1^* > 0 \\ \frac{10}{3} + 2 \cdot \frac{7}{3} = 8 \Rightarrow y_2^* > 0 \end{cases}$$

$$\begin{cases} x_1^* > 0 \Rightarrow 7y_1^* + y_2^* = 4 \\ x_2^* > 0 \Rightarrow 5y_1^* + 2y_2^* = 5 \end{cases}$$

$$y_1^* = \frac{1}{3}, y_2^* = \frac{5}{3}$$

$$f_{\min}^* = 25$$

Пример

N	Базис	C(базис)	B опор.	A1	A2	A3	A4
1	A1	4	3,333333333	1	0	0,222222222	-0,55555555
2	A2	5	2,333333333	0	1	-0,11111111	0,777777777
				d1=0	d2=0	d3=0,333333	d4=1,666666

$$y^* = C_b A_B^{-1} = (4{,}5) \cdot \begin{pmatrix} 0{,}222 & -0{,}555 \\ -0{,}111 & 0{,}777 \end{pmatrix} = (0{,}333;1{,}666)$$

Экономическая интерпретация двойственных задач

Экономическая интерпретация двойственных задач

Задача об оптимальном плане производства продукции

- n – видов продукции, ;
- m – видов ресурсов (сырья), ;
- a_{ij} – количество ресурса i-го вида, требующегося для производства единицы продукции j-го вида;
- b_i – запасы ресурса i-го вида ;
- c_j – доход (прибыль) от реализации единицы продукции j-го вида.

Экономическая интерпретация двойственных задач

Задача об оптимальном плане производства продукции

- Необходимо найти такой план производства продукции, при котором достигается максимальная прибыль, для реализации которого достаточно имеющихся ресурсов.
- Оценить каждый из видов сырья, используемых для производства продукции. Оценки, приписываемые каждому из видов сырья должны быть такими, чтобы оценка всего используемого сырья была минимальна, а суммарная оценка сырья, используемого на производство единицы продукции любого вида, - не меньше цены единицы продукции данного вида.
- Найти интервалы устойчивости двойственных оценок по отношению к изменениям ресурсов каждого типа.

Экономическая интерпретация двойственных задач

	A	B	C	D	Запасы
C_1	1	0	2	1	180
C_2	0	1	3	2	210
C_3	4	2	0	4	800
Цена за единицу продукции	9	6	4	7	

Экономическая интерпретация двойственных задач

Построим модели

Экономическая интерпретация двойственных задач

Построим модели

$$F = 9x_1 + 6x_2 + 4x_3 + 7x_4 \to \max$$

$$\begin{cases} x_1 + 2x_3 + x_4 \le 180 \\ x_2 + 2x_3 + x_4 \le 210 \\ 4x_1 + 2x_2 + 4x_4 \le 800 \end{cases}$$

$$x_j \ge 0,\ j = \overline{1,4}$$

Экономическая интерпретация двойственных задач

Построим модели

$$F = 9x_1 + 6x_2 + 4x_3 + 7x_4 \to \max$$

$$\begin{cases} x_1 + 2x_3 + x_4 \le 180 \\ x_2 + 2x_3 + x_4 \le 210 \\ 4x_1 + 2x_2 + 4x_4 \le 800 \end{cases}$$

$$x_j \ge 0,\ j = \overline{1,4}$$

$$F^* = 180y_1 + 210y_2 + 800y_3 \to \min$$

$$\begin{cases} y_1 + 4y_3 \ge 9 \\ y_2 + 2y_3 \ge 6 \\ 2y_1 + 3y_2 \ge 4 \\ y_1 + 2y_2 + 4y_3 \ge 7 \end{cases}$$

$$y_i \ge 0, i = \overline{1,3}$$

Экономическая интерпретация двойственных задач

Приведем к канонической форме

$$F = 9x_1 + 6x_2 + 4x_3 + 7x_4 \to \max$$

$$\begin{cases} x_1 + 2x_3 + x_4 + x_5 = 180 \\ x_2 + 3x_3 + 2x_4 + x_6 = 210 \\ 4x_1 + 2x_2 + 4x_4 + x_7 = 800 \end{cases}$$

$$x_j \geq 0,\ j = \overline{1,7}$$

Экономическая интерпретация двойственных задач

базис	Сб.	B	9	6	4	7	0	0	0
			A1	A2	A3	A4	A5	A6	A7
A5	0	180	[1]	0	2	1	1	0	0
A6	0	210	0	1	3	2	0	1	0
A7	0	800	4	2	0	4	0	0	1
		F = 0	-9	-6	-4	-7	0	0	0

- При данном плане ничего не производится, сырье не используется, F = 0.
- Δj показывают на сколько увеличится F (цена за произведенную продукцию) при введении в план единицы j-го вида продукции.
- Отсюда следует, что целесообразно включить в план изделие А в объеме min{180/1, 800/4} = 180.
- Тогда сможем изготовить 180 единиц изделия А. На это потребуется 180 единиц С1 и 180 · 4 С3.
- Т.е. максимум количества изделия А ограничивается запасами сырья С1. При этом все сырье С1 израсходуется.

Экономическая интерпретация двойственных задач

Оптимальная симплекс-таблица

базис	Сб.	В	9	6	4	7	0	0	0
			A1	A2	A3	A4	A5	A6	A7
A1	9	95	1	0	-3/2	0	0	-1/2	1/4
A5	0	85	0	0	7/2	1	1	1/2	-1/4
A2	6	210	0	1	3	2	0	1	0
		2115	0	0	1/2	5	0	3/2	9/4

$$x^* = (95,210,0,0) \quad y^* = (0, {}^3\!/_2, {}^9\!/_4)$$

При оптимальном плане производится 95 изделий А, 210 изделий В, при этом остается неиспользованными 85 единиц С1.

Экономическая интерпретация двойственных задач

1. Подставим x^* в ограничения прямой задачи:

$$\begin{cases} 95 + 2\cdot 0 + 0 < 180 \\ 210 + 3\cdot 0 + 2\cdot 0 = 210 \\ 4\cdot 95 + 2\cdot 210 + 4\cdot 0 = 800 \end{cases}$$

Второе и третье ограничения выполняются как «=» => ресурсы 2-го и 3-го видов полностью используются в оптимальном плане, являются дефицитными ($y_2^* = 3/2 > 0, y_3^* > 0$).

Первое ограничение выполняется как строгое «<» => ресурс первого вида не является дефицитным ($y_1^* = 0$). Его остатки $x_5^* = 85$
=> положительную двойственную оценку имеют лишь те виды ресурсов, которые полностью используются в оптимальном плане.

Экономическая интерпретация двойственных задач

2. Подставим y^* в ограничение двойственной задачи

$$\left\{\begin{aligned} &0+4\cdot\frac{9}{4}=9 \\ &\frac{3}{2}+2\cdot\frac{9}{4}=6 \\ &2\cdot 0+3\cdot\frac{3}{2}>4 \\ &0+2\cdot\frac{3}{2}+4\cdot\frac{9}{4}>7 \end{aligned}\right\}$$

Первое и второе ограничения выполняются как «=» => двойственные оценки ресурсов, используемых для производства единицы продукции A и B, равны в точности доходам => производить эти изделия целесообразно => $x_1^*=95>0$, $x_2^*=210>0$.

Третье и четвертое ограничения выполняются как «>» => производить изделия C и D экономически не выгодно => $x_3^*=0$, $x_4^*=0$.

Экономическая интерпретация двойственных задач

3. Величина двойственной оценки показывает, насколько возрастает значение целевой функции при увеличении дефицитного ресурса на одну единицу.

Увеличение ресурса С2 на одну единицу приведет к получению нового оптимального плана, в котором прибыль возрастает на 3/2: 2115+3/2. При этом коэффициенты матрицы A_b^{-1} (столбца А6) оптимальной симплекс-таблицы показывают, что указанное увеличение прибыли достигается за счет уменьшения выпуска изделий А на ½ единицы, увеличения выпуска изделия В на 1 единицу и увеличения остатка ресурса С1 на ½ единицы (использования ресурса расхода С1 сократится на ½ единицы).

Увеличение ресурса С3 на 1 единицу приведет к получению нового оптимального плана, в котором прибыль возрастает на 9/4: 2115 + 9/4. Это произойдет за счет увеличения выпуска изделия А на ¼ единицы, при этом расход сырья С1 возрастает (остаток уменьшится) на ¼ единицы.

Экономическая интерпретация двойственных задач

- Двойственные оценки связаны с оптимальным планом прямой задачи. Всякое изменение исходных данных прямой задачи оказывает влияние на ее оптимальный план и на систему двойственных оценок.
- В свою очередь двойственные оценки служат инструментом анализа и принятия правильного решения в условиях меняющихся коммерческих ситуаций.

Анализ устойчивости двойственных оценок

Будем рассматривать максимальные значения целевой функции прямой задачи, как функцию свободных членов системы ограничений:

$F_{\max}(b_1,\ldots,b_m)$.

Утверждение: В оптимальном плане двойственной задачи значение переменной y_i^* численно равно частной производной функции $F_{\max}(b_1,\ldots,b_m)$ по соответствующему аргументу: $\frac{\partial F_{\max}}{\partial b_i} = y_i^*$, $i=\overline{1,m}$

Двойственные оценки ресурсов показывают, на сколько единиц изменяется доход (F) от реализации продукции при изменении запаса соответствующего ресурса на одну единицу.

Большей оценке соответствует наиболее дефицитный ресурс. Для недефицитного ресурса $y_i^* = 0$.

Анализ устойчивости двойственных оценок

Представляет интерес определить такие интервалы изменения b_i, в которых оптимальный план двойственной задачи не меняется.

$$x_b^* = A_B^{-1}(b + \Delta b) = A_B^{-1}b + A_B^{-1}\Delta b = x_b + A_B^{-1}\Delta b$$

Это имеет место для всех тех значений $b_i + \Delta b_i$, при которых x_b^* не содержит отрицательных (т.е. являются допустимым решением).

$$A_B^{-1}\begin{pmatrix} b_1 + \Delta b_1 \\ . \\ . \\ b_m + \Delta b_m \end{pmatrix} \geq 0$$

Анализ устойчивости двойственных оценок

Определим интервалы устойчивости для нашей задачи:

$$A_B^{-1} = \begin{pmatrix} 0 & -1/2 & 1/4 \\ 1 & 1/2 & -1/4 \\ 0 & 1 & 0 \end{pmatrix} \quad b = \begin{pmatrix} 180 \\ 210 \\ 800 \end{pmatrix}$$

$$A_B^{-1}(b+\Delta b) = \begin{pmatrix} 0 & -1/2 & 1/4 \\ 1 & 1/2 & -1/4 \\ 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 180+\Delta b_1 \\ 210+\Delta b_2 \\ 800+\Delta b_3 \end{pmatrix} =$$

$$\begin{pmatrix} 95 - 1/2 \cdot \Delta b_2 + y_4 \cdot \Delta b_3 \\ 85 + \Delta b_1 + 1/2 \cdot \Delta b_2 - 1/4 \cdot \Delta b_3 \\ 210 + \Delta b_2 \end{pmatrix} \geq 0$$

Анализ устойчивости двойственных оценок

Частные случаи

1) Если $\Delta b_2 = 0$, $\Delta b_3 = 0$ => $85 + \Delta b_1 \geq 0$ => $\Delta b_1 \geq -85$

2) Если $\Delta b_1 = 0$, $\Delta b_3 = 0$

$$\begin{cases} 95 - 1/2 \cdot \Delta b_2 \geq 0 \\ 210 + \Delta b_2 \geq 0 \\ 85 + 1/2 \cdot \Delta b_3 \geq 0 \end{cases} \quad \begin{matrix} \Delta b_2 \leq 190 \\ \Delta b_2 \geq -210 \\ \Delta b_2 \geq -170 \end{matrix} \Rightarrow -170 \leq \Delta b_2 \leq 190$$

3) Если $\Delta b_1 = 0$, $\Delta b_2 = 0$

$$\begin{cases} 95 + 1/4 \cdot b_3 \geq 0 \\ 85 - 1/4 \cdot b_3 \geq 0 \end{cases} \quad \begin{matrix} \Delta b_3 \geq -380 \\ \Delta b_3 \leq 340 \end{matrix} \qquad -380 \leq \Delta b_3 \leq 340$$

=> интервалы изменения ресурсов:

$$210 - 170 \leq b_2 \leq 210 + 190 \qquad 800 - 380 \leq b_3 \leq 800 + 380$$

$$40 \leq b_2 \leq 400 \qquad 420 \leq b_3 \leq 1140$$

Первый вид ресурса в оптимальном плане недоиспользован, является недефицитным.
Увеличение данного ресурса приведет лишь к росту его остатка.

При этом изменений в оптимальном плане не будет, т.к. $y_1^* = 0$.

Анализ устойчивости двойственных оценок

Предельные значения изменения всякого из ресурсов, для которого двойственные оценки остаются неизменными, определяются следующим образом:

$$\Delta b_i^- = \max_{x_{ji>0}} \left\{ \frac{-x_j^*}{x_{ji}} \right\} \le \Delta b_i \le \min_{x_{ji}<0} \left\{ \frac{-95}{-1/2} \right\}$$

$$-85 \le \Delta b_1$$

$$-170 \le \Delta b_2 \le 190$$

$$\max_{x_{ji}>0} \left\{ \frac{-95}{1/4} \right\} \le \Delta b_3 \le \left\{ \frac{-85}{-1/4} \right\} \quad -380 \le \Delta b_3 \le 340$$

Анализ устойчивости двойственных оценок

Аналогично можно определить интервалы устойчивости с точки зрения дохода изделия.

$$\max_{x_{jk}<0}\left\{\frac{x_j^*}{x_{jk}}\right\} \le x_k \le \min_{x_{jk}>0}\left\{\frac{x_j^*}{x_{jk}}\right\}$$

Пример для x_3 и x_4:

$$\max\left\{\frac{95}{-3/2}\right\} \le x_3 \le \min\left\{\frac{85}{7/2}, \frac{210}{3}\right\} \quad -60 \le x_3 \le \left[\frac{170}{7}\right] = 24$$

$$\max\{\varnothing\} \le x_4 \le \min\left\{\frac{85}{1}, \frac{210}{2}\right\} = 85 \quad x_4 \le 85$$

=> В производство можно вводить изделие С до 24 единиц, или изделие D до 85 единиц.

Ошибки, упомянутые в лекции, представлены в качестве заметок к каждому слайду

Сушкевич Владислав
Калугер Роман

Методы оптимизации

Транспортная задача

Д.В. Домашова

Постановка транспортной задачи

m – поставщиков однородной продукции (источников)
n – потребителей однородной продукции (стоков)

a_i - запасы i-го поставщика

b_j - потребности (спрос) j-го потребителя

c_{ij} - стоимость перевозки из пункта i в j

Постановка транспортной задачи

m – поставщиков однородной продукции (источников)
n – потребителей однородной продукции (стоков)

a_i - запасы i-го поставщика

b_j - потребности (спрос) j-го потребителя

c_{ij} - стоимость перевозки из пункта i в j

Требуется найти такой план перевозок продукции от поставщиков к потребителям, который обеспечивал бы спрос потребителей и вывоз продукции от поставщиков при минимальных суммарных транспортных расходах

Постановка транспортной задачи

m – поставщиков однородной продукции (источников)
n – потребителей однородной продукции (стоков)

a_i - запасы i-го поставщика

b_j - потребности (спрос) j-го потребителя

c_{ij} - стоимость перевозки из пункта i в j

x_{ij} - количество груза, перевезенного из пункта i в j

Постановка транспортной задачи

m – поставщиков однородной продукции (источников)
n – потребителей однородной продукции (стоков)

a_i - запасы i-го поставщика

b_j - потребности (спрос) j-го потребителя

c_{ij} - стоимость перевозки из пункта i в j

x_{ij} - количество груза, перевезенного из пункта i в j

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

$$\sum_{j=1}^{n} x_{ij} \le a_i, \quad i = \overline{1,m}$$

$$\sum_{j=1}^{m} x_{ij} \ge b_j, \quad j = \overline{1,n}$$

$$x_{ij} \ge 0, \quad i = \overline{1,m}, \quad j = \overline{1,n}$$

Постановка транспортной задачи

Определение: Транспортная задача, в которой сумма запасов равна сумме потребностей, называется закрытой. В противном случае задача – открытая.

В случае, если транспортная задача является открытой, невозможно удовлетворить всех потребителей (если сумма потребностей больше суммы запасов) или вывезти все грузы от поставщиков (если сумма запасов больше, чем сумма потребностей).

Классическая транспортная задача

m – поставщиков однородной продукции (источников)
n – потребителей однородной продукции (стоков)

a_i - мощность i-го источника

b_j - мощность j-го стока

c_{ij} - стоимость перевозки из пункта i в j

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

$$\sum_{j=1}^{n} x_{ij} = a_i\,,\ i = \overline{1,m}$$

$$\sum_{j=1}^{m} x_{ij} = b_j\,,\ j = \overline{1,n}$$

$$x_{ij} \geq 0\,,\ i = \overline{1,m}\,,\ \ j = \overline{1,n}$$

Классическая транспортная задача

Приведение открытой ТЗ к закрытой

1) Если сумма запасов больше суммы потребностей ($\sum_{i=1}^{m} a_i > \sum_{j=1}^{n} b_j$), то введем в таблицу еще одного потребителя, потребность которого определим, как $\sum_{i=1}^{m} a_i - \sum_{j=1}^{n} b_j$ Так как грузы к новому потребителю (фиктивному) отправляться не будут, то и стоимость этих перевозок равна нулю т.е. цены (тарифы) в новой строке будут равны 0.

2) Если сумма запасов меньше суммы потребностей ($\sum_{i=1}^{m} a_i < \sum_{j=1}^{n} b_j$), то вводим в таблицу еще одного поставщика, запас груза у которого определим, как $\sum_{j=1}^{n} b_j - \sum_{i=1}^{m} a_i$. Цены в новом столбце проставим равными нулю из тех же соображений, что и в первом случае.

Решение транспортной задачи

1) Любая транспортная задача, как задача ЛП, может быть решена симплекс-методом, однако, специфика задач рассмотренного класса (каждая неизвестная входит лишь в два уравнения-ограничения и коэффициенты при неизвестных в ограничениях равны единице) позволила выработать более эффективные вычислительные методы.

2) Транспортную задачу можно представить с помощью сети => можно использовать для их решения эффективные алгоритмы.

Решение транспортной задачи

- Теорема: необходимым и достаточным условием разрешимости транспортной задачи является равенство суммы запасов сумме потребностей.
- Так как транспортная задача является задачей линейного программирования, то и методика нахождения оптимального решения остается той же:
  - находится первоначальный опорный план,
  - проверяется на оптимальность и если план не оптимален, то
  - переход к другому опорному плану, улучшающему целевую функцию в смысле оптимума (а именно уменьшающую значение целевой функции).
- Критерий отсутствия решения не требуется, так как решению подлежат лишь закрытые ТЗ

Решение транспортной задачи методом потенциалов

- Решение ТЗ проводится на основе теории двойственности.
- Поставим двойственную к закрытой ТЗ.

	$x_{11}\ x_{12}\ \dots\ x_{1n}$	$x_{21}\ x_{22}\ \dots\ x_{2n}$	…	$x_{m1}\ x_{m2}\ \dots\ x_{mn}$	Мощность	
1 2 . . . m	1 1 . . . 1	 1 1 . . . 1	 …	 1 1 . . . 1	a_1 a_2 . . . a_m	u_1 u_2 . . . u_m
					Спрос	
1 2 . . . n	1 1 . . . 1	1 1 . . . 1	 …	1 1 . . . 1	b_1 b_2 . . . b_j	v_1 v_2 . . . v_n
	$c_{11}\ c_{12}\ \dots\ c_{1n}$	$c_{21}\ c_{22}\ \dots\ c_{2n}$	…	$c_{m1}\ c_{m2}\ \dots\ c_{mn}$		

Решение транспортной задачи методом потенциалов

Двойственная задача

$$G = (u_1, \ldots, u_m, v_1, \ldots, v_n) = \sum_{i=1}^{m} a_i u_i + \sum_{j=1}^{n} b_j v_j \to \max$$

$$u_i + v_j = c_{ij}$$

где переменные u_i , v_j - не ограничены в знаке.

Решение транспортной задачи методом потенциалов

Из второй теоремы двойственности =>

$$(u_i^* + v_i^* - c_{ij}) \cdot x_{ij}^* = 0$$

т.е.

$$u_i^* + v_j^* = c_{ij} \quad \forall x_{ij}^* \neq 0 \quad (1)$$

$$u_i^* + v_j^* \leq c_{ij} \quad \forall x_{ij}^* = 0 \quad (2)$$

Решение транспортной задачи методом потенциалов

- **Идея решения транспортной задачи**
- На каждой итерации решения ТЗ для текущего опорного решения исходной задачи получают одно из соответствующих решений двойственной задачи, используя соотношения (1).
- Далее, для него осуществляют проверку условий (2).
- Если они выполнены => текущее опорное решение транспортной задачи является оптимальным.
- Иначе осуществляется переход к новому (лучшему) опорному решению, в котором значение целевой функции будет лучше (меньше), чем в предыдущем.

Решение транспортной задачи методом потенциалов

- Нужно уметь:
- Находить опорное решение ТЗ
- Иметь правило перехода к новому опорному решению
- Критерий отсутствия решения не требуется

Решение транспортной задачи
Нахождение начального опорного решения

1) метод северо-западного угла.

В верхнюю левую клетку (северо-западный угол) таблицы записываем наименьшее из чисел b_1 и a_1, пересчитываем запасы и потребности и столбец с исчерпанным запасом или строку с удовлетворенной потребностью исключаем из дальнейшего расчета.

В оставшейся части таблицы снова находим северо-западный угол, заполняем эту клетку, вычеркиваем строку или столбец и опять обращаемся к северо-западному углу и т.д.

Важнейшим условием построения опорного плана является назначение в выбранной клетке наибольшей возможной перевозки.

Решение транспортной задачи
Нахождение начального опорного решения

1) метод северо-западного угла.

	1	2	3	запасы
1	2	8	9	60
2	3	5	8	70
3	4	1	4	120
4	2	4	7	130
5	4	1	2	100
спрос	140	180	160	

Проверим, является ли задача закрытой

Решение транспортной задачи
Нахождение начального опорного решения

1) метод северо-западного угла.

	1	2	3	запасы
1	2	8	9	60
2	3	5	8	70
3	4	1	4	120
4	2	4	7	130
5	4	1	2	100
спрос	140	180	160	480 / 480

Решение транспортной задачи

Нахождение начального опорного решения

1) метод северо-западного угла.

	1	2	3	запасы
1	2^{60}	8	9	60
2	3^{70}	5	8	70
3	4^{10}	1^{110}	4	120
4	2	4^{70}	7^{60}	130
5	4	1	2^{100}	100
спрос	140	180	160	480 / 480

Решение транспортной задачи
Нахождение начального опорного решения

1) метод северо-западного угла.

	1	2	3	запасы
1	2^{60}	8	9	60
2	3^{70}	5	8	70
3	4^{10}	1^{110}	4	120
4	2	4^{70}	7^{60}	130
5	4	1	2^{100}	100
спрос	140	180	160	480 / 480

$$F = 1380$$

Решение транспортной задачи
Нахождение начального опорного решения

2) метод минимальных элементов

Клетки ТЗ заполняются по такому же принципу, как в методе сверо-западного угла, но в первую очередь заполняются клетки с минимальной стоимостью поставки

Решение транспортной задачи
Свойства опорного решения ТЗ

Теорема: Число положительных компонентов в опорном плане (число заполненных клеток в таблице) меньше или равно ***m+n-1***.

Доказательство: В процессе построения опорного плана на каждом шаге заполняли одну клетку таблицы. При этом либо потребности, либо запасы в соответствующей строке или столбце становятся равными нулю, (либо оба вместе). При заполнении последней клетки одновременно удовлетворялись спрос потребителя и исчерпывались запасы поставщика => число заполненных клеток максимум ***m+n-1***.

Если в процессе построения плана встретится клетка (кроме последней), после заполнения которой запасы и потребности столбца и строки становятся равными нулю, то число неизвестных будет меньше ***m+n-1***.

Теорема: Если для транспортной задачи выполнены условия $a_i \in N_0$, $b_j \in N_0$, $N_0 = \{0,1,\ldots\}$, то в любом её допустимом базисном решении, базисные переменные принимают значения из N_0.

Решение транспортной задачи методом потенциалов

$$\begin{cases} u_1+v_1=2 \\ u_1+v_2=3 \\ u_1+v_3=4 \\ u_2+v_3=1 \\ u_2+v_4=4 \\ u_3+v_4=7 \\ u_3+v_5=2 \end{cases} \begin{cases} u_1=0 \\ v_1=2 \\ v_2=3 \\ v_3=4 \\ u_2=-3 \\ v_4=7 \\ u_3=0, v_5=2 \end{cases}$$

$$u_i = c_{ij} - v_j$$

$$v_j = c_{ij} - u_i$$

Решение транспортной задачи методом потенциалов

	1	2	3	запасы
1	2^{60}	8	9	$v_1 = 2$
2	3^{70}	5	8	$v_2 = 3$
3	4	1^{110}	4	$v_3 = 4$
4	2	4^{70}	7^{60}	$v_4 = 7$
5	4	1	2^{100}	$v_5 = 2$
спрос	$u_1 = 0$	$u_2 = -3$	$u_3 = 0$	480 480

Решение транспортной задачи методом потенциалов

Проверяем на оптимальность:

$u_i + v_j \le c_{ij} \Rightarrow d_{ij} = c_{ij} - u_i - v_j \ge 0$ => оптимальное

$$d_{12} = 8 - (-3) - 2 = 9 \ge 0$$
$$d_{13} = 9 - 0 - 2 = 7 \ge 0$$
$$d_{22} = 5 - (-3) - 3 = 5 \ge 0$$
$$d_{23} = 8 - 0 - 3 = 5 \ge 0$$
$$d_{33} = 4 - 0 - 4 = 0 \ge 0$$
$$d_{41} = 2 - 0 - 7 = -5 < 0$$
$$d_{51} = 4 - 0 - 2 = 2 \ge 0$$
$$d_{52} = 1 - (-3) - 2 = 2 \ge 0$$

$d_{41} = -5 < 0$ - клетка пересчета

Решение транспортной задачи методом потенциалов

- Путем перераспределения перевозок будем улучшать план.
- Построим цикл пересчета.
- Циклом пересчета в таблице Т3 называется ломаная линия, вершины которой находятся в заполненных клетках, в клетке пересчета она имеет начало и конец, а звенья располагаются вдоль строк и столбцов таблицы.
- Обозначим вершины ломанной, начиная с клетки пересчета знаками +,-.
- Новый план получим следующим образом: в клетку пересчета записывается наименьшая из величин поставок, стоящих в минусовых клетках. Одновременно это число вычитается из величин поставок «-» клеток и прибавляется к величинам поставок «+» клеток.

Решение транспортной задачи методом потенциалов

	1	2	3	запасы
1	2^{60}	8	9	$v_1 = 2$
2	3^{70}	5	8	$v_2 = 3$
3	4^{10-W}	1^{110+W}	4	$v_3 = 4$
4	2^{W}	4^{70-W}	7^{60}	$v_4 = 7$
5	4	1	2^{100}	$v_5 = 2$
спрос	$u_1 = 0$	$u_2 = -3$	$u_3 = 0$	480 / 480

Решение транспортной задачи методом потенциалов

	1	2	3	v_j
1	2^{60}	8	9	
2	3^{70}	5	8	
3	4	1^{120}	4	
4	2^{10}	4^{60}	7^{60}	
5	4	1	2^{100}	
u_i				

Проверяем полученный план на оптимальность:

1) Вычисляем потенциалы

Решение транспортной задачи методом потенциалов

	1	2	3	v_j
1	2^{60}	8	9	2
2	3^{70}	5	8	3
3	4	1^{120}	4	-1
4	2^{10}	4^{60}	7^{60}	2
5	4	1	2^{100}	-3
u_i	0	2	5	

Решение транспортной задачи методом потенциалов

	1	2	3	v_j
1	2^{60}	8	9	2
2	3^{70}	5	8	3
3	4	1^{120}	4	-1
4	2^{10}	4^{60}	7^{60}	2
5	4	1	2^{100}	-3
u_i	0	2	5	

2) Проверяем выполнение условия (2) для незаполненных клеток

$$d_{12} = 8 - 2 - 2 = 4 \geq 0$$
$$d_{13} = 9 - 5 - 2 = 2 \geq 0$$
$$d_{22} = 5 - 2 - 3 = 0 \geq 0$$
$$d_{23} = 8 - 5 - 3 = 0 \geq 0$$
$$d_{31} = 4 - 0 + 1 = 5 \geq 0$$
$$d_{33} = 4 - 5 - (-1) = 0 \geq 0$$
$$d_{51} = 4 - 0 - (-3) = 7 \geq 0$$
$$d_{52} = 1 - 2 - (-3) = 2 \geq 0$$

Условия (2) выполнены, следовательно, текущее опорное решение является оптимальным

$$F^* = 1330$$
$$G^* = 0 \cdot 140 + 2 \cdot 180 + 5 \cdot 160 + 60 \cdot 2 + 70 \cdot 3 - 120 + 2 \cdot 130 - 3 \cdot 100 = 1330$$

Решение транспортной задачи методом потенциалов

- Вырожденный опорный план

Ошибки искали Воронович и Миронова

- 2 слайд c_{ij} - стоимость одной единицы груза из пункта i в j. Нарисовать сетевую постановку задачи
- 5+7 слайд $\sum_{i=1}^{m} x_{ij} \geq b_{ij}, j = \overline{1,n}$(3 случай)
- Слайд 11 не b_jв столбике, b_n
- Слайд 12 после G не надо =. Знак меньше или равно $u_i + v_j \leq c_{ij}$
- 13 слайд $\left(u_i^* + v_j^* - c_{ij}\right) * x_{ij}^* = 0$
- 24 слайд : дописать у 4^{10}

Методы оптимизации

Целочисленное программирование

Д.В. Домашова

Общая постановка задачи целочисленного программирования отличается от общей постановки задачи ЛП лишь наличием дополнительного ограничения: требования целочисленности, в соответствии с которым значения всех или части переменных являются целыми числами.

Постановка задачи ЦЛП

$$f = c_1 x_1 + c_2 x_2 + ... + c_n x_n \rightarrow max$$

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

$$x_j \in N_0, \; j = \overline{1,k}, \; k \leq n$$

k<n – задача частично-целочисленная

k=n – задача полностью целочисленная

Пример

$$F = x_1 + 1{,}5 \cdot x_2 \to \max$$

$$\begin{cases} 2x_1 + 4x_2 \le 17 \\ 10x_1 + 4x_2 \le 45 \end{cases} \quad x_1, x_2 \ge 0, \quad x_1, x_2 \in N \cup \{0\}$$

$$x^* = \left(\tfrac{7}{2}, \tfrac{5}{2}\right) = (3{,}5,\ 2{,}5) \qquad f^* = 7{,}25$$

Подходы к решению ЗЦЛП
Методы отсечений

1. Решается задача ЛП, получающаяся из исходной отбрасыванием требования целочисленности $x = (x_1, ..., x_n)$.

Если найденное решение x^1 - целочисленное, то оно является решением ЗЦЛП.

Если найденное решение x^1 - не целочисленное, то к ограничениям задачи, решаемой на первом этапе, добавляется ограничение вида:

$\sum_{j=1}^{n} a_{m+1,j} \cdot x_j \geq b_{m+1}$, которое:

1) Отсекает точку x^1, т.е.: $\sum_{j=1}^{n} a_{m+1,j} \cdot x_j^1 < b_{m+1}$

2) Сохраняет в допустимом множестве все целочисленные точки допустимого множества исходной задачи. Такое ограничение называется правильным отсечением.

2. На втором этапе находится решение x^2 задачи ЛП с дополнительным ограничением. Если x^2 - не целочисленное, тогда вводится новое правильное отсечение вида $\sum_{j=1}^{n} a_{m+2,j} \cdot x_j \geq b_{m+2}$ и т.д. до тех пор, пока решение очередной задачи ЛП, не окажется целочисленным.

Подходы к решению ЗЦЛП
Комбинаторные методы

- В основе комбинаторных методов лежит идея перебора всех элементов множества допустимых решений, удовлетворяющих требованию целочисленности, с целью нахождения оптимального решения.
  - Такими методами являются методы ветвей и границ.
  - Различные методы типа ветвей и границ существенно используют специфику конкретной задачи и заметно отличаются друг от друга.
  - Все они основаны на последовательном разбиении допустимого множества на подмножества (ветвление) и вычислении оценок (границ), позволяющих отбрасывать подмножества, заведомо не содержащие решений задачи.

Подходы к решению ЗЦЛП
Общая идея методов ветвей и границ

Задача: $\underset{x \in X}{f(x) \to \min}$

1) В зависимости от специфики задачи выбирается некоторый способ вычисления оценок снизу $d(X^{'})$, функции $f(x)$ на множествах $X^{'} \subset X$:
(в частности может быть $X^{'} = X$)
$f(x) \geq d(X^{'})$, $x \in X^{'}$.
Оценка снизу часто вычисляется путем релаксации, т.е. замены задачи минимизации $f(x)$ по множеству $X^{'}$ задачей минимизации по некоторому более широкому множеству. (Например, релаксацией целочисленного или частично целочисленной задачи может состоять в отбрасывании требования целочисленности.)

2) Выбирается также правило ветвления, состоящее в выборе разветвляемого подмножества $X^{'}$ из числа подмножеств, на которые к данному шагу разбито множество X, и выборе способа разбиения $X^{'}$ на непересекающиеся подмножества.
Обычно из числа кандидатов на ветвление выбирается множество $X^{'}$ с наименьшей оценкой, поскольку именно в таком множестве естественно искать минимум в первую очередь.
При этом рассматриваются только такие способы вычисления оценок снизу, в которых оценки для подмножеств, получившихся в результате разветвления $X^{'}$ не меньше $d(X^{'})$.

Метод отсечений Гомори
1. Полностью целочисленная задача

Рассмотрим полностью целочисленную задачу, представленную в канонической форме:

$$F = \sum_{j=1}^{n} c_j x_j \to \max$$

$$\sum_{j=1}^{n} a_{ij} x_j = b_j, i = \overline{1, m}$$

$$x_j \geq 0, \ j = \overline{1, n}, x_j \in N_0, \ j = \overline{1, n}$$

Будем считать, что c_j, a_{ij}, b_j Є Z.

Метод отсечений Гомори
1. Полностью целочисленная задача

Иначе строим правильное отсечение.

Для этого выбираем любое базисное x_i^*, которому соответствует нецелое значение, и выписываем *i*-ую строку оптимальной симплекс-таблицы.

$$x_i + \sum_{j \in S} a_{ij}^* x_j = b_i^* \qquad (1)$$

где *S* – множество индексов свободных переменных.

Полагая, что в (1) все переменные целочисленные, получаем:

$\sum_{j \in S} a_{ij}^* x_j - b_i^* = a \in Z$ для любых $d_j \in Z$ существует $d \in Z$:

$$\sum_{j \in S} (a_{ij}^* - d_j) x_j - (b_i^* - d_i) = d$$

при этом, если считать $d_j = [a_{ij}^*], j \in S, d_i = [b_i^*]$ получим:

$$\sum_{j \in S} \gamma_{ij} x_j = d + \gamma_i, d \in Z \qquad (2)$$

где $\gamma_{ij} = \{a_{ij}^*\}, \gamma_i = \{b_i^*\}$

Метод отсечений Гомори
1. Полностью целочисленная задача

$$\sum_{j\in S}\{a_{ij}^*\}x_j = d + \{b_i^*\}, d \in Z$$

Т.к. левая часть равенства (2) является неотрицательной ($\gamma_{ij} \geq 0,\ x_j \geq 0$), то d Є N_0 и

отсечение Гомори:

$$\sum_{j\in S}\{a_{ij}^*\}x_j \geq \{b_i^*\}$$

Метод отсечений Гомори
2. Частично-целочисленная задача

Из (1) следует, что

$$x_i + \sum_{j \in S} a_{ij}^* x_j = [b_i^*] + \{b_i^*\} \quad (3)$$

$$x_i - [b_i^*] = \{b_i^*\} - \sum_{j \in S} a_{ij}^* x_j$$

при этом для переменной x_i, удовлетворяющей требованию Є N_0 выполняется одно из условий:

а) $x_i \le [b_i^*]$ б) $x_i \ge [b_i^*] + 1$

Согласно (3) эти условия могут быть представлены в следующем виде:

$$\sum_{j \in S} a_{ij}^* x_j \ge \{b_i^*\} \quad (4) \qquad \sum_{j \in S} a_{ij}^* x_j \le \{b_i^*\} - 1 \quad (5)$$

Метод отсечений Гомори
2. Частично-целочисленная задача

Пусть S^{+} - множество значений j: $a_{ij}^{*} \geq 0$

S^{-} - множество значений j: $a_{ij}^{*} < 0 \qquad S = S^{+} \cup S^{-}$

Из (4) следует, что $\sum\limits_{j \in S^{+}} a_{ij}^{*} x_j \geq \{b_i^{*}\}$ (6)

а (5) может быть преобразовано к следующему неравенству:

$$\sum_{j \in S^{-}} a_{ij}^{*} x_j \leq \{b_i^{*}\} - 1 \quad \text{или} \quad \frac{\{b_i^{*}\}}{\{b_i^{*}\} - 1} \sum_{j \in S^{-}} a_{ij}^{*} x_j \geq \{b_i^{*}\} \qquad (7)$$

Метод отсечений Гомори

2. Частично-целочисленная задача

Неравенства (4), (5) и следствия из них (6), (7) не могут выполняться одновременно. Но независимо от того, какой случай имеет место, для каждого допустимого решения будет выполняться ограничение:

$$\sum_{j\in S^+} a_{ij}^* x_j + \frac{\{b_i^*\}}{\{b_i^*\}-1}\sum_{j\in S^-} a_{ij}^* x_j \ge \{b_i^*\} \qquad (8)$$

Неравенство (8) определяет новое дополнительное ограничение.
Это ограничение получено без учета требования целочисленности для некоторых переменных модели.

$$\sum_{j\in S^+}(a_{ij}^* - [a_{ij}^*])x_j + \frac{\{b_i^*\}}{1-\{b_i^*\}}\sum_{j\in S^-}(1-\{a_{ij}^*\})x_j \ge \{b_i^*\} - [a_{ij}^*]$$

Метод отсечений Гомори

2. Частично-целочисленная задача

$$\sum_{j\in S^+} a_{ij}^* x_j + \frac{\{b_i^*\}}{\{b_i^*\}-1}\sum_{j\in S^-} a_{ij}^* x_j \geq \{b_i^*\} \quad (8)$$

(8) можно записать в следующем виде:

$$\sum_{j\in S} \gamma_{ij} x_j \geq \{b_i^*\}$$

для переменных, которые могут быть нецелыми:

$$\gamma_{ij} = \begin{cases} \dfrac{\{b_i^*\}}{\{b_i^*\}-1} a_{ij}^*, \text{ если } a_{ij}^* < 0 \\ a_{ij}^*, \text{ если } a_{ij}^* \geq 0 \end{cases},$$

для переменных, которые могут быть только целыми:

$$\gamma_{ij} = \begin{cases} \{a_{ij}^*\}, \text{ если } \{a_{ij}^*\} \leq \{b_i^*\} \\ \dfrac{1-\{a_{ij}^*\}}{1-\{b_i^*\}}\{b_i^*\}, \text{ если } \{a_{ij}^*\} > \{b_i^*\} \end{cases}$$

Метод отсечений Гомори
Пример

$$f(x_1, x_2) = x_1 + x_2 \to \max$$

$$\begin{cases} x_1 \le 2{,}5 \\ x_2 \le 2{,}5 \end{cases}$$

$$x_1, x_2 \in N_0$$

Метод отсечений Гомори
Пример

$$f(x_1, x_2) = x_1 + x_2 \to \max$$

$$\begin{cases} x_1 \le 2{,}5 \\ x_2 \le 2{,}5 \end{cases}$$

$$x_1, x_2 \in N_0$$

$$x_{max} = (2{,}5; 2{,}5)$$

Метод отсечений Гомори
Пример

$$f(x_1, x_2) = x_1 + x_2 \to \max$$

$$\begin{cases} x_1 \le 2{,}5 \\ x_2 \le 2{,}5 \end{cases}$$

$$x_1, x_2 \in N_0$$

$$x_{max} = (2{,}5; 2{,}5)$$

$$f(x_1, x_2) = x_1 + x_2 \to \max$$

$$\begin{cases} x_1 + x_3 = 2{,}5 \\ x_2 + x_4 = 2{,}5 \end{cases}$$

$$x_1, x_2 \in N_0, x_3, x_4 \ge 0$$

баз	Сб	B	1	1	0	0
			A_1	A_2	A_3	A_4
A_1	1	2,5	1	0	1	0
A_2	1	2,5	0	1	0	1
			0	0	-1	-1

Метод отсечений Гомори
Пример

$$f(x_1, x_2) = x_1 + x_2 \to \max$$
$$\begin{cases} x_1 \le 2{,}5 \\ x_2 \le 2{,}5 \end{cases}$$
$$x_1, x_2 \in N_0$$

$$f(x_1, x_2) = x_1 + x_2 \to \max$$
$$\begin{cases} x_1 + x_3 = 2{,}5 \\ x_2 + x_4 = 2{,}5 \end{cases}$$
$$x_1, x_2 \in N_0, x_3, x_4 \ge 0$$

$$x_{max} = (2{,}5; 2{,}5)$$

баз	Сб	B	1	1	0	0
			A_1	A_2	A_3	A_4
A_1	1	2,5	1	0	1	0
A_2	1	2,5	0	1	0	1
			0	0	-1	-1

$$\{1\} \cdot x_1 + \{0\} \cdot x_2 + 1 \cdot x_3 + 0 \cdot x_4 \ge \{2{,}5\}$$
$$x_3 \ge 0{,}5$$

Метод отсечений Гомори
Пример

$$f(x_1, x_2) = x_1 + x_2 \to \max$$

$$\begin{cases} x_1 \le 2{,}5 \\ x_2 \le 2{,}5 \end{cases}$$

$$x_1, x_2 \in N_0$$

$$x_{max} = (2{,}5; 2{,}5)$$

$$f(x_1, x_2) = x_1 + x_2 \to \max$$

$$\begin{cases} x_1 + x_3 = 2{,}5 \\ x_2 + x_4 = 2{,}5 \end{cases}$$

$$x_1, x_2 \in N_0, x_3, x_4 \ge 0$$

баз	Сб	B	1	1	0	0
			A_1	A_2	A_3	A_4
A_1	1	2,5	1	0	1	0
A_2	1	2,5	0	1	0	1
			0	0	-1	-1

$$\{1\} \cdot x_1 + \{0\} \cdot x_2 + 1 \cdot x_3 + 0 \cdot x_4 \ge \{2{,}5\}$$

$$x_3 \ge 0{,}5$$

$$x_3 = 2{,}5 - x_1$$

$$2{,}5 - x_1 \ge 0{,}5$$

$$x_1 \le 2$$

Метод отсечений Гомори
Пример

$$f(x_1, x_2) = x_1 + x_2 \to \max$$

$$\begin{cases} x_1 + x_3 = 2 \\ x_2 + x_4 = 2{,}5 \end{cases}$$

$$x_1, x_2 \in N_0, x_3, x_4 \geq 0$$

баз	Сб	B	1	1	0	0
			A_1	A_2	A_3	A_4
A_1	1	2	1	0	1	0
A_2	1	2,5	0	1	0	1
			0	0	-1	-1

$x_{max} = (2; 2{,}5)$

Метод отсечений Гомори
Пример

$$f(x_1, x_2) = x_1 + x_2 \rightarrow \max$$

$$\begin{cases} x_1 + x_3 = 2 \\ x_2 + x_4 = 2{,}5 \end{cases}$$

$$x_1, x_2 \in N_0, x_3, x_4 \geq 0$$

баз	Сб	B	1	1	0	0
			A_1	A_2	A_3	A_4
A_1	1	2	1	0	1	0
A_2	1	2,5	0	1	0	1
			0	0	-1	-1

$$x_{max} = (2; 2{,}5)$$

$$\{0\} \cdot x_1 + \{1\} \cdot x_2 + 0 \cdot x_3 + 1 \cdot x_4 \geq \{2{,}5\}$$

$$x_4 \geq 0{,}5$$

$$x_4 = 2{,}5 - x_2$$

$$2{,}5 - x_2 \geq 0{,}5$$

$$x_2 \leq 2$$

Метод отсечений Гомори
Пример

$$f(x_1, x_2) = x_1 + x_2 \to \max$$

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

$$x_1, x_2 \in N_0, x_3, x_4 \geq 0$$

баз	Сб	B	1	1	0	0
			A_1	A_2	A_3	A_4
A_1	1	2	1	0	1	0
A_2	1	2	0	1	0	1
			0	0	-1	-1

$x_{max} = (2; 2)$

Ошибки в данной презентации искали Котовану и Павлов.

Ошибки:
Слайд 3 : добавить $x_j \geq 0, j = \overline{1, n}$

Методы оптимизации

Целочисленное программирование

Метод ветвей и границ решения задачи коммивояжера

Д.В. Домашова

Общая идея методов ветвей и границ

Задача: $f(x) \to \min\limits_{x \in X}$

1) В зависимости от специфики задачи выбирается некоторый способ вычисления оценок снизу $d(X')$, функции $f(x)$ на множествах $X' \subset X$: (в частности может быть $X' = X$)

$f(x) \geq d(X'),\ x \in X'$.

Оценка снизу часто вычисляется путем релаксации, т.е. замены задачи минимизации $f(x)$ по множеству X' задачей минимизации по некоторому более широкому множеству. (Например, релаксацией целочисленного или частично целочисленной задачи может состоять в отбрасывании требования целочисленности.)

2) Выбирается также правило ветвления, состоящее в выборе разветвляемого подмножества X' из числа подмножеств, на которые к данному шагу разбито множество X, и выборе способа разбиения X' на непересекающиеся подмножества.

Обычно из числа кандидатов на ветвление выбирается множество X' с наименьшей оценкой, поскольку именно в таком множестве естественно искать минимум в первую очередь.

При этом рассматриваются только такие способы вычисления оценок снизу, в которых оценки для подмножеств, получившихся в результате разветвления X' не меньше $d(X')$.

Метод ветвей и границ решения задачи коммивояжера
Постановка задачи

Постановка задачи:
n – городов

c_{ij} - стоимость (расстояние) переезда из i в j

$j,i=1,2,\ldots n;\ j\neq i$

$$C=\begin{pmatrix} c_{11} & \cdots & c_{1n} \\ \cdots & \cdots & \cdots \\ c_{n1} & \cdots & c_{nn} \end{pmatrix}$$

Выезжая из одного, коммивояжер должен объехать все и вернуться в исходный город. В каждый город можно заезжать только один раз.

Нужно: Найти замкнутый маршрут объезда всех городов минимальной стоимости.

Метод ветвей и границ решения задачи коммивояжера
Постановка задачи

Введем переменные:

$$x_{ij} = \begin{cases} 1, & \text{если коммивояжер едет из } i \text{ в } j \\ 0, & \text{если коммивояжер не едет из } i \text{ в } j \end{cases}$$

$j, i = 1, 2, \ldots n;\ j \neq i$

Метод ветвей и границ решения задачи коммивояжера
Постановка задачи

Введем переменные:

$$x_{ij} = \begin{cases} 1, & \text{если коммивояжер едет из } i \text{ в } j \\ 0, & \text{если коммивояжер не едет из } i \text{ в } j \end{cases}$$

$j,i=1,2,\ldots n;\ j\neq i$

Математическая модель

$$F = \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}x_{ij} \to \min$$

$\sum_{i=1}^{n} x_{ij} = 1$ - в каждый *j*-ый город приезжает один раз, $j = \overline{1,n}$.

$\sum_{j=1}^{n} x_{ij} = 1$ - из каждого *i*-го города выезжает ровно один раз, $i = \overline{1,n}$.

$u_i - u_j + nx_{ij} \le n-1$ - обеспечивает замкнутость маршрута и отсутствие петель

$u_i \in \{1,2,\ldots\}$, $x_{ij} \in \{0,1\}$, $i, j = \overline{1,n}$

Метод ветвей и границ решения задачи коммивояжера
Постановка задачи

1) Покажем, что любой набор переменных, u_i, x_{ij}, удовлетворяющий ограничениям этой ЦЛП, определяет допустимый маршрут коммивояжеров.
Пусть некоторый допустимый набор переменных определяет маршрут, распадающийся на не связанные между собой подциклы. Возьмем любой подцикл. Сложим ограничения неравенства, соответствующие переездам, входящим в этот подцикл.
Например: 1-2-3-1

$$u_1 - u_2 + n \le n - 1$$

$$u_2 - u_3 + n \le n - 1$$

$$u_3 - u_1 + n \le n - 1$$

Складываем уравнения => $3n \le 3(n-1)$ - неверное неравенство.

Итак, разности $u_i - u_j$ взаимно уничтожаются, и получается неверное неравенство: $kn \le k(n-1)$, где k – число переездов.

2) Покажем, что произвольному дополнительному маршруту коммивояжера соответствует некоторый, набор переменных, удовлетворяющих ограничению задачи. Положим $u_i = p$, если коммивояжер на маршруте прибывает в i-ый город после p-го переезда.

=> $u_i - u_j \le n - 1 \ \forall i, j$, при $x_{ij} = 0$

$$(p - (p+1) \le n - 1)$$

$u_i - u_j + nx_{ij} = p - (p+1) + n = n - 1$ => Задача ЛП – правильная.

Метод ветвей и границ решения задачи коммивояжера Определения

Определение: Циклом t назовем набор из n упорядоченных пар городов, образующих маршрут проходящий через каждый город только один раз. $t = \{(i_1, i_2), (i_2, i_3), \ldots, (i_{n-1}, i_n), (i_n, i_1)\}$

Издержки цикла $z(t) = \sum_{k=1}^{n-1} c_{ik,ik+1} + c_{in,i1}$

Для каждого допустимого маршрута каждая строка и каждый столбец содержат ровно по одному элементу, соответствующему этому маршруту.

$$C = \begin{pmatrix} \infty & \cdots & c_{1n} \\ \cdots & \cdots & \cdots \\ c_{n1} & \cdots & \infty \end{pmatrix}$$

Метод ветвей и границ решения задачи коммивояжера Алгоритм

1. Определение нижних границ.

Для определения нижних границ воспользуемся понятием приведения матрицы.

Определение: Матрица, получаемая из данной вычитанием из элементов каждой строки минимального элемента этой строки, а затем вычитанием из элементов каждого столбца минимального элемента этого столбца называется приведенной матрицей:

1) $\| c_{ij}^{'} \| = \| c_{ij} - \min c_{ij} \|$

2) $\| c_{ij}^{''} \| = \| c_{ij} - \min\limits_{i=\overline{1,n}} c_{ij}^{'} \|$

Определение: Приводящая константа:

$$h = \sum_{i=1}^{n} \min_{j=\overline{1,n}} c_{ij} + \sum_{j=1}^{n} \min_{i=\overline{1,n}} c_{ij}^{'}$$

Метод ветвей и границ решения задачи коммивояжера
Алгоритм

Пусть t – некоторый маршрут.

$z(t)$ - его издержки по исходной матрице

$z^{'}(t)$ - его издержки по приведенной матрице

Метод ветвей и границ решения задачи коммивояжера Алгоритм

Пусть t – некоторый маршрут.

$z(t)$ - его издержки по исходной матрице

$z^{'}(t)$ - его издержки по приведенной матрице

=> $z(t) = z^{'}(t) + h$

=> h является нижней границей издержек для всех циклов t исходной матрицы расстояний, поскольку h - сумма минимальных элементов строк и столбцов.

Метод ветвей и границ решения задачи коммивояжера Алгоритм

2. Ветвление

На каждом шаге алгоритма будет строиться одно звено оптимального маршрута. Для построения решения в начале целесообразно выбрать звено нулевой длины, а затем последовательно добавлять звенья нулевой или минимальной длины.

Процесс ветвления можно наглядно представить в виде дерева, каждая вершина которого соответствует некоторому множеству маршрутов, являющегося подмножеством множества X.

Метод ветвей и границ решения задачи коммивояжера Алгоритм

Пусть G_0 - множество всех маршрутов. Разобьем G_0 на два подмножества:

- Множество маршрутов, включающих переезд из *i* в *j* $\{(i, j)\}$
- Множество маршрутов, не содержащих пару *i,j* $\{\overline{(i, j)}\}$

Пару городов *(i,j)* для ветвления будем выбирать среди тех пар, которым в приведенной матрице соответствуют нулевые элементы, причем выбирается такая пара *(i,j)* чтобы подмножество $\{\overline{(i, j)}\}$ имело максимальную оценку.

Разветвляя далее множество с меньшей оценкой, в конце концов будет получено подмножество, содержащее один маршрут. Двигаясь по дереву в обратном направлении, получим маршрут:

Метод ветвей и границ решения задачи коммивояжера Алгоритм

Выбор пары городов *{i,j}* для ветвления:

1) c''_{ij} =0

2) оценка множества $\{\overline{i,\ j}\}$ должна быть максимальной.

Рассмотрим маршруты, которые будут включены в $\{\overline{i,\ j}\}$. Поскольку город i должен быть связан с некоторым другим городом, то каждый маршрут из $\{\overline{i,\ j}\}$ должен содержать звено, длина которого не меньше минимального элемента i-ой строки, не считая c''_{ij}.

Вычисляем сумму этих минимальных элементов для каждого c''_{ij} =0. Назовем эту сумму оценкой пары *(i,j)* или штрафом за использование звена *(i,j)*:

$$\Theta_{ij} = \min_{\forall j' \neq j} c''_{ij'} + \min_{\forall i' = i} c''_{i'j}$$

=> в качестве пары городов для ветвления выбираем ту *(i,j)*, для которой Θ_{ij} является максимальной.

=> $$\left\{ \begin{array}{l} c''_{ij} = 0 \\ \Theta_{ij} = \max \Theta_{ij} \end{array} \right\}$$

Метод ветвей и границ решения задачи коммивояжера
Алгоритм

3. Вычисление нижней границы множества $\left\{\overline{(i, j)}\right\}$.

Нижняя граница множества $\left\{\overline{(i, j)}\right\}$ определяется как сумма оценки разветвляемого множества и максимального значения Θ_{ij} .

Метод ветвей и границ решения задачи коммивояжера
Алгоритм

3. Вычисление нижней границы множества $\left\{\overline{(i,\, j)}\right\}$.

Нижняя граница множества $\left\{\overline{(i,\, j)}\right\}$ определяется как сумма оценки разветвляемого множества и максимального значения Θ_{ij} .

4. Так как из каждого города можно выезжать только один раз и в каждый город можно въезжать только один раз, то строку *i* и столбец *j* из дальнейшего рассмотрения исключаем. Чтобы не получить замкнутых неполных циклов, нужно наложить необходимые запреты, в частности, на переезд из *j* в *i*, т.е. положить $c_{ij} = \infty$.

Метод ветвей и границ решения задачи коммивояжера Алгоритм

3. Вычисление нижней границы множества $\{\overline{(i,\,j)}\}$.

Нижняя граница множества $\{\overline{(i,\,j)}\}$ определяется как сумма оценки разветвляемого множества и максимального значения Θ_{ij} .

4. Так как из каждого города можно выезжать только один раз и в каждый город можно въезжать только один раз, то строку *i* и столбец *j* из дальнейшего рассмотрения исключаем. Чтобы не получить замкнутых неполных циклов, нужно наложить необходимые запреты, в частности, на переезд из *j* в *i*, т.е. положить $c_{ij} = \infty$.

5. Если усеченная матрица ещё не имеет размерности 2х2, то приводим полученную матрицу и находим оценку множества $\{(i,\,j)\}$ как сумму оценки разветвляемого множества и полученной приводящей константы, переход пункт 2.

Метод ветвей и границ решения задачи коммивояжера Алгоритм

3. Вычисление нижней границы множества $\left\{\overline{(i,\, j)}\right\}$.

Нижняя граница множества $\left\{\overline{(i,\, j)}\right\}$ определяется как сумма оценки разветвляемого множества и максимального значения Θ_{ij} .

4. Так как из каждого города можно выезжать только один раз и в каждый город можно въезжать только один раз, то строку *i* и столбец *j* из дальнейшего рассмотрения исключаем. Чтобы не получить замкнутых неполных циклов, нужно наложить необходимые запреты, в частности, на переезд из *j* в *i*, т.е. положить $c_{ij} = \infty$.

5. Если усеченная матрица ещё не имеет размерности 2х2, то приводим полученную матрицу и находим оценку множества $\{(i,\, j)\}$ как сумму оценки разветвляемого множества и полученной приводящей константы, переход пункт 2.

6. Если полученная матрица имеет размерность 2х2, то определяемые ею пары городов завершают маршрут.
Приводя эту матрицу и добавляя приводящую константу к оценке последнего разветвляемого множества, получим оценку маршрута.

Метод ветвей и границ решения задачи коммивояжера Алгоритм

3. Вычисление нижней границы множества $\left\{\overline{(i, j)}\right\}$.

Нижняя граница множества $\left\{\overline{(i, j)}\right\}$ определяется как сумма оценки разветвляемого множества и максимального значения Θ_{ij} .

4. Так как из каждого города можно выезжать только один раз и в каждый город можно въезжать только один раз, то строку *i* и столбец *j* из дальнейшего рассмотрения исключаем. Чтобы не получить замкнутых неполных циклов, нужно наложить необходимые запреты, в частности, на переезд из *j* в *i*, т.е. положить $c_{ij} = \infty$.

5. Если усеченная матрица ещё не имеет размерности 2х2, то приводим полученную матрицу и находим оценку множества $\{(i, j)\}$ как сумму оценки разветвляемого множества и полученной приводящей константы, переход пункт 2.

6. Если полученная матрица имеет размерность 2х2, то определяемые ею пары городов завершают маршрут.

Приводя эту матрицу и добавляя приводящую константу к оценке последнего разветвляемого множества, получим оценку маршрута.

Критерий оптимальности

Если эта оценка не больше оценок всех тупиковых ветвей, то маршрут описанный деревом ветвей, является оптимальным, иначе процесс ветвления должен быть продолжен, исходя из множества с меньшей оценкой.

Метод ветвей и границ решения задачи коммивояжера Пример

Пример. Имеется четыре пункта, расстояние между которыми описано матрицей расстояний. Найти оптимальный (минимальный) замкнутый маршрут объезда городов.

$$S = \begin{array}{c c} & \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \hline \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array} & \begin{pmatrix} \infty & 13 & 12 & 4 \\ 13 & \infty & 7 & 8 \\ 12 & 7 & \infty & 5 \\ 4 & 8 & 5 & \infty \end{pmatrix} \end{array}$$

Метод ветвей и границ решения задачи коммивояжера Пример

1 Приводим матрицу: ***S***.

$$
\begin{array}{cc}
 & \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\
\begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array} &
\begin{pmatrix}
\infty & 13 & 12 & 4 \\
13 & \infty & 7 & 8 \\
12 & 7 & \infty & 5 \\
4 & 8 & 5 & \infty
\end{pmatrix}
\begin{array}{c} 4 \\ 7 \\ 5 \\ 4 \end{array}
\end{array}
$$

Метод ветвей и границ решения задачи коммивояжера
Пример

1 Приводим матрицу: ***S***.

$$\begin{array}{c|c} & \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array} & \begin{pmatrix} \infty & 13 & 12 & 4 \\ 13 & \infty & 7 & 8 \\ 12 & 7 & \infty & 5 \\ 4 & 8 & 5 & \infty \end{pmatrix} \begin{array}{c} 4 \\ 7 \\ 5 \\ 4 \end{array} \end{array} \rightarrow \begin{array}{c|c} & \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array} & \begin{pmatrix} \infty & 9 & 8 & 0 \\ 6 & \infty & 0 & 1 \\ 7 & 2 & \infty & 0 \\ 0 & 4 & 1 & \infty \end{pmatrix} \\ & \begin{array}{cccc} 0 & 2 & 0 & 0 \end{array} \end{array}$$

Метод ветвей и границ решения задачи коммивояжера Пример

1 Приводим матрицу: ***S***.

$$\begin{array}{cc} & \begin{array}{cccc}1 & 2 & 3 & 4\end{array} \\ \begin{array}{c}1\\2\\3\\4\end{array} & \begin{pmatrix} \infty & 13 & 12 & 4 \\ 13 & \infty & 7 & 8 \\ 12 & 7 & \infty & 5 \\ 4 & 8 & 5 & \infty \end{pmatrix} \end{array} \begin{array}{c}4\\7\\5\\4\end{array} \rightarrow \begin{array}{cc} & \begin{array}{cccc}1 & 2 & 3 & 4\end{array} \\ \begin{array}{c}1\\2\\3\\4\end{array} & \begin{pmatrix} \infty & 9 & 8 & 0 \\ 6 & \infty & 0 & 1 \\ 7 & 2 & \infty & 0 \\ 0 & 4 & 1 & \infty \end{pmatrix} \\ & \begin{array}{cccc}0 & 2 & 0 & 0\end{array} \end{array} \rightarrow \begin{array}{cc} & \begin{array}{cccc}1 & 2 & 3 & 4\end{array} \\ \begin{array}{c}1\\2\\3\\4\end{array} & \begin{pmatrix} \infty & 7 & 8 & 0 \\ 6 & \infty & 0 & 1 \\ 7 & 0 & \infty & 0 \\ 0 & 2 & 1 & \infty \end{pmatrix} \end{array}$$

2. Найдем сумму приводящих констант: h=4+7+5+4+2=22

Найдем оценку множеств G_0: $w(G_0)=22$.

Метод ветвей и границ решения задачи коммивояжера Пример

1 Приводим матрицу: ***S***.

$$
\begin{array}{c} \\ 1\\2\\3\\4 \end{array}
\begin{array}{c} \begin{array}{cccc}1&2&3&4\end{array} \\ \begin{pmatrix} \infty & 13 & 12 & 4 \\ 13 & \infty & 7 & 8 \\ 12 & 7 & \infty & 5 \\ 4 & 8 & 5 & \infty \end{pmatrix} \end{array}
\begin{array}{c} \\ 4\\7\\5\\4 \end{array}
\to
\begin{array}{c} \\ 1\\2\\3\\4 \\ \\ \end{array}
\begin{array}{c} \begin{array}{cccc}1&2&3&4\end{array} \\ \begin{pmatrix} \infty & 9 & 8 & 0 \\ 6 & \infty & 0 & 1 \\ 7 & 2 & \infty & 0 \\ 0 & 4 & 1 & \infty \end{pmatrix} \\ \begin{array}{cccc}0&2&0&0\end{array} \end{array}
\to
\begin{array}{c} \\ 1\\2\\3\\4 \end{array}
\begin{array}{c} \begin{array}{cccc}1&2&3&4\end{array} \\ \begin{pmatrix} \infty & 7 & 8 & 0 \\ 6 & \infty & 0 & 1 \\ 7 & 0 & \infty & 0 \\ 0 & 2 & 1 & \infty \end{pmatrix} \end{array}
$$

2. Найдем сумму приводящих констант: h=4+7+5+4+2=22

Найдем оценку множеств G_0: $w(G_0)$=22.

Метод ветвей и границ решения задачи коммивояжера
Пример

3. Укажем претендентов на ветвление и осуществим выбор пары для ветвления:

$S_{14}=0$, $S_{23}=0$, $S_{32}=0$, $S_{34}=0$, $S_{41}=0$

$Q_{14}=7+0=7$, $Q_{23}=1+1=2$, $Q_{32}=0+2=2$, $Q_{34}=0+0=0$, $Q_{41}=6+1=7$

$\max Q_{ij}=Q_{14}=7$ (можно $Q_{41}=7$).

$$
\begin{array}{c c c}
 & \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} & \\
\begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array} & \begin{pmatrix} \infty & 7 & 8 & 0 \\ 6 & \infty & 0 & 1 \\ 7 & 0 & \infty & 0 \\ 0 & 2 & 1 & \infty \end{pmatrix} & \begin{array}{c} 7 \\ 1 \\ 0 \\ 1 \end{array} \\
 & \begin{array}{cccc} 6 & 2 & 1 & 0 \end{array} &
\end{array}
$$

Итак, для ветвления выбираем пару (1,4) и начинаем строить дерево ветвей.

Метод ветвей и границ решения задачи коммивояжера Пример

4. Найдем оценку множества {1,4} – множества всех маршрутов не содержащих пару (1,4): $w(\{1,4\})=w(G_0)+Q(1,4)=22+7=29$.

5. Вычеркнем строку «1», столбец «4» и наложим запрет на переезд из «4» в «1»

$$\begin{array}{c c} & \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \begin{array}{c} 2 \\ 3 \\ 4 \end{array} & \begin{pmatrix} 6 & \infty & 0 \\ 7 & 0 & \infty \\ \infty & 2 & 1 \end{pmatrix} \end{array}$$

Метод ветвей и границ решения задачи коммивояжера Пример

4. Найдем оценку множества $\{1,4\}$ – множества всех маршрутов не содержащих пару (1,4): $w(\{1,4\})=w(G_0)+Q(1,4)=22+7=29$.

5. Вычеркнем строку «1», столбец «4» и наложим запрет на переезд из «4» в «1»

$$\begin{array}{cc} & \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \begin{array}{c} 2 \\ 3 \\ 4 \end{array} & \begin{pmatrix} 6 & \infty & 0 \\ 7 & 0 & \infty \\ \infty & 2 & 1 \end{pmatrix} \end{array}$$

Метод ветвей и границ решения задачи коммивояжера Пример

6. Приведем матрицу

$$\begin{array}{cc} & \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \begin{array}{c} 2 \\ 3 \\ 4 \end{array} & \begin{pmatrix} 6 & \infty & 0 \\ 7 & 0 & \infty \\ \infty & 2 & 1 \end{pmatrix} \end{array}$$

Метод ветвей и границ решения задачи коммивояжера Пример

6. Приведем матрицу

$$
\begin{array}{c c c}
 & \begin{array}{ccc} 1 & 2 & 3 \end{array} & \\
\begin{array}{c} 2 \\ 3 \\ 4 \end{array} & \begin{pmatrix} 6 & \infty & 0 \\ 7 & 0 & \infty \\ \infty & 2 & 1 \end{pmatrix} & \begin{array}{c} 0 \\ 0 \\ 1 \end{array}
\end{array}
$$

Метод ветвей и границ решения задачи коммивояжера Пример

6. Приведем матрицу

$$\begin{array}{c} \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \begin{pmatrix} 6 & \infty & 0 \\ 7 & 0 & \infty \\ \infty & 2 & 1 \end{pmatrix} \end{array} \begin{array}{c} \\ 0 \\ 0 \\ 1 \end{array} \rightarrow \begin{array}{c} \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \begin{pmatrix} 6 & \infty & 0 \\ 7 & 0 & \infty \\ \infty & 1 & 0 \end{pmatrix} \end{array}$$

Метод ветвей и границ решения задачи коммивояжера
Пример

6. Приведем матрицу

$$\begin{array}{c|ccc|c} & 1 & 2 & 3 & \\ \hline 2 & 6 & \infty & 0 & 0 \\ 3 & 7 & 0 & \infty & 0 \\ 4 & \infty & 2 & 1 & 1 \end{array} \rightarrow \begin{array}{c|ccc|} & 1 & 2 & 3 \\ \hline 2 & 6 & \infty & 0 \\ 3 & 7 & 0 & \infty \\ 4 & \infty & 1 & 0 \\ \hline & 6 & 0 & 0 \end{array}$$

Метод ветвей и границ решения задачи коммивояжера Пример

6. Приведем матрицу

$$\begin{array}{cc} & \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \begin{array}{c} 2 \\ 3 \\ 4 \end{array} & \begin{pmatrix} 6 & \infty & 0 \\ 7 & 0 & \infty \\ \infty & 2 & 1 \end{pmatrix} \end{array} \begin{array}{c} 0 \\ 0 \\ 1 \end{array} \rightarrow \begin{array}{cc} & \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \begin{array}{c} 2 \\ 3 \\ 4 \end{array} & \begin{pmatrix} 6 & \infty & 0 \\ 7 & 0 & \infty \\ \infty & 1 & 0 \end{pmatrix} \\ & \begin{array}{ccc} 6 & 0 & 0 \end{array} \end{array} \rightarrow \begin{array}{cc} & \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \begin{array}{c} 2 \\ 3 \\ 4 \end{array} & \begin{pmatrix} 0 & \infty & 0 \\ 1 & 0 & \infty \\ \infty & 1 & 0 \end{pmatrix} \end{array}$$

Метод ветвей и границ решения задачи коммивояжера Пример

6. Приведем матрицу

$$\begin{array}{c|c} & \begin{array}{ccc}1 & 2 & 3\end{array} \\ \begin{array}{c}2\\3\\4\end{array} & \begin{pmatrix} 6 & \infty & 0 \\ 7 & 0 & \infty \\ \infty & 2 & 1 \end{pmatrix} \end{array} \begin{array}{c}0\\0\\1\end{array} \rightarrow \begin{array}{cc} & \begin{array}{ccc}1 & 2 & 3\end{array} \\ \begin{array}{c}2\\3\\4\end{array} & \begin{pmatrix} 6 & \infty & 0 \\ 7 & 0 & \infty \\ \infty & 1 & 0 \end{pmatrix} \\ & \begin{array}{ccc}6 & 0 & 0\end{array} \end{array} \rightarrow \begin{array}{cc} & \begin{array}{ccc}1 & 2 & 3\end{array} \\ \begin{array}{c}2\\3\\4\end{array} & \begin{pmatrix} 0 & \infty & 0 \\ 1 & 0 & \infty \\ \infty & 1 & 0 \end{pmatrix} \end{array}$$

7. Найдем h=1+6=7 и найдем оценку {1,4}: $w(\{1,4\})=w(G_0)+h=22+7=29$

Метод ветвей и границ решения задачи коммивояжера Пример

6. Приведем матрицу

$$\begin{array}{c} \\ 2 \\ 3 \\ 4 \end{array}\begin{array}{c} \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \begin{pmatrix} 6 & \infty & 0 \\ 7 & 0 & \infty \\ \infty & 2 & 1 \end{pmatrix} \end{array}\begin{array}{c} \\ 0 \\ 0 \\ 1 \end{array} \rightarrow \begin{array}{c} \\ 2 \\ 3 \\ 4 \\ \\ \end{array}\begin{array}{c} \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \begin{pmatrix} 6 & \infty & 0 \\ 7 & 0 & \infty \\ \infty & 1 & 0 \end{pmatrix} \\ \begin{array}{ccc} 6 & 0 & 0 \end{array} \end{array} \rightarrow \begin{array}{c} \\ 2 \\ 3 \\ 4 \end{array}\begin{array}{c} \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \begin{pmatrix} 0 & \infty & 0 \\ 1 & 0 & \infty \\ \infty & 1 & 0 \end{pmatrix} \end{array}$$

7. Найдем h=1+6=7 и найдем оценку {1,4}: $w(\{1,4\})=w(G_0)+h=22+7=29$

8. Укажем претендентов для ветвления и выберем пару для ветвления:
$S_{21}=0$, $S_{23}=0$, $S_{32}=0$, $S_{43}=0$
$Q_{21}=0+1=1$, $Q_{23}=0+0=0$, $Q_{32}=1+1=2$, $Q_{43}=1+0=1$
$\max Q_{ij}=Q_{32}=2$

$$\begin{array}{c} \\ 2 \\ 3 \\ 4 \\ \\ \end{array}\begin{array}{c} \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \begin{pmatrix} 0 & \infty & 0 \\ 1 & 0 & \infty \\ \infty & 1 & 0 \end{pmatrix} \\ \begin{array}{ccc} 1 & 1 & 0 \end{array} \end{array}\begin{array}{c} \\ 0 \\ 1 \\ 1 \\ \\ \end{array}$$

Итак, для ветвления берем пару (3,2).

Метод ветвей и границ решения задачи коммивояжера Пример

9. Найдем оценку {3,2}: $w(\{3,2\})=w(\{1,4\})+Q_{32}=29+2=31$

Метод ветвей и границ решения задачи коммивояжера
Пример

9. Найдем оценку {3,2}: $w(\{3,2\})=w(\{1,4\})+Q_{32}=29+2=31$

10. Вычеркнем строку «3», столбец «2» и наложим запрет на переезд из «2» в «3».

$$\begin{array}{cc} & \begin{array}{cc} 1 & 3 \end{array} \\ \begin{array}{c} 2 \\ 4 \end{array} & \begin{pmatrix} 0 & \infty \\ \infty & 0 \end{pmatrix} \end{array}$$

Метод ветвей и границ решения задачи коммивояжера Пример

9. Найдем оценку {3,2}: $w(\{3,2\})=w(\{1,4\})+Q_{32}=29+2=31$

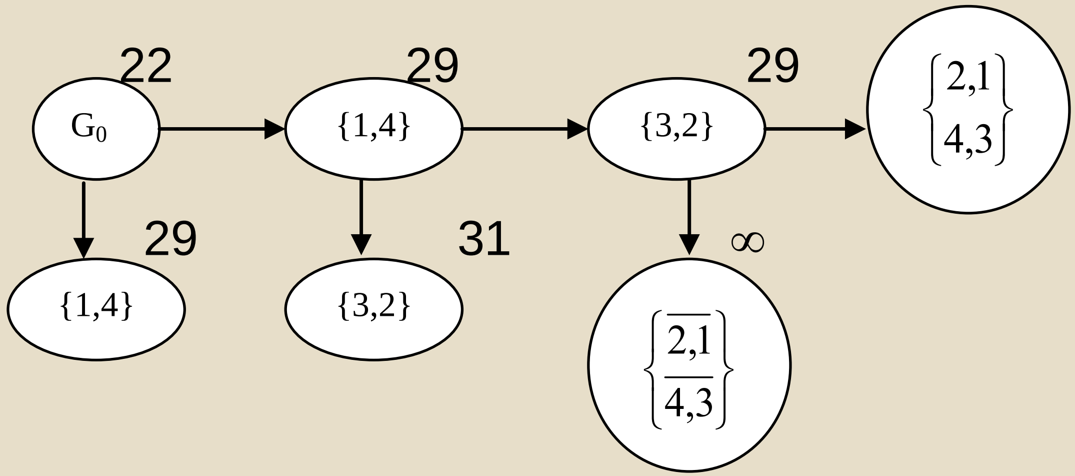
10. Вычеркнем строку «3», столбец «2» и наложим запрет на переезд из «2» в «3».

$$\begin{array}{cc} & \begin{array}{cc} 1 & 3 \end{array} \\ \begin{array}{c} 2 \\ 4 \end{array} & \begin{pmatrix} 0 & \infty \\ \infty & 0 \end{pmatrix} \end{array}$$

11. Матрица приведенная, следовательно $h=0$ и $w(\{3,2\})=w(\{1,4\})+h=29$

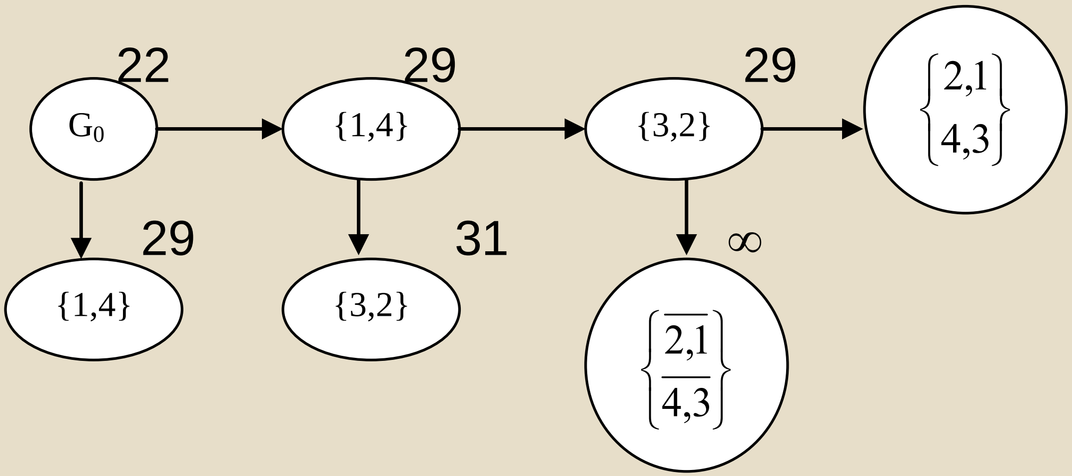
Метод ветвей и границ решения задачи коммивояжера Пример

12. Так как матрица размерности 2*2, то не запрещенные пары (2,1) и (4,3) завершают маршрут.



Метод ветвей и границ решения задачи коммивояжера Пример

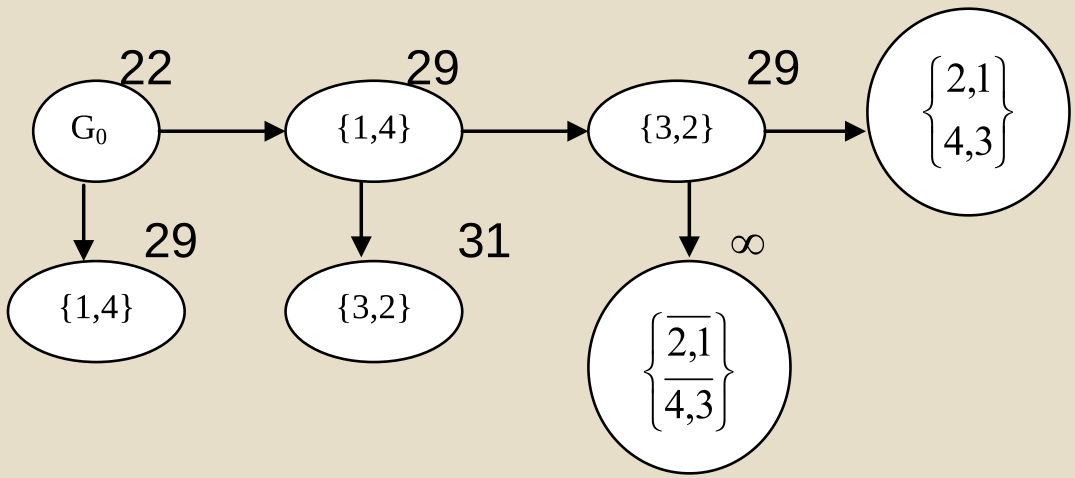
12. Так как матрица размерности 2*2, то не запрещенные пары (2,1) и (4,3) завершают маршрут.



13. Проверяем критерий оптимальности.

Метод ветвей и границ решения задачи коммивояжера Пример

12. Так как матрица размерности 2*2, то не запрещенные пары (2,1) и (4,3) завершают маршрут.



13. Проверяем критерий оптимальности.

Поскольку оценка последнего подмножества не больше оценок тупиковых ветвей, то пары (1,4), (3,2), (2,1), (4,3) задают оптимальный маршрут, который можно представить, скажем так:
1-4-3-2-1 с расстоянием S_{min}=29.

Метод ветвей и границ решения задачи коммивояжера Пример2

	1	2	3	4	5	6
1	∞	4	10	13	7	5
2	4	∞	3	4	1	12
3	10	3	∞	7	5	9
4	13	4	7	∞	11	8
5	7	1	5	11	∞	2
6	5	12	9	8	2	∞

Метод ветвей и границ решения задачи коммивояжера Пример2

	1	2	3	4	5	6	min
1	∞	**4**	**10**	**13**	**7**	**5**	**4**
2	4	∞	3	4	1	12	**1**
3	10	3	∞	7	5	9	**3**
4	13	4	7	∞	11	8	**4**
5	7	1	5	11	∞	2	**1**
6	5	12	9	8	2	∞	**2**

	1	2	3	4	5	6
1	∞	0	6	9	3	1
2	3	∞	2	3	0	11
3	7	0	∞	4	2	6
4	9	0	3	∞	7	4
5	6	0	4	10	∞	1
6	3	10	7	6	0	∞
min	**3**	**0**	**2**	**3**	**0**	**1**

	1	2	3	4	5	6	min
1	∞	0	4	6	3	0	0
2	0	∞	0	0	0	10	0
3	4	0	∞	1	2	5	1
4	6	0	1	∞	7	3	1
5	3	0	2	7	∞	0	0
6	0	10	5	3	0	∞	0
min	**0**	**0**	**1**	**1**	**0**	**0**	

h=4+1+3+4+1+2+3+0+2+3+0+1
h=24

Считаем оценки:
Q12=0, Q16=0, Q21=0, **Q23=1,**
Q24=1, Q25=0, Q32=1, Q42=1,
Q56=0, Q52=0,Q61=0, Q65=0

Метод ветвей и границ решения задачи коммивояжера Пример2

	1	2	4	5	6	min
1	∞	0	6	3	0	**0**
3	4	∞	1	2	5	**1**
4	6	0	∞	7	3	**0**
5	3	0	7	∞	0	**0**
6	0	10	3	0	∞	**0**

	1	2	4	5	6
1	∞	0	6	3	0
3	3	∞	0	1	4
4	6	0	∞	7	3
5	3	0	7	∞	0
6	0	10	3	0	∞
min	**0**	**0**	**0**	**0**	

h=1

Считаем оценки:

Q12=0, Q42=3, Q61=3, Q16=0,
Q52=0, Q65=1, **Q34=4,** Q56=0

Метод ветвей и границ решения задачи коммивояжера
Пример2

	1	2	5	6	min
1	∞	0	3	0	**0**
4	6	∞	7	3	**3**
5	3	0	∞	0	**0**
6	0	10	0	∞	**0**

	1	2	5	6
1	∞	0	3	0
4	6	∞	7	3
5	3	0	∞	0
6	0	10	0	∞
min	**0**	**0**	**0**	**0**

h=3
Считаем оценки:
Q12=0, Q52=0, Q65=1, Q16=0,
Q56=0, **Q46=3,** Q61=3

Метод ветвей и границ решения задачи коммивояжера
Пример2

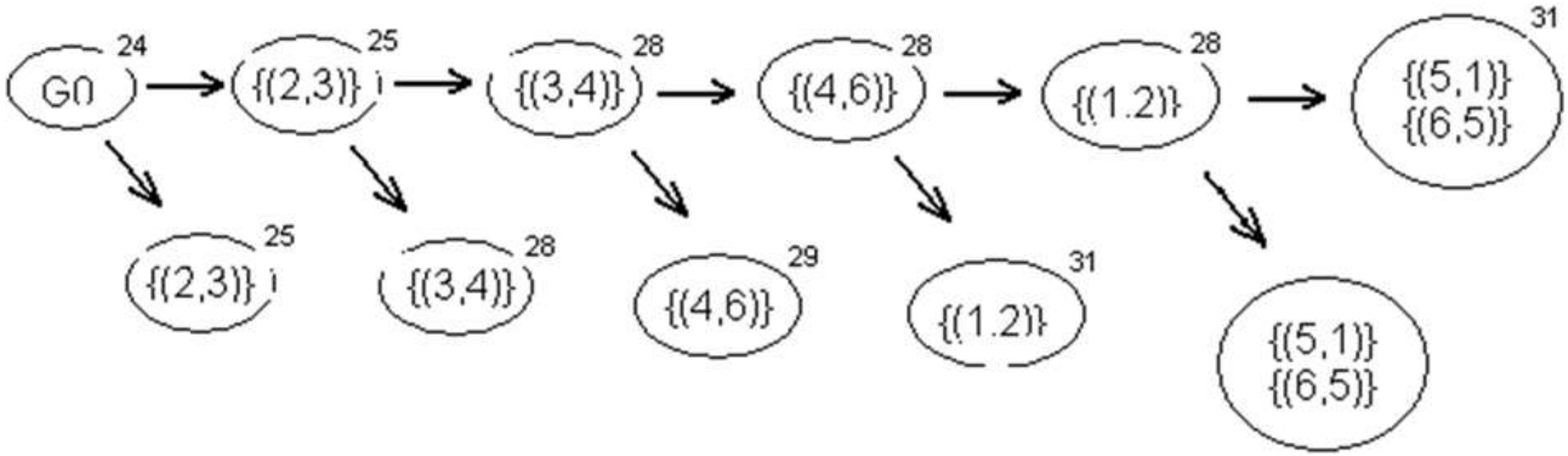
	1	2	5	min
1	∞	0	3	**0**
5	3	0	∞	**0**
6	0	∞	0	**0**
min	0	0	0	

h=0
Считаем оценки:
h=3
Q12=3 Q61=3, Q22=3, Q65=3

	1	5	min
5	3	∞	**3**
6	∞	0	**0**

	1	5
5	0	∞
6	∞	0

Метод ветвей и границ решения задачи коммивояжера
Пример2



Метод ветвей и границ решения задачи коммивояжера
Пример2

Возврат:
Ставим запрет на (2,3)

	1	2	3	4	5	6	min
1	∞	4	10	13	7	5	**4**
2	4	∞	∞	4	1	12	**1**
3	10	3	∞	7	5	9	**3**
4	13	4	7	∞	11	8	**4**
5	7	1	5	11	∞	2	**1**
6	5	12	9	8	2	∞	**2**

	1	2	3	4	5	6	min
1	∞	0	3	6	3	0	0
2	0	∞	∞	0	0	10	0
3	4	0	∞	1	2	5	1
4	6	0	0	∞	7	3	0
5	3	0	1	7	∞	0	0
6	0	10	4	3	0	∞	0
min	**0**	**0**	**1**	**1**	**0**	**1**	

	1	2	3	4	5	6
1	∞	0	6	9	3	1
2	3	∞	∞	3	0	11
3	7	0	∞	4	2	6
4	9	0	3	∞	7	4
5	6	0	4	10	∞	1
6	3	10	7	6	0	∞
min	**3**	**0**	**3**	**3**	**0**	**1**

Считаем оценки:
Q12=0, Q16=0,Q21=0, **Q24=1,**
Q25=0, Q32=1, Q42=0, Q43=1,
Q52=0, Q56=0, Q61=0, Q65=0

h=4+1+3+4+1+2+3+0+3+3+0+1
h=25

Метод ветвей и границ решения задачи коммивояжера
Пример2

	1	2	3	5	6	min
1	∞	0	3	3	0	**0**
3	4	0	∞	2	5	**0**
4	6	∞	0	7	3	**0**
5	3	0	1	∞	0	**0**
6	0	10	4	0	∞	**0**
min	0	0	0	0	**0**	

h=0
Считаем оценки:
Q12=0, Q32=2, Q52=0, Q61=3,
Q16=0, **Q43=4,** Q56=0, Q65=2

Метод ветвей и границ решения задачи коммивояжера Пример2

	1	2	5	6	min
1	∞	0	3	0	**0**
3	4	∞	2	5	**2**
5	3	0	∞	0	**0**
6	0	10	0	∞	**0**

	1	2	5	6
1	∞	0	3	0
3	2	∞	0	5
5	3	0	∞	0
6	0	10	0	∞
min	**0**	**0**	**0**	**0**

h=2
Считаем оценки:
Q12=0, Q52=0, Q65=0, Q16=0,
Q56=0, **Q35=2,** Q61=2

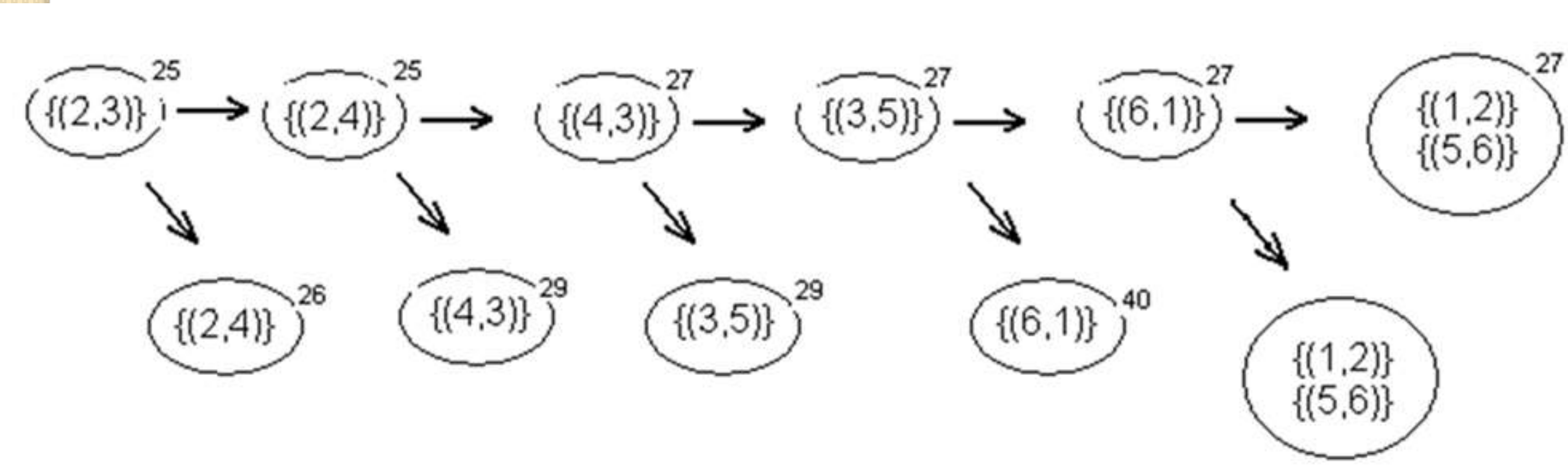
Метод ветвей и границ решения задачи коммивояжера Пример2

	1	2	6	min
1	∞	0	0	**0**
5	3	∞	0	**0**
6	0	10	∞	**0**
min	**0**	**0**	**0**	

h=0
Считаем оценки:
Q12=10, Q16=0, Q56=3, **Q61=13**

	2	6
1	0	∞
5	∞	0

Метод ветвей и границ решения задачи коммивояжера
Пример2



Метод ветвей и границ решения задачи коммивояжера Пример2

Возврат:
Ставим запрет на (2,4)

	1	2	3	4	5	6	min
1	∞	4	10	13	7	5	**4**
2	4	∞	∞	∞	1	12	**1**
3	10	3	∞	7	5	9	**3**
4	13	4	7	∞	11	8	**4**
5	7	1	5	11	∞	2	**1**
6	5	12	9	8	2	∞	**2**

	1	2	3	4	5	6	min
1	∞	0	3	5	3	0	0
2	0	∞	∞	∞	0	10	0
3	4	0	∞	0	2	5	0
4	6	0	0	∞	7	3	0
5	3	0	1	6	∞	0	0
6	0	10	4	2	0	∞	0
min	**0**	**0**	**1**	**2**	**0**	**0**	

	1	2	3	4	5	6
1	∞	0	6	9	3	1
2	3	∞	∞	∞	0	11
3	7	0	∞	4	2	6
4	9	0	3	∞	7	4
5	6	0	4	10	∞	1
6	3	10	7	6	0	∞
min	**3**	**0**	**3**	**4**	**0**	**1**

Считаем оценки:
Q12=0, Q32=0, Q52=0, Q16=0, **Q34=2,** Q56=0, Q21=0, Q42=0, Q61=0, Q25=0, Q43=1, Q65=0

h=4+1+3+4+1+2+3+0+3+4+0+1
h=26

Метод ветвей и границ решения задачи коммивояжера Пример2

	1	2	3	5	6	min
1	∞	0	3	3	0	**0**
2	0	∞	∞	0	10	**0**
4	6	0	∞	7	3	**0**
5	3	0	1	∞	0	**0**
6	0	10	4	0	∞	**0**
min	0	0	1	0	**0**	

h=1
Считаем оценки:
Q12=0, **Q42=3,** Q61=0, Q16=0, Q52=0,
Q65=0, Q21=0, Q53=2, Q25=0, Q56=0

	1	3	5	6	min
1	∞	2	3	0	**0**
2	0	∞	0	10	**0**
5	3	0	∞	0	**0**
6	0	3	0	∞	**0**
min	0	0	0	**0**	

h=0
Считаем оценки:
Q16=2, Q56=0, Q21=0, Q61=0
Q22=0, Q65=0, Q53=2

Метод ветвей и границ решения задачи коммивояжера Пример2

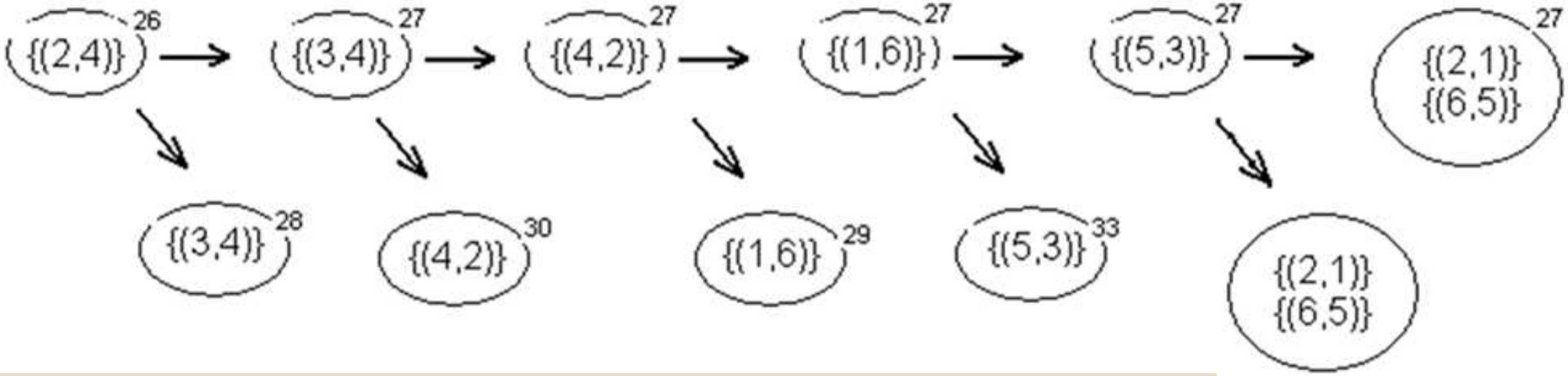
	1	3	5	min
2	0	∞	0	**0**
5	3	0	∞	**0**
6	∞	3	0	**0**
min	0	0	0	

h=0
Считаем оценки:
Q21=3, **Q53=6,** Q25=0, Q65=3

	1	5
2	0	∞
6	∞	0

h=0

Метод ветвей и границ решения задачи коммивояжера Пример2



ОТВЕТ:

Оптимальный маршрут: 1-6-5-3-4-2-1

Длина маршрута: 27

Делаем проверку: 5+4+7+4+5+2=27

	1	2	3	4	5	6
1	∞	4	10	13	7	5
2	4	∞	3	4	1	12
3	10	3	∞	7	5	9
4	13	4	7	∞	11	8
5	7	1	5	11	∞	2
6	5	12	9	8	2	∞

Ошибки(Билашевский Е Н + Скорый В.А.)

Слайд №8 (28:30) в 1)под min нет указания что j=I,n

2)Нет штриха над Cij после знака "="

Слайд №10 (33:44) Фраза 'сумма строк и столбцов '

Изменить на фразу'сумма строк и столбцов промежуточной матрицы'

Слайд№12(45:40) Указали что не нужно обращать внимание на фразу 'Двигаясь по дереву в обратном направлении получим '

Слайд№13(47:20)нет продолжение 'Поскольку необходимо чтобы в город j можно было бы попасть из некоторого другого города не равного j ,то каждый маршрут из не {I,j}должен содержать звено длина которого не меньше минимального элемента j-го столбца не считая Cij два штриха ' после фразы 'не меньше не меньше минимального элемента i-ой строки не считая Cij два штриха'

(50:10)Исправить фразу 'Назовем эту сумму оценкой пары(I,j)' или штрафом за неиспользование звена (I,j)

(50:50) В формуле под вторым min должно быть для любого j штрих ≠j

Слайд№15 (57:40) Конец пункта 4 должно быть Cji два штриха =∞

Слайд№22(1:08:30) Под пунктом 2 должно быть:'Найдем сумму всех вычитаемых элементов в процессе приведения',а не 'сумму приводящих констант', (1:09:25) когда проговаривали алгоритм решения была немного неправильная нумерация так на слайде пункт 1и 2 –один пункт алгоритма решения и так вся нумерация сбилась

Слайд№34(1:26:40)Не хватает черточки сверху или слова 'НЕ' во фразе 'Найдем оценку {3,2}'

Ошибки(Билашевский Е. Н. + Скорый В.А.)

- 1) Отсутствие черточек у тупиковых веток на слайдах 37, 38, 39 ($\{\overline{1,4}\}$ и $\{\overline{3,2}\}$).
- 2) h=3 на 44 слайде стоит не на том месте (должна стоять позже, после 2 таблички).
- 3) Отсутствие черточек у тупиковых веток на слайде 45 ($\{\overline{(2,3)}\}$, $\{\overline{(3,4)}\}$, $\{\overline{(4,6)}\}$, $\{\overline{(1,2)}\}$, $\{\overline{(5,1)}\}$, $\{\overline{(6,5)}\}$).
- 4) Отсутствие черточек у тупиковых веток и на начале маршрута на слайде 50 ($\{\overline{(2,3)}\}$, $\{\overline{(2,4)}\}$, $\{\overline{(4,3)}\}$, $\{\overline{(3,5)}\}$, $\{\overline{(6,1)}\}$, $\{\overline{(1,2)}\}$, $\{\overline{(5,6)}\}$).
- 5) Отсутствие черточек у тупиковых веток и на начале маршрута на слайде 54 ($\{\overline{(2,4)}\}$, $\{\overline{(3,4)}\}$, $\{\overline{(4,2)}\}$, $\{\overline{(1,6)}\}$, $\{\overline{(5,3)}\}$, $\{\overline{(2,1)}\}$, $\{\overline{(6,5)}\}$).

Методы оптимизации

Целочисленное программирование

Венгерский алгоритм решения задачи о назначениях

Д.В. Домашова

Постановка задачи о назначениях

n - видов ресурсов, которые нужно распределить на n объектов

C_{ij} – затраты (выигрыш (прибыль)), связанные с назначением ***i***-го ресурса на ***j***-ый объект.

Предполагается, что каждый ресурс назначается ровно один раз и каждому объекту приписывается ровно один ресурс.

Требуется: Минимизировать стоимость назначений.

Постановка задачи о назначениях

Определим неизвестные $\boldsymbol{x_{ij}}$:

$$x_{ij} = \begin{cases} 1, & \text{если } i-\text{ый ресурс назначанется на } j-\text{ый объек} \\ 0, & \text{иначе} \end{cases}$$

Постановка задачи о назначениях

Определим неизвестные x_{ij}:

$$x_{ij} = \begin{cases} 1, & \text{если } i - \text{ый ресурс назначанется на } j - \text{ый объек} \\ 0, & \text{иначе} \end{cases}$$

$$\sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij} \rightarrow \overset{1)}{min} \, (\overset{2)}{max})$$

$\sum_{i=1}^{n} x_{ij} = 1, \quad j = \overline{1,n}$ - на каждый *j*-ый объект – один вид ресурса.

$\sum_{j=1}^{n} x_{ij} = 1, \quad i = \overline{1,n}$ - каждый *i*-ый ресурс на один объект.

Постановка задачи о назначениях

Допустимое решение называется назначением.

Допустимое решение строится путем выбора ровно одного элемента в каждой строке матрицы ***X=[X$_{ij}$]*** и ровно одного элемента в каждом столбце этой матрицы.

Пример:

$$C = \begin{pmatrix} 4 & 7 & 0 \\ 0 & 3 & 8 \\ 6 & 3 & 9 \end{pmatrix} \qquad X = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$$

соответствующие ему затраты   допустимое решение

Венгерский алгоритм решения задачи о назначениях Обоснование

Рассмотрим задачу о назначениях с матрицей стоимостей ***C=[C$_{ij}$]***.

Предположим, что каждый элемент i-ой строки складывается с действительным числом γ_i, а каждый элемент j-столбца – с действительным числом δ_j.

В результате будет получена новая матрица стоимостей D:

$$d_{ij} = c_{ij} + \gamma_i + \delta_j$$

Покажем, что задача минимизации функции $\sum_i \sum_j c_{ij} x_{ij}$ эквивалентна минимизации функции $\sum_i \sum_j d_{ij} x_{ij}$ (т.е. такое преобразование не меняет точку минимума целевой функции)

Венгерский алгоритм решения задачи о назначениях Обоснование

$d_{ij} = c_{ij} + \gamma_i + \delta_j$

$c_{ij} = d_{ij} - \gamma_i - \delta_j$

$c_{ij}x_{ij} = d_{ij}x_{ij} - \gamma_i x_{ij} - \delta_j x_{ij}$

Венгерский алгоритм решения задачи о назначениях Обоснование

$d_{ij} = c_{ij} + \gamma_i + \delta_j$

$c_{ij} = d_{ij} - \gamma_i - \delta_j$

$c_{ij}x_{ij} = d_{ij}x_{ij} - \gamma_i x_{ij} - \delta_j x_{ij}$

$$\sum_i \sum_j \mathrm{c_{ij}x_{ij}} = \sum_i \sum_j \mathrm{d_{ij}x_{ij}} - \sum_i \sum_j \gamma_{\mathrm{ij}}\mathrm{x_{ij}} - \sum_i \sum_j \delta_{\mathrm{ij}}\mathrm{x_{ij}} =$$

Венгерский алгоритм решения задачи о назначениях
Обоснование

$d_{ij} = c_{ij} + \gamma_i + \delta_j$

$c_{ij} = d_{ij} - \gamma_i - \delta_j$

$c_{ij}x_{ij} = d_{ij}x_{ij} - \gamma_i x_{ij} - \delta_j x_{ij}$

$$\sum_i \sum_j c_{ij}x_{ij} = \sum_i \sum_j d_{ij}x_{ij} - \sum_i \sum_j \gamma_{ij}x_{ij} - \sum_i \sum_j \delta_{ij}x_{ij} =$$

$$= \sum_i \sum_j d_{ij}x_{ij} - \sum_i \gamma_i - \sum_j \delta_j$$

Венгерский алгоритм решения задачи о назначениях

- Идея венгерского алгоритма:
- Из элементов каждой строки и каждого столбца матрицы стоимостей вычитаются их наименьшие элементы, после чего ведется поиск допустимого решения, единичным элементам которого соответствуют нулевые элементы модифицированной матрицы стоимостей.
- Если такое допустимое решение существует, то оно является оптимальным назначением.
- Иначе матрица стоимостей модифицируется еще раз с целью получить в ней большее число нулевых элементов.

Венгерский алгоритм решения задачи о назначениях

Алгоритм состоит из трех шагов:

- **Редукция строк и столбцов**
- **Построение назначения**
- **Модификация матрицы стоимостей**

Венгерский алгоритм решения задачи о назначениях

1 Редукция строк и столбцов

- Из каждого элемента строки вычитается ее min элемент.
- Из каждого элемента столбца вычитается его min элемент.
- Цель шага – получить в каждой строке и столбце хотя бы один нулевой элемент.

Венгерский алгоритм решения задачи о назначениях

2. Построение назначения.

Если в каждой строке и в каждом столбце матрицы стоимостей можно выбрать по одному нулевому элементу $\Rightarrow$ соответствующее допустимое решение будет оптимальным, иначе goto п. 3.

а) Рассмотреть строки в порядке возрастания их номеров.

Найти строки, содержащие ровно один не вычеркнутый нулевой элемент.

В каждой такой строке произвести назначение, соответствующее не вычеркнутому нулевому элементу.

В каждом столбце, в котором было произведено назначение, вычеркнуть все не вычеркнутые ранее нулевые элементы.

Венгерский алгоритм решения задачи о назначениях

2. Построение назначения.

б) Рассмотреть столбцы в порядке возрастания их номеров.

Найти столбцы, содержащие ровно один не вычеркнутый нулевой элемент.

В каждом таком столбце произвести назначение, соответствующее этому нулевому элементу.

В каждой строке, в которой было произведено назначение, вычеркнуть все не вычеркнутые ранее нулевые элементы.

в) Выполнять а), б) до тех пор, пока не будет вычеркнуто max возможное число нулей.

Если построенное назначение полное $\Rightarrow$ оно оптимально, иначе goto п. 3.

Венгерский алгоритм решения задачи о назначениях

3. Модификация матрицы стоимостей.

- Определим для редуцированной матрицы стоимостей минимальное множество строк и столбцов, содержащих все нулевые элементы, и найдем минимальный элемент вне данного множества.
- Если значение данного элемента вычесть из всех остальных элементов матрицы, то на месте нулей будут стоять отрицательные величины и, по крайней мере, один элемент, не принадлежащий выделенному множеству строк и столбцов, станет равным нулю. Однако, теперь назначение нулевой стоимости может не быть оптимальным, поскольку матрица содержит отрицательные элементы. Для того, чтобы матрица не содержала отрицательных элементов, прибавим абсолютную величину наименьшего отрицательного элемента ко всем элементам выделенных строк и столбцов.
- Тогда к элементам расположенным на пересечении выделенных строк и столбцов, данная величина будет прибавляться дважды. Кроме того, как и раньше, все отрицательные элементы будут преобразованы в нулевые или положительные элементы.
- $\Rightarrow$ Новая матрица стала содержать больше нулей, расположенных вне строк и столбцов, соответствующих нулевым элементам текущего неоптимального решения.

Венгерский алгоритм решения задачи о назначениях

3. Модификация матрицы стоимостей.

а) Вычислить число 0-ей в каждой не вычеркнутой строке и каждом не вычеркнутом столбце.

б) Вычеркнуть строку или столбец с max числом нулей. В случае равенства числа 0 в нескольких строках и столбцах вычеркнуть любую из этих строк (или любую из столбцов).

в) Выполнять а), б) до тех пор, пока не будут вычеркнуты все нули.

г) Из всех не вычеркнутых элементов вычесть min не вычеркнутый элемент и прибавить его к каждому элементу, расположенному на пересечении двух линий.

Венгерский алгоритм решения задачи о назначениях

Пример

$$C = \begin{pmatrix} 2 & 10 & 9 & 7 \\ 15 & 4 & 14 & 8 \\ 13 & 14 & 16 & 11 \\ 4 & 15 & 13 & 19 \end{pmatrix}$$

Венгерский алгоритм решения задачи о назначениях

1. Редукция

$$C = \begin{pmatrix} 2 & 10 & 9 & 7 \\ 15 & 4 & 14 & 8 \\ 13 & 14 & 16 & 11 \\ 4 & 15 & 13 & 19 \end{pmatrix} \begin{array}{|c} 2 \\ 4 \\ 11 \\ 4 \end{array}$$

Венгерский алгоритм решения задачи о назначениях

1. Редукция

$$C=\begin{pmatrix} 2 & 10 & 9 & 7 \\ 15 & 4 & 14 & 8 \\ 13 & 14 & 16 & 11 \\ 4 & 15 & 13 & 19 \end{pmatrix}\begin{vmatrix} 2 \\ 4 \\ 11 \\ 4 \end{vmatrix}\rightarrow \begin{array}{c}\begin{pmatrix} 0 & 8 & 7 & 5 \\ 11 & 0 & 10 & 4 \\ 2 & 3 & 5 & 0 \\ 0 & 11 & 9 & 15 \end{pmatrix} \\ \hline \begin{matrix} 0 & 0 & 5 & 0 \end{matrix}\end{array}$$

Венгерский алгоритм решения задачи о назначениях

1. Редукция

$$C=\begin{pmatrix}2 & 10 & 9 & 7\\ 15 & 4 & 14 & 8\\ 13 & 14 & 16 & 11\\ 4 & 15 & 13 & 19\end{pmatrix}\begin{vmatrix}2\\4\\11\\4\end{vmatrix}\rightarrow \begin{array}{c}\begin{pmatrix}0 & 8 & 7 & 5\\ 11 & 0 & 10 & 4\\ 2 & 3 & 5 & 0\\ 0 & 11 & 9 & 15\end{pmatrix}\\ \hline \begin{matrix}0 & 0 & 5 & 0\end{matrix}\end{array} \qquad \begin{pmatrix}0 & 8 & 2 & 5\\ 11 & 0 & 5 & 4\\ 2 & 3 & 0 & 0\\ 0 & 11 & 4 & 15\end{pmatrix}$$

Венгерский алгоритм решения задачи о назначениях

2. Построение назначения

$$\begin{pmatrix} 0 & 8 & 2 & 5 \\ 11 & 0 & 5 & 4 \\ 2 & 3 & 0 & 0 \\ 0 & 11 & 4 & 15 \end{pmatrix}$$

Венгерский алгоритм решения задачи о назначениях

2. Построение назначения

$$\begin{pmatrix} 0* & 8 & 2 & 5 \\ 11 & 0* & 5 & 4 \\ 2 & 3 & 0* & 0 \\ 0 & 11 & 4 & 15 \end{pmatrix}$$

$\Rightarrow$ назначение не полное

Венгерский алгоритм решения задачи о назначениях

3. Модификация матрицы стоимостей

$$\begin{pmatrix} 0 & 8 & 2 & 5 \\ 11 & 0 & 5 & 4 \\ 2 & 3 & 0 & 0 \\ 0 & 11 & 4 & 15 \end{pmatrix}$$

Венгерский алгоритм решения задачи о назначениях

3. Модификация матрицы стоимостей

$$\begin{pmatrix} 0 & 8 & 2 & 5 \\ 11 & 0 & 5 & 4 \\ 2 & 3 & 0 & 0 \\ 0 & 11 & 4 & 15 \end{pmatrix}$$

$$\begin{pmatrix} 0 & 8 & 2 & 5 \\ 11 & 0 & 5 & 4 \\ 2 & 3 & 0 & 0 \\ 0 & 11 & 4 & 15 \end{pmatrix}$$

Венгерский алгоритм решения задачи о назначениях

3. Модификация матрицы стоимостей

$$\begin{pmatrix} 0 & 8 & 2 & 5 \\ 11 & 0 & 5 & 4 \\ 2 & 3 & 0 & 0 \\ 0 & 11 & 4 & 15 \end{pmatrix}$$

$$\begin{pmatrix} 0 & 8 & 2 & 5 \\ 11 & 0 & 5 & 4 \\ 2 & 3 & 0 & 0 \\ 0 & 11 & 4 & 15 \end{pmatrix}$$

min среди не вычеркнутых = 2
Вычитаем его из невычеркнутых элементов и прибавляем к элементам, расположенному на пересечении линий.

$$\begin{pmatrix} 0 & 6 & 0 & 3 \\ 13 & 0 & 5 & 4 \\ 4 & 3 & 0 & 0 \\ 0 & 9 & 2 & 13 \end{pmatrix}$$

Венгерский алгоритм решения задачи о назначениях

3. Модификация матрицы стоимостей

$$\begin{pmatrix} 0 & 8 & 2 & 5 \\ 11 & 0 & 5 & 4 \\ 2 & 3 & 0 & 0 \\ 0 & 11 & 4 & 15 \end{pmatrix}$$

$$\begin{pmatrix} 0 & 8 & 2 & 5 \\ 11 & 0 & 5 & 4 \\ 2 & 3 & 0 & 0 \\ 0 & 11 & 4 & 15 \end{pmatrix}$$

min среди не вычеркнутых = 2
Вычитаем его из невычеркнутых элементов и прибавляем к элементам, расположенному на пересечении линий.

$$\begin{pmatrix} 0 & 6 & 0 & 3 \\ 13 & 0 & 5 & 4 \\ 4 & 3 & 0 & 0 \\ 0 & 9 & 2 & 13 \end{pmatrix}$$

Выполняем назначение

$$\begin{pmatrix} 0 & 6 & 0* & 3 \\ 13 & 0* & 5 & 4 \\ 4 & 3 & 0 & 0* \\ 0* & 9 & 2 & 13 \end{pmatrix}$$

Назначение полное $\Rightarrow$ оптимальное
Стоимость оптимального назначения:
9 + 4 + 4 + 11 = 28

2 слайд C_{ij}-________ $1 \le i \le n$ $1 \le j \le n$
3 слайд обрезано слово объек**Т**
4 слайд в конце написать что X_{ij} принадлежат множеству состоящем из 0 и 1 для всех $1 \le i \le n$ $1 \le j \le n$
8 слайд <u>написать</u> γ_i(гамма) δ_j(дельта) <u>вместо</u> γ_{ij}(гамма) δ_{ij}(дельта)
26 слайд выписать в конце ответ – матрицу x^*

Ошибки искали:Шемякин И.В.
и Косвинцев К.Е.
Группа:С18-702

Методы оптимизации

Динамическое программирование

Д.В. Домашова

Динамическое программирование

- Это раздел математического программирования, посвященный исследованию многошаговых задач принятия оптимальных решений.

При этом многошаговость задачи:

- либо отражает реальное протекание принятия решений во времени;
- либо вводится в эту задачу искусственно за счет расчленения процесса принятия однократного решения на отдельные этапы, шаги.

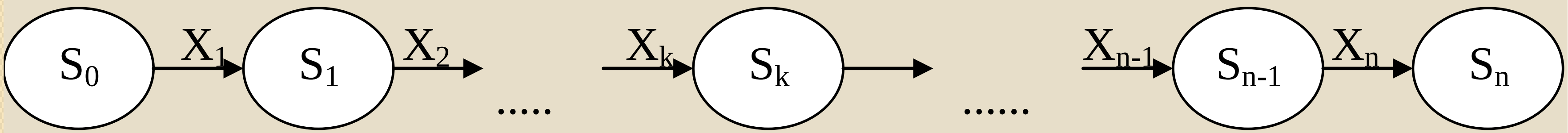
Динамическое программирование

- Цель такого представления состоит в сведении исходной задачи высокой размерности к решению на каждом шаге задачи небольшой размерности (часто одномерной).
- Методы ДП могут применяться к разнообразным задачам планирования и управления, например, управление запасами, замены и ремонта оборудования и др.

Задача управления (общая схема многошагового процесса)

Пусть имеется некоторая система S. В результате управления эта система переводится из некоторого состояния S_0 в конечное состояние S_n.

Предположим, что управление можно разбить на “n” шагов, т.е. решение принимается последовательно на любом шаге, а управление, переводящие систему S из состояния S_0 в S_n, представляет собой совокупность “n” пошаговых управлений.



$X = (X_1, ..., X_n)$ – управление (политика), переводящее систему S из S_0 в S_n, где X_k – управление на к-ом шаге.

Задача управления (общая схема многошагового процесса)

- $X = (X_1, \ldots, X_n)$ – управление (политика), переводящее системy S из S_0 в S_n, где

  X_k – управление на к-ом шаге.

- Переменные X_k удовлетворяют некоторым ограничениям: как исходным, так и ограничениям, возникающим за счет ранее сделанных выборов

  $X_1, \ldots, X_{к-1}$.

- Каждое решение приносит определенный выигрыш (доход), при этом качество каждого из управлений X характеризуется соответствующем значением функции $W = F(S_0, X)$ – показатель эффективности.

Условия, которым должна удовлетворять задача, решаемая методом ДП

1. Состояние S_k системы после к-ого шага зависит только от предшествующего состояния S_{k-1} и управления на к-ом шаге X_k и не зависит от предшествующих состояний и управлений. Это требование называется “отсутствием последействия”.

   $S_k = \varphi_k(S_{k-1}, X_k),$ - уравнение состояний $k = \overline{1, n}$ (1)

2. Целевая функция W = F(S_0, X) является аддитивной от показателя эффективности каждого шага

   $$W_k = f_k(S_{k-1}, X_k), k = \overline{1, n}$$

   $$\Rightarrow W = \sum_{k=1}^{n} W_k = \sum_{k=1}^{n} f_k(S_{k-1}, X_k) \qquad (2)$$

Последовательность X = (X_1, ..., X_n) допустимых управлений X_k на отдельных шагах называется политикой.

Задача управления (общая схема многошагового процесса)

Задача управления состоит в поиске такой оптимальной стратегии управления (оптимальной политики) $X^* = (X_1^*, ..., X_n^*)$, в результате реализации которой система S переходит из начального состояния S_0 в конечное S_n и при этом функция (2) принимает оптимальное значение (например, max), т.е. $W \to \max$.

Сформулированная задача является многоэтапной. В целом ряде задач многоэтапность не следует из их условий. Однако в целях нахождения решения методом ДП их следует рассматривать как многоэтапные.

Принцип оптимальности Беллмана(1953)

Оптимальная политика обладает тем свойством, что каковы бы ни были

- решение, принятое на 1-ом шаге, и
- состояние системы после первого шага,

последующие решения должны составлять оптимальное относительно этого состояния поведение.

Любое оптимальное решение может быть образовано только оптимальными частными решениями.

Принцип оптимальности Беллмана(1953)

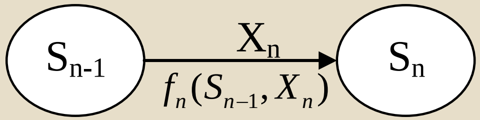
- Для применения принципа оптимальности в конкретных задачах пользуются приемом, часто называемым погружением.
- Он состоит в том, что вместо решения исходной задачи с данным начальным состоянием S_0 и данным числом шагов “n” решается целое семейство задач с произвольным начальным состоянием и с произвольным числом шагов .
- Формализация принципа оптимальности приводит к некоторым функциональным уравнениям, решение которых и составляет основу вычислительной схем ДП.
- Во многих случаях функциональные уравнения ДП представляют собой систему рекуррентных соотношений.
- Их называют уравнениями Беллмана.

Принцип оптимальности Беллмана(1953)

- Вместо исходной задачи с фиксированным числом шагов “n” и начальным состоянием S_0 рассматривается последовательность задач. Полагая последовательно n=1,2,... при различных S, получаем одношаговую, двухшаговую и т.д. задачи.
- На каждом шаге любого состояния системы S_{k-1} решение X_k нужно выбирать “с оглядкой”, т.к. этот выбор влияет на последующее состояние системы S_k и дальнейший процесс управления, зависящий от S_k.
- Но! На последнем шаге можно для любого состояния S_{n-1} планировать локально-оптимально, исходя из соображений этого шага.

Уравнения Беллмана

Согласно принципу оптимальности X_n нужно выбирать так, чтобы для любого состояния S_{n-1} получить максимум целевой функции на этом шаге.



Уравнения Беллмана

Согласно принципу оптимальности X_n нужно выбирать так, чтобы для любого состояния S_{n-1} получить максимум целевой функции на этом шаге.

$$S_{n-1} \xrightarrow[f_n(S_{n-1}, X_n)]{X_n} S_n$$

Обозначим $W_n^*(S_{n-1})$ - максимум целевой функции – показатель эффективности n-ого шага при условии, что к началу последнего шага система S была в произвольном состоянии S_{n-1}, а на последнем шаге управление было оптимальным.

$W_n^*(S_{n-1})$ называется условным max целевой функции на n-ом шаге.

$$W_n^*(S_{n-1}) = \max_{\{X_n\}} f_n(S_{n-1}, X_n) \quad \text{, где} \qquad (3)$$

максимум берется по всем допустимым управлениям X_n.

Уравнения Беллмана

Согласно принципу оптимальности X_n нужно выбирать так, чтобы для любого состояния S_{n-1} получить максимум целевой функции на этом шаге.

$$S_{n-1} \xrightarrow[f_n(S_{n-1}, X_n)]{X_n} S_n$$

Обозначим $W_n^*(S_{n-1})$ - максимум целевой функции – показатель эффективности n-ого шага при условии, что к началу последнего шага система S была в произвольном состоянии S_{n-1}, а на последнем шаге управление было оптимальным.

$W_n^*(S_{n-1})$ называется условным max целевой функции на n-ом шаге.

$$W_n^*(S_{n-1}) = \max_{\{X_n\}} f_n(S_{n-1}, X_n) \text{, где} \qquad (3)$$

максимум берется по всем допустимым управлениям X_n.

Решение X_n , при котором достигается $W_n^*(S_{n-1})$, также зависит от S_{n-1} и называется условным оптимальным управлением на n-ом шаге. Обозначим его $X_n^*(S_{n-1})$.

Решив одномерную задачу локальной оптимизации для всех возможных состояний S_{n-1}, получим две последовательности значений: $W_n^*(S_{n-1})$ и $X_n^*(S_{n-1})$.

Уравнения Беллмана

Согласно принципу оптимальности X_n нужно выбирать так, чтобы для любого состояния S_{n-1} получить максимум целевой функции на этом шаге.

$$S_{n-1} \xrightarrow[f_n(S_{n-1}, X_n)]{X_n} S_n$$

Обозначим $W_n^*(S_{n-1})$ - максимум целевой функции – показатель эффективности n-ого шага при условии, что к началу последнего шага система S была в произвольном состоянии S_{n-1}, а на последнем шаге управление было оптимальным.

$W_n^*(S_{n-1})$ называется условным max целевой функции на n-ом шаге.

$$W_n^*(S_{n-1}) = \max_{\{X_n\}} f_n(S_{n-1}, X_n) \text{, где} \qquad (3)$$

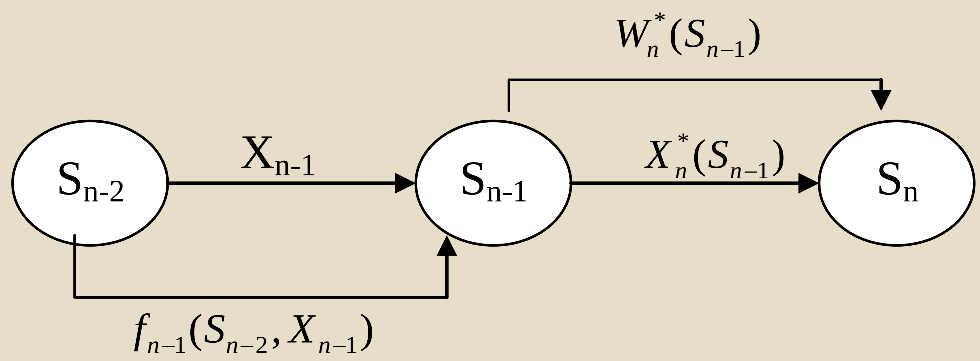
максимум берется по всем допустимым управлениям X_n.

Решение X_n , при котором достигается $W_n^*(S_{n-1})$, также зависит от S_{n-1} и называется условным оптимальным управлением на n-ом шаге. Обозначим его $X_n^*(S_{n-1})$.

Решив одномерную задачу локальной оптимизации для всех возможных состояний S_{n-1}, получим две последовательности значений: $W_n^*(S_{n-1})$ и $X_n^*(S_{n-1})$.

Уравнения Беллмана

Рассмотрим теперь 2-ух шаговую задачу: присоединим к n-ому шагу (n-1)-ый.

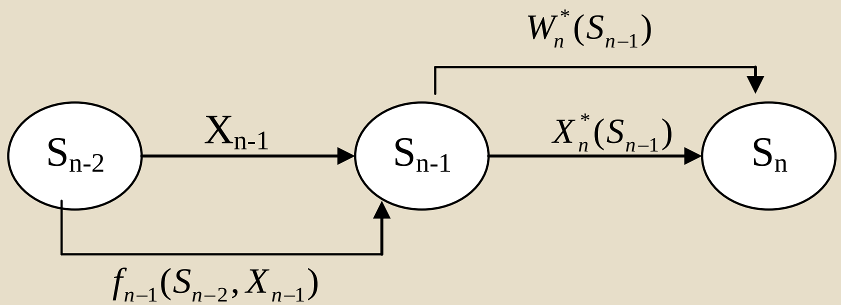


Для любого состояния S_{n-2} и любых произвольных управлений X_{n-1} и оптимальном управлении на n-ом шаге значение целевой функции на 2-ух последних шагах равно:

$$f_{n-1}(S_{n-2}, X_{n-1}) + W_n^*(S_{n-1}) \rightarrow \max \qquad (4)$$

Тогда по принципу оптимальности для любого S_{n-2} решение нужно выбирать так, чтобы оно вместе с оптимальным управлением на последнем n-ом шаге приводило бы к максимуму целевой функции на 2-ух последних шагах.

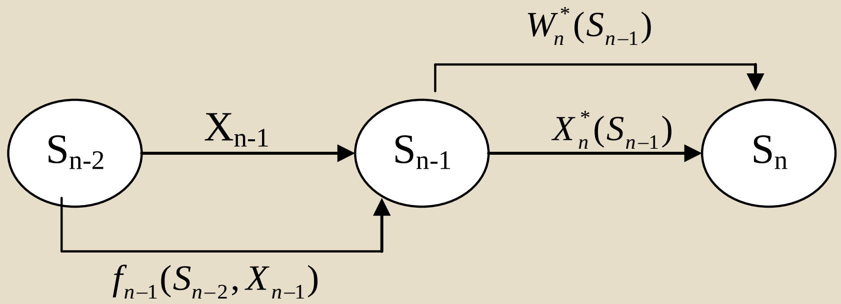
Уравнения Беллмана



Следовательно, ищем максимум (4) по всем допустимым X_{n-1}.

$$W_{n-1}^*(S_{n-2}) = \max_{\{X_{n-1}\}}\left\{f_{n-1}(S_{n-2}, X_{n-1}) + W_n^*(S_{n-1})\right\} \qquad (5)$$

Уравнения Беллмана



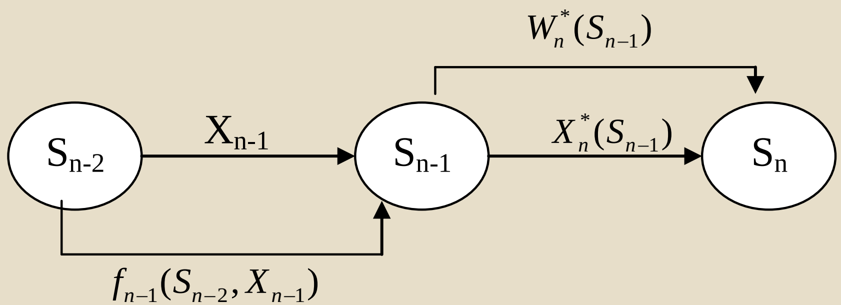
Следовательно, ищем максимум (4) по всем допустимым X_{n-1}.

$$W_{n-1}^*(S_{n-2}) = \max_{\{X_{n-1}\}} \left\{ f_{n-1}(S_{n-2}, X_{n-1}) + W_n^*(S_{n-1}) \right\} \qquad (5)$$

Максимум этой суммы зависит от S_{n-2} и равен $W_{n-1}^*(S_{n-2})$, называется условным максимумом целевой функции при оптимальном управлении на 2-ух последних шагах.

Соответствующее $X_{n-1}^*(S_{n-2})$ называется условным оптимальным управлением на (n-1)-ом шаге.

Уравнения Беллмана



Следовательно, ищем максимум (4) по всем допустимым X_{n-1}.

$$W_{n-1}^*(S_{n-2}) = \max_{\{X_{n-1}\}}\left\{f_{n-1}(S_{n-2}, X_{n-1}) + W_n^*(S_{n-1})\right\} \qquad (5)$$

Максимум этой суммы зависит от S_{n-2} и равен $W_{n-1}^*(S_{n-2})$, называется условным максимумом целевой функции при оптимальном управлении на 2-ух последних шагах.
Соответствующее $X_{n-1}^*(S_{n-2})$ называется условным оптимальным управлением на (n-1)-ом шаге.

$W_{n-1}^*(S_{n-2})$ зависит только от S_{n-2} и X_{n-1}, т.к. S_{n-1} можно найти из уравнения состояния (1): $S_k = \varphi_k(S_{k-1}, X_k)$ при k=n-1:

$S_{n-1} = \varphi_{n-1}(S_{k-2}, X_{n-1})$, подставим вместо S_{n-1} в функцию $W_n^*(S_{n-1})$.
В результате максимизируя только по одной переменной X_{n-1} вновь получаем две последовательности: $W_{n-1}^*(S_{n-2})$ и $X_{n-1}^*(S_{n-2})$.

Уравнения Беллмана

Далее рассматривается 3-х шаговая задача: к 2 –ум последним присоединяется (n-2)-ой шаг и т.д.

Уравнения Беллмана

Далее рассматривается 3-х шаговая задача: к 2 –ум последним присоединяется (n-2)-ой шаг и т.д.

Обозначим $W_k^*(S_{k-1})$ - условный максимум целевой функции, получаемый при оптимальном управлении на n-k+1 шагах, начиная с k-ого до конца, при условии, что к началу k–ого шага система находилась в состоянии S_{k-1}:

$$W_k^*(S_{k-1}) = \max_{\{(X_k,...,X_n)\}} \sum_{i=k}^{n} f_i(S_{i-1}, X_i) \Rightarrow W_{k+1}^*(S_k) = \max_{\{(X_{k+1},...,X_n)\}} \sum_{i=k+1}^{n} f_i(S_{i-1}, X_i)$$

Уравнения Беллмана

Далее рассматривается 3-х шаговая задача: к 2 –ум последним присоединяется (n-2)-ой шаг и т.д.

Обозначим $W_k^*(S_{k-1})$ - условный максимум целевой функции, получаемый при оптимальном управлении на n-k+1 шагах, начиная с k-ого до конца, при условии, что к началу k–ого шага система находилась в состоянии S_{k-1}:

$$W_k^*(S_{k-1}) = \max_{\{(X_k,\dots,X_n)\}} \sum_{i=k}^{n} f_i(S_{i-1}, X_i) \Rightarrow W_{k+1}^*(S_k) = \max_{\{(X_{k+1},\dots,X_n)\}} \sum_{i=k+1}^{n} f_i(S_{i-1}, X_i)$$

Целевая функция на (n-k) последних шагах при любом X_k и оптимальном управлении на последних (n-k) шагах равна:

$$f_k(S_{k-1}, X_k) + W_{k+1}^*(S_k) \to \max$$

По принципу оптимальности X_k выбирается из условия max, т.е.

$$W_k^*(S_{k-1}) = \max_{\{X_{n-1}\}} \left\{ f_k(S_{k-1}, X_k) + W_{k+1}^*(S_k) \right\} \qquad (6)$$

Уравнения Беллмана

Далее рассматривается 3-х шаговая задача: к 2 –ум последним присоединяется (n-2)-ой шаг и т.д.

Обозначим $W_k^*(S_{k-1})$ - условный максимум целевой функции, получаемый при оптимальном управлении на n-k+1 шагах, начиная с k-ого до конца, при условии, что к началу k–ого шага система находилась в состоянии S_{k-1}:

$$W_k^*(S_{k-1}) = \max_{\{(X_k,\dots,X_n)\}} \sum_{i=k}^{n} f_i(S_{i-1}, X_i) \Rightarrow W_{k+1}^*(S_k) = \max_{\{(X_{k+1},\dots,X_n)\}} \sum_{i=k+1}^{n} f_i(S_{i-1}, X_i)$$

Целевая функция на (n-k) последних шагах при любом X_k и оптимальном управлении на последних (n-k) шагах равна:

$$f_k(S_{k-1}, X_k) + W_{k+1}^*(S_k) \to \max$$

По принципу оптимальности X_k выбирается из условия max, т.е.

$$W_k^*(S_{k-1}) = \max_{\{X_{n-1}\}} \left\{ f_k(S_{k-1}, X_k) + W_{k+1}^*(S_k) \right\} \qquad (6)$$

Управление X_k на k-ом шаге, при котором достигается максимум обозначим $X_k^*(S_{k-1})$ - условное оптимальное управление на k-ом шаге.

Вместо S_k из уравнения состояния подставим $S_k = \varphi_k(S_{k-1}, X_k)$.

Уравнение (6) называется <u>уравнением Беллмана</u>.

Это рекуррентное соотношение, позволяющее найти предыдущее значение, зная последующее.

Уравнения Беллмана

Процесс решения уравнений

$$W_n^*(S_{n-1}) = \max_{\{X_{n-1}\}} f_n(S_{n-1}, X_n)$$

$$W_k^*(S_{k-1}) = \max_{\{X_{n-1}\}} \left\{ f_k(S_{k-1}, X_k) + W_{k+1}^*(S_k) \right\}, k = n-1, n-2, \ldots, 1$$

называется <u>условной оптимизацией</u>.

<u>Замечание</u>: Мы описали способ решения ДП, начинающийся с последнего шага. Можно n-ый и 1-ый шаги поменять местами.

Нахождение решения

В результате условной оптимизации получаются две последовательности:

1) $W_n^*(S_{n-1}), W_{n-1}^*(S_{n-2}), \ldots, W_2^*(S_1), W_1^*(S_0)$ - условные максимумы целевой функции на последнем, на двух последних,…, на “n” шагах

2) $X_n^*(S_{n-1}), X_{n-1}^*(S_{n-2}), \ldots, X_2^*(S_1), X_1^*(S_0)$ - условные оптимальные управления на n-ом, (n-1)-ом,…, 1 шагах.

Нахождение решения

В результате условной оптимизации получаются две последовательности:

1) $W_n^*(S_{n-1}), W_{n-1}^*(S_{n-2}), \ldots, W_2^*(S_1), W_1^*(S_0)$ - условные максимумы целевой функции на последнем, на двух последних,…, на “n” шагах
2) $X_n^*(S_{n-1}), X_{n-1}^*(S_{n-2}), \ldots, X_2^*(S_1), X_1^*(S_0)$ - условные оптимальные управления на n-ом, (n-1)-ом,…, 1 шагах.

Используя эти последовательности, можно найти решение задачи при данных «n» и S_0 следующим образом:
по определению $W_1^*(S_0)$ - условный максимум целевой функции за «n» шагов при условии, что к началу 1-ого шага система была в состоянии S_0, т.е. $W_{\max} = W_1^*(S_0)$.
Далее следует использовать последовательность условных оптимальных управлений и уравнения состояний.

Нахождение решения

В результате условной оптимизации получаются две последовательности:

1) $W_n^*(S_{n-1}), W_{n-1}^*(S_{n-2}), \dots, W_2^*(S_1), W_1^*(S_0)$ - условные максимумы целевой функции на последнем, на двух последних,…, на “n” шагах
2) $X_n^*(S_{n-1}), X_{n-1}^*(S_{n-2}), \dots, X_2^*(S_1), X_1^*(S_0)$ - условные оптимальные управления на n-ом, (n-1)-ом,…, 1 шагах.

Используя эти последовательности, можно найти решение задачи при данных «n» и S_0 следующим образом:

по определению $W_1^*(S_0)$ - условный максимум целевой функции за «n» шагов при условии, что к началу 1-ого шага система была в состоянии S_0, т.е. $W_{\max} = W_1^*(S_0)$.

Далее следует использовать последовательность условных оптимальных управлений и уравнения состояний.

При фиксированном S_0 получаем $X_1^* = X_1^*(S_0)$. Далее из уравнения состояний находим $S_1^* = \varphi_1(S_0, X_1^*)$ и подставляем это выражение в последовательность условных оптимальных управлений:

$X_2^* = X_2^*(S_1^*)$ и т.д. по следующей цепочке:

$$X_1^* = X_1^*(S_0) \to S_1^* = \varphi_1(S_0, X_1^*) \Rightarrow X_2^* = X_2^*(S_1^*) \to S_2^* = \varphi_2(S_1, X_2^*) \Rightarrow$$

$$X_3^* = X_3^*(S_2^*) \to \dots \to S_{n-1}^* = \varphi_{n-1}(S_{n-2}, X_{n-1}^*) \Rightarrow X_n^* = X_n^*(S_{n-1}^*)$$

$\Rightarrow$ получаем оптимальное решение задачи ДП: $X_1^* = (X_1^*, X_2^*, \dots, X_n^*)$

Задача распределения ресурсов

Имеется однородный ресурс в количестве S единиц, который должен быть распределен между N предприятиями.
Использование i-ым предприятием x_i единиц ресурса дает доход, определяемый значением нелинейной функции $f_i(x_i)$. Требуется найти распределение ресурсов между предприятиями, обеспечивающее максимальный доход.

$$F = \sum_{i=1}^{n} f_i(x_i) \to \max$$

$$\sum_{i=1}^{n} x_i = S, x_i \geq 0, i = \overline{1, n}$$

Задача распределения ресурсов

В данной задаче принятие решений является однократным, а многошаговость вводится формально. Вместо того, чтобы рассматривать допустимые варианты распределения ресурсов между “n” предприятиями и оценивать их эффективность, будем рассматривать следующий многошаговый процесс:

1 шаг состоит в оценки эффективности выделения ресурса на 1-ое предприятие (первое направление);

2 шаг : выделение ресурса на первые два предприятия;

...

n шаг : оценка эффективности распределения на “n” предприятий.

Следовательно, получаем “n” этапов, на каждом из которых состояние системы описывается объемом ресурса, подлежащим распределению между “k” предприятиями.

Задача распределения ресурсов

В данной задаче принятие решений является однократным, а многошаговость вводится формально. Вместо того, чтобы рассматривать допустимые варианты распределения ресурсов между "n" предприятиями и оценивать их эффективность, будем рассматривать следующий многошаговый процесс:

1 шаг состоит в оценки эффективности выделения ресурса на 1-ое предприятие (первое направление);

2 шаг : выделение ресурса на первые два предприятия;

...

n шаг : оценка эффективности распределения на "n" предприятий.

Следовательно, получаем "n" этапов, на каждом из которых состояние системы описывается объемом ресурса, подлежащим распределению между "k" предприятиями.

Управлениями будут являться решения об объеме ресурса, выделенного k-му предприятию.

Задача состоит в выборе таких управлений, при которых целевая функция принимает максимальное значение.

Задача распределения ресурсов

Для применения схемы ДП погружают данную задачу в семейство задач с любым числом шагов k≤n и любым запасом ресурса $X_C \le S$.

Пусть $W_i(C)$ - максимальный доход при распределении объема С ресурса между i предприятиями, $i=\overline{1,n-1}$

$$W_i(C)=\max\sum_{i=1}^{k} f_i(x_i), k=\overline{1,n},$$

где максимум берется по всем неотрицательным x_i : $x_1+...+x_k=C$.

Задача распределения ресурсов

Для применения схемы ДП погружают данную задачу в семейство задач с любым числом шагов k≤n и любым запасом ресурса $X_C \le S$.

Пусть $W_i(C)$ - максимальный доход при распределении объема С ресурса между i предприятиями, $i=\overline{1,n-1}$

$$W_i(C)=\max\sum_{i=1}^{k} f_i(x_i), k=\overline{1,n},$$

где максимум берется по всем неотрицательным x_i : $x_1+...+x_k=C$.

Следовательно, применение принципа оптимальности приводит к рекуррентным соотношениям:

$$W_i(C)=\max_{0\le X_i\le C}\left\{f_i(x_i)+W_{i-1}(C-x_i)\right\}, i=\overline{2,n-1},$$ при $\forall$ допустимых С

$$W_1(C)=\max_{0\le X_1\le C}\left\{f_1(x_1)\right\},$$ при $\forall$ допустимых С ($0\le C\le S$)

Значение функции $\varphi_n(C)$ вычисляется лишь для данного значения C=S:

$$W_n(S)=\max_{0\le X_{ni}\le S}\left\{f_n(x_n)+\varphi_{n-1}(C-x_n)\right\}$$

Задача распределения ресурсов

Рекуррентные соотношения позволяют вычислить значения $W_1(C)$, $W_2(C)$,…, $W_n(C)$ при <u>всех</u> допустимых С и найти оптимальные политики. Оптимальный доход для исходной задачи определяется значением $W_n(S)$.

Следовательно, зная $W_n(S)$ можно определить x_n^0, соответствующее оптимальному решению:

x_{n-1}^0 определяется из $W_{n-1}(S - x_n^0)$

x_{n-2}^0 определяется из $W_{n-2}(S - x_n^0 - x_{n-1}^0)$

.

.

.

x_1^0 определяется из $W_1(S - x_n^0 - x_{n-1}^0 - \ldots - x_2^0)$

Задача распределения ресурсов (пример)

0	0	0	0	0
1	3	4	5	7
2	8	2	3	10
3	5	6	2	1
4	4	3	3	2

1. $$\varphi_1(x) = \max_{0 \le x_1 \le x} \{f_1(x_1)\}$$

$\varphi_1(0)=0,\ x_1^0=0$

$\varphi_1(1)=\max\{0,3\}=3,\quad x_1^0=1$

$\varphi_1(2)=\max\{0,3,8\}=8,\ x_1^0=2$

$\varphi_1(3)=\max\{0,3,8,5\}=8,\ x_1^0=2$

$\varphi_1(4)=\max\{0,3,8,5,4\}=8,\ x_1^0=2$

Задача распределения ресурсов (пример)

0	0	0	0	0
1	3	4	5	7
2	8	2	3	10
3	5	6	2	1
4	4	3	3	2

2. $$\varphi_2(x) = \max_{0 \le x2 \le x} \{ f_2(x_2) + \varphi_1(x - x_2) \}$$

Задача распределения ресурсов (пример)

0	0	0	0	0
1	3	4	5	7
2	8	2	3	10
3	5	6	2	1
4	4	3	3	2

2. $$\varphi_2(x) = \max_{0 \le x2 \le x} \{ f_2(x_2) + \varphi_1(x - x_2) \}$$

$\varphi_2(0)=0,\ x_2^0=0$

$$\varphi_2(1) = \max \begin{Bmatrix} f_2(1) & +\varphi_1(0) \\ f_2(0) & +\varphi_1(1) \end{Bmatrix} = \max \begin{Bmatrix} 4+0 \\ 0+3 \end{Bmatrix} = 4,\ x_2^0 = 1$$

Задача распределения ресурсов (пример)

0	0	0	0	0
1	3	4	5	7
2	8	2	3	10
3	5	6	2	1
4	4	3	3	2

2. $$\varphi_2(x) = \max_{0 \le x2 \le x} \{f_2(x_2) + \varphi_1(x - x_2)\}$$

$\varphi_2(0)=0,\ x_2^0=0$

$$\varphi_2(1) = \max\begin{Bmatrix} f_2(1) + \varphi_1(0) \\ f_2(0) + \varphi_1(1) \end{Bmatrix} = \max\begin{Bmatrix} 4+0 \\ 0+3 \end{Bmatrix} = 4,\ x_2^0 = 1$$

$$\varphi_2(2) = \max\begin{Bmatrix} f_2(2) + \varphi_1(0) \\ f_2(1) + \varphi_1(1) \\ f_2(0) + \varphi_1(2) \end{Bmatrix} = \max\begin{Bmatrix} 2+0 \\ 4+3 \\ 0+8 \end{Bmatrix} = 8,\ x_2^0 = 0$$

Задача распределения ресурсов (пример)

0	0	0	0	0
1	3	4	5	7
2	8	2	3	10
3	5	6	2	1
4	4	3	3	2

2. $$\varphi_2(x) = \max_{0 \le x2 \le x} \{f_2(x_2) + \varphi_1(x - x_2)\}$$

$\varphi_2(0)=0,\ x_2^0=0$

$$\varphi_2(1) = \max\begin{Bmatrix} f_2(1) & +\varphi_1(0) \\ f_2(0) & +\varphi_1(1) \end{Bmatrix} = \max\begin{Bmatrix} 4+0 \\ 0+3 \end{Bmatrix} = 4,\ x_2^0 = 1$$

$$\varphi_2(2) = \max\begin{Bmatrix} f_2(2)+\varphi_1(0) \\ f_2(1)+\varphi_1(1) \\ f_2(0)+\varphi_1(2) \end{Bmatrix} = \max\begin{Bmatrix} 2+0 \\ 4+3 \\ 0+8 \end{Bmatrix} = 8,\ x_2^0 = 0$$

$$\varphi_2(3) = \max\begin{Bmatrix} f_2(3)+\varphi_1(0) \\ f_2(2)+\varphi_1(1) \\ f_2(1)+\varphi_1(2) \\ f_2(0)+\varphi_1(3) \end{Bmatrix} = \max\begin{Bmatrix} 6+0 \\ 2+3 \\ 4+8 \\ 0+8 \end{Bmatrix} = 12,\ x_2^0 = 1$$

Задача распределения ресурсов (пример)

0	0	0	0	0
1	3	4	5	7
2	8	2	3	10
3	5	6	2	1
4	4	3	3	2

2. $$\varphi_2(x) = \max_{0 \le x2 \le x} \{f_2(x_2) + \varphi_1(x - x_2)\}$$

$\varphi_2(0)=0,\ x_2^0=0$

$$\varphi_2(1) = \max\begin{Bmatrix} f_2(1) + \varphi_1(0) \\ f_2(0) + \varphi_1(1) \end{Bmatrix} = \max\begin{Bmatrix} 4+0 \\ 0+3 \end{Bmatrix} = 4,\ x_2^0 = 1$$

$$\varphi_2(2) = \max\begin{Bmatrix} f_2(2) + \varphi_1(0) \\ f_2(1) + \varphi_1(1) \\ f_2(0) + \varphi_1(2) \end{Bmatrix} = \max\begin{Bmatrix} 2+0 \\ 4+3 \\ 0+8 \end{Bmatrix} = 8,\ x_2^0 = 0$$

$$\varphi_2(3) = \max\begin{Bmatrix} f_2(3) + \varphi_1(0) \\ f_2(2) + \varphi_1(1) \\ f_2(1) + \varphi_1(2) \\ f_2(0) + \varphi_1(3) \end{Bmatrix} = \max\begin{Bmatrix} 6+0 \\ 2+3 \\ 4+8 \\ 0+8 \end{Bmatrix} = 12,\ x_2^0 = 1$$

$$\varphi_2(4) = \max\begin{Bmatrix} f_2(4) + \varphi_1(0) \\ f_2(3) + \varphi_1(1) \\ f_2(2) + \varphi_1(2) \\ f_2(1) + \varphi_1(3) \\ f_2(0) + \varphi_1(4) \end{Bmatrix} = \max\begin{Bmatrix} 3+0 \\ 6+3 \\ 2+8 \\ 4+8 \\ 0+8 \end{Bmatrix} = 12,\ x_2^0 = 1$$

Задача распределения ресурсов (пример)

0	0	0	0	0
1	3	4	5	7
2	8	2	3	10
3	5	6	2	1
4	4	3	3	2

3. $$\varphi_3(x) = \max_{0 \le x3 \le x}\{f_3(x_3) + \varphi_2(x - x_3)\}$$

Задача распределения ресурсов (пример)

0	0	0	0	0
1	3	4	5	7
2	8	2	3	10
3	5	6	2	1
4	4	3	3	2

3. $$\varphi_3(x) = \max_{0 \le x3 \le x} \{ f_3(x_3) + \varphi_2(x - x_3) \}$$

$\varphi_3(0)=0,\ x_3^0=0$

$$\varphi_3(1) = \max \begin{Bmatrix} f_3(1) & +\varphi_2(0) \\ f_3(0) & +\varphi_2(1) \end{Bmatrix} = \max \begin{Bmatrix} 5+0 \\ 0+4 \end{Bmatrix} = 5,\ x_3^0 = 1$$

Задача распределения ресурсов (пример)

0	0	0	0	0
1	3	4	5	7
2	8	2	3	10
3	5	6	2	1
4	4	3	3	2

3. $$\varphi_3(x) = \max_{0 \le x3 \le x} \{f_3(x_3) + \varphi_2(x - x_3)\}$$

$\varphi_3(0)=0,\ x_3^0=0$

$$\varphi_3(1) = \max\begin{Bmatrix} f_3(1) & +\varphi_2(0) \\ f_3(0) & +\varphi_2(1) \end{Bmatrix} = \max\begin{Bmatrix} 5+0 \\ 0+4 \end{Bmatrix} = 5,\ x_3^0 = 1$$

$$\varphi_3(2) = \max\begin{Bmatrix} f_3(2)+\varphi_2(0) \\ f_3(1)+\varphi_2(1) \\ f_3(0)+\varphi_2(2) \end{Bmatrix} = \max\begin{Bmatrix} 3+0 \\ 5+4 \\ 0+8 \end{Bmatrix} = 9,\ x_3^0 = 1$$

Задача распределения ресурсов (пример)

0	0	0	0	0
1	3	4	5	7
2	8	2	3	10
3	5	6	2	1
4	4	3	3	2

3. $$\varphi_3(x) = \max_{0 \le x3 \le x} \{f_3(x_3) + \varphi_2(x - x_3)\}$$

$\varphi_3(0)=0,\ x_3^0=0$

$$\varphi_3(1) = \max\begin{Bmatrix} f_3(1) & +\varphi_2(0) \\ f_3(0) & +\varphi_2(1) \end{Bmatrix} = \max\begin{Bmatrix} 5+0 \\ 0+4 \end{Bmatrix} = 5,\ x_3^0 = 1$$

$$\varphi_3(2) = \max\begin{Bmatrix} f_3(2)+\varphi_2(0) \\ f_3(1)+\varphi_2(1) \\ f_3(0)+\varphi_2(2) \end{Bmatrix} = \max\begin{Bmatrix} 3+0 \\ 5+4 \\ 0+8 \end{Bmatrix} = 9,\ x_3^0 = 1$$

$$\varphi_3(3) = \max\begin{Bmatrix} f_3(3)+\varphi_2(0) \\ f_3(2)+\varphi_2(1) \\ f_3(1)+\varphi_2(2) \\ f_3(0)+\varphi_2(3) \end{Bmatrix} = \max\begin{Bmatrix} 2+0 \\ 3+4 \\ 5+8 \\ 0+12 \end{Bmatrix} = 13,\ x_3^0 = 1$$

Задача распределения ресурсов (пример)

0	0	0	0	0
1	3	4	5	7
2	8	2	3	10
3	5	6	2	1
4	4	3	3	2

3. $$\varphi_3(x) = \max_{0\le x3\le x}\{f_3(x_3) + \varphi_2(x - x_3)\}$$

$\varphi_3(0)=0,\ x_3^0=0$

$$\varphi_3(1) = \max\begin{Bmatrix} f_3(1) + \varphi_2(0) \\ f_3(0) + \varphi_2(1) \end{Bmatrix} = \max\begin{Bmatrix} 5+0 \\ 0+4 \end{Bmatrix} = 5,\ x_3^0 = 1$$

$$\varphi_3(2) = \max\begin{Bmatrix} f_3(2) + \varphi_2(0) \\ f_3(1) + \varphi_2(1) \\ f_3(0) + \varphi_2(2) \end{Bmatrix} = \max\begin{Bmatrix} 3+0 \\ 5+4 \\ 0+8 \end{Bmatrix} = 9,\ x_3^0 = 1$$

$$\varphi_3(3) = \max\begin{Bmatrix} f_3(3) + \varphi_2(0) \\ f_3(2) + \varphi_2(1) \\ f_3(1) + \varphi_2(2) \\ f_3(0) + \varphi_2(3) \end{Bmatrix} = \max\begin{Bmatrix} 2+0 \\ 3+4 \\ 5+8 \\ 0+12 \end{Bmatrix} = 13,\ x_3^0 = 1$$

$$\varphi_3(4) = \max\begin{Bmatrix} f_3(4) + \varphi_2(0) \\ f_3(3) + \varphi_2(1) \\ f_3(2) + \varphi_2(2) \\ f_3(1) + \varphi_2(3) \\ f_3(0) + \varphi_2(4) \end{Bmatrix} = \max\begin{Bmatrix} 3+0 \\ 2+4 \\ 3+8 \\ 5+12 \\ 0+12 \end{Bmatrix} = 17,\ x_3^0 = 1$$

Задача распределения ресурсов (пример)

0	0	0	0	0
1	3	4	5	7
2	8	2	3	10
3	5	6	2	1
4	4	3	3	2

4 $\varphi_4(x) = \max_{0 \le x4 \le x} \{f_4(x_4) + \varphi_4(x - x_4)\}$

Задача распределения ресурсов (пример)

$$\begin{array}{c|cccc|c} 0 & 0 & 0 & 0 & 0 & \mathbf{4} \\ 1 & 3 & 4 & 5 & 7 & \\ 2 & 8 & 2 & 3 & 10 & \\ 3 & 5 & 6 & 2 & 1 & \\ 4 & 4 & 3 & 3 & 2 & \end{array}$$

$$\varphi_4(x) = \max_{0 \le x4 \le x} \{ f_4(x_4) + \varphi_4(x - x_4) \}$$

$$\varphi_4(0)=0$$

Задача распределения ресурсов (пример)

0	0	0	0	0	**4**
1	3	4	5	7	
2	8	2	3	10	
3	5	6	2	1	
4	4	3	3	2	

$$\varphi_4(x) = \max_{0 \le x4 \le x} \{ f_4(x_4) + \varphi_4(x - x_4) \}$$

$\varphi_4(0)=0$

$$\varphi_4(1) = \max_{X4} \begin{Bmatrix} f_4(1) & +\varphi_3(0) \\ f_4(0) & +\varphi_3(1) \end{Bmatrix} = \begin{Bmatrix} 7+0 \\ 0+3 \end{Bmatrix} = 7, x_4^0 = 1$$

Задача распределения ресурсов (пример)

0	0	0	0	0	***4***
1	3	4	5	7	
2	8	2	3	10	
3	5	6	2	1	
4	4	3	3	2	

$$\varphi_4(x) = \max_{0 \le x4 \le x} \{ f_4(x_4) + \varphi_4(x - x_4) \}$$

$$\varphi_4(0)=0$$

$$\varphi_4(1) = \max_{X4} \begin{Bmatrix} f_4(1) & +\varphi_3(0) \\ f_4(0) & +\varphi_3(1) \end{Bmatrix} = \begin{Bmatrix} 7+0 \\ 0+3 \end{Bmatrix} = 7, x_4^0 = 1$$

$$\varphi_4(2) = \max_{X4} \begin{Bmatrix} f_4(2) + \varphi_3(0) \\ f_4(1) + \varphi_3(1) \\ f_4(0) + \varphi_3(2) \end{Bmatrix} = \begin{Bmatrix} 10+0 \\ 7+5 \\ 0+9 \end{Bmatrix} = 12, x_4^0 = 0$$

Задача распределения ресурсов (пример)

0	0	0	0	0	4
1	3	4	5	7	
2	8	2	3	10	
3	5	6	2	1	
4	4	3	3	2	

$$\varphi_4(x) = \max_{0 \le x4 \le x} \{ f_4(x_4) + \varphi_4(x - x_4) \}$$

$$\varphi_4(0) = 0$$

$$\varphi_4(1) = \max_{X4} \begin{Bmatrix} f_4(1) + \varphi_3(0) \\ f_4(0) + \varphi_3(1) \end{Bmatrix} = \begin{Bmatrix} 7+0 \\ 0+3 \end{Bmatrix} = 7, x_4^0 = 1$$

$$\varphi_4(2) = \max_{X4} \begin{Bmatrix} f_4(2) + \varphi_3(0) \\ f_4(1) + \varphi_3(1) \\ f_4(0) + \varphi_3(2) \end{Bmatrix} = \begin{Bmatrix} 10+0 \\ 7+5 \\ 0+9 \end{Bmatrix} = 12, x_4^0 = 0$$

$$\varphi_4(3) = \max_{X4} \begin{Bmatrix} f_4(3) + \varphi_3(0) \\ f_4(2) + \varphi_3(1) \\ f_4(1) + \varphi_3(2) \\ f_4(0) + \varphi_3(3) \end{Bmatrix} = \begin{Bmatrix} 1+0 \\ 10+5 \\ 7+9 \\ 0+13 \end{Bmatrix} = 16, x_4^0 = 1$$

Задача распределения ресурсов (пример)

0	0	0	0	0	**4**
1	3	4	5	7	
2	8	2	3	10	
3	5	6	2	1	
4	4	3	3	2	

$$\varphi_4(x) = \max_{0 \le x4 \le x} \{ f_4(x_4) + \varphi_4(x - x_4) \}$$

$\varphi_4(0)=0$

$$\varphi_4(1) = \max_{X4} \begin{Bmatrix} f_4(1) + \varphi_3(0) \\ f_4(0) + \varphi_3(1) \end{Bmatrix} = \begin{Bmatrix} 7+0 \\ 0+3 \end{Bmatrix} = 7, \boldsymbol{x}_4^0 = 1$$

$$\varphi_4(2) = \max_{X4} \begin{Bmatrix} f_4(2) + \varphi_3(0) \\ f_4(1) + \varphi_3(1) \\ f_4(0) + \varphi_3(2) \end{Bmatrix} = \begin{Bmatrix} 10+0 \\ 7+5 \\ 0+9 \end{Bmatrix} = 12, \boldsymbol{x}_4^0 = 0$$

$$\varphi_4(3) = \max_{X4} \begin{Bmatrix} f_4(3) + \varphi_3(0) \\ f_4(2) + \varphi_3(1) \\ f_4(1) + \varphi_3(2) \\ f_4(0) + \varphi_3(3) \end{Bmatrix} = \begin{Bmatrix} 1+0 \\ 10+5 \\ 7+9 \\ 0+13 \end{Bmatrix} = 16, \boldsymbol{x}_4^0 = 1$$

$$\varphi_4(4) = \max_{X4} \begin{Bmatrix} f_4(4) + \varphi_3(0) \\ f_4(3) + \varphi_3(1) \\ f_4(2) + \varphi_3(2) \\ f_4(1) + \varphi_3(3) \\ f_4(0) + \varphi_3(4) \end{Bmatrix} = \begin{Bmatrix} 2+0 \\ 1+5 \\ 10+9 \\ 7+13 \\ 0+17 \end{Bmatrix} = 20, \boldsymbol{x}_4^0 = 1$$

Задача распределения ресурсов (пример)

0	0	0	0	0	**4**
1	3	4	5	7	
2	8	2	3	10	
3	5	6	2	1	
4	4	3	3	2	

$$\varphi_4(x) = \max_{0 \le x4 \le x} \{f_4(x_4) + \varphi_4(x - x_4)\}$$

$$\varphi_4(0)=0$$

$$\varphi_4(1) = \max_{X4} \begin{Bmatrix} f_4(1) + \varphi_3(0) \\ f_4(0) + \varphi_3(1) \end{Bmatrix} = \begin{Bmatrix} 7+0 \\ 0+3 \end{Bmatrix} = 7, \boldsymbol{x}_4^0 = 1$$

$$\varphi_4(2) = \max_{X4} \begin{Bmatrix} f_4(2) + \varphi_3(0) \\ f_4(1) + \varphi_3(1) \\ f_4(0) + \varphi_3(2) \end{Bmatrix} = \begin{Bmatrix} 10+0 \\ 7+5 \\ 0+9 \end{Bmatrix} = 12, \boldsymbol{x}_4^0 = 0$$

$$\varphi_4(3) = \max_{X4} \begin{Bmatrix} f_4(3) + \varphi_3(0) \\ f_4(2) + \varphi_3(1) \\ f_4(1) + \varphi_3(2) \\ f_4(0) + \varphi_3(3) \end{Bmatrix} = \begin{Bmatrix} 1+0 \\ 10+5 \\ 7+9 \\ 0+13 \end{Bmatrix} = 16, \boldsymbol{x}_4^0 = 1$$

$$\varphi_4(4) = \max_{X4} \begin{Bmatrix} f_4(4) + \varphi_3(0) \\ f_4(3) + \varphi_3(1) \\ f_4(2) + \varphi_3(2) \\ f_4(1) + \varphi_3(3) \\ f_4(0) + \varphi_3(4) \end{Bmatrix} = \begin{Bmatrix} 2+0 \\ 1+5 \\ 10+9 \\ 7+13 \\ 0+17 \end{Bmatrix} = 20, \boldsymbol{x}_4^0 = 1$$

4-му – $\boldsymbol{X_4^0}$**=** 1 $\Rightarrow$ $S=S-1=3$ (осталось 3) $\Rightarrow$ $\boldsymbol{\varphi_3(3)}$ при $\boldsymbol{X_3^0=1}$ $\Rightarrow$ 3-му – 1 $\Rightarrow$ $S=S-1=2$ (осталось 2) $\Rightarrow$ $\boldsymbol{\varphi_3(2)}$ при $\boldsymbol{X_2^0=0}$ $\Rightarrow$ $S=S-1=2$ (осталось 2) $\Rightarrow$ $\boldsymbol{\varphi_1(2)}$ при $\boldsymbol{X_1^0=2}$

$X^0= (2,0,1,1)$ Прибыль$=8+5+7=20$

Задача о рюкзаке

Имеется рюкзак грузоподъемностью W.

P_i – вес одного предмета i-ого типа
V_i – стоимость (ценность) одного предмета i-ого типа
X_i – число предметов i-ого типа, которые будут загружаться на транспортировочное средство.

Требуется заполнить его грузом, состоящим из предметов “N” различных типов таким образом, чтобы стоимость (ценность) всего груза была максимальной.

Задача о рюкзаке

Имеется рюкзак грузоподъемностью W.

P_i – вес одного предмета i-ого типа
V_i – стоимость (ценность) одного предмета i-ого типа
X_i – число предметов i-ого типа, которые будут загружаться на транспортировочное средство.

Требуется заполнить его грузом, состоящим из предметов “N” различных типов таким образом, чтобы стоимость (ценность) всего груза была максимальной.

$$W(x) = \sum_{i=1}^{N} x_i V_i \to \max$$

$$\sum_{i=1}^{N} x_i P_i \le S, x_i = 0,1,\ldots - \textit{целочисленное}$$

Задача о рюкзаке

Имеется рюкзак грузоподъемностью W.

P_i – вес одного предмета i-ого типа
V_i – стоимость (ценность) одного предмета i-ого типа
X_i – число предметов i-ого типа, которые будут загружаться на транспортировочное средство.

Требуется заполнить его грузом, состоящим из предметов “N” различных типов таким образом, чтобы стоимость (ценность) всего груза была максимальной.

$$W(x) = \sum_{i=1}^{N} x_i V_i \to \max$$

$$\sum_{i=1}^{N} x_i P_i \le S, x_i = 0,1,\ldots - \textit{целочисленное}$$

Задача о рюкзаке

Решение задачи разбивается на “n” этапов, на каждом из которых определяется максимальная стоимость груза, состоящего из предметов:1-го типа (1-ый этап); 1-го и 2-го типов (2-ой этап) и т.д.

1 шаг состоит в оценке ценности груза, состоящего из предметов 1-ого типа (первого вида);
2 шаг : состоит в оценке ценности груза, состоящего из предметов 2-ух типов (первого и второго вида);
...
n шаг : состоит в оценке ценности груза, состоящего из предметов всех типов.

Задача о рюкзаке

Решение задачи разбивается на “n” этапов, на каждом из которых определяется максимальная стоимость груза, состоящего из предметов:1-го типа (1-ый этап); 1-го и 2-го типов (2-ой этап) и т.д.

1 шаг состоит в оценке ценности груза, состоящего из предметов 1-ого типа (первого вида);

2 шаг : состоит в оценке ценности груза, состоящего из предметов 2-ух типов (первого и второго вида);

...

n шаг : состоит в оценке ценности груза, состоящего из предметов всех типов.

Рекуррентное уравнение Беллмана для задачи о рюкзаке

$W_i(C)$ - максимальная стоимость груза, состоящего из предметов типа $k=\overline{1,i}$

$$W_i(C) = \max_{0 \le X_i \le \left[\frac{C}{P_i}\right]} \{x_i V_i + W_{i-1}(C - x_i P_i)\}$$

, при $\forall$ допустимых C ($0 \le C \le S$), где

$x_i V_i$ - стоимость взятых предметов i-го типа

$W_{i-1}(C - x_i P_i)$ - максимальная стоимость груза, состоящего из предметов типа $k=\overline{1,i-1}$ с общим весом не более $(C - x_i P_i)$

Будем считать $W_0(C)=0, \forall C: 0 \le C \le S$. Последовательно найдем значения функций, $W_1(C)$ $W_2(C)$ $,\ldots, W_{N-1}(C), при\ \forall \underline{\underline{C}}$, а также $W_N(S)$.

Задача о рюкзаке (пример)

Пусть S=10 – грузоподъемность и

C	1	2	3	4
P	4	3	2	1
V	28	20	13	6

Задача о рюкзаке (пример)

Пусть S=10 – грузоподъемность и

C	1	2	3	4
P	4	3	2	1
V	28	20	13	6

$$W_1(C) = \max_{0 \le X_1 \le \left[\frac{10}{4}\right]} \{x_1 \cdot 28\}, x_1 = 0,1,2$$

Задача о рюкзаке (пример)

Пусть S=10 – грузоподъемность и

C	1	2	3	4
P	4	3	2	1
V	28	20	13	6

$$W_1(C) = \max_{0 \le X_1 \le \left[\frac{10}{4}\right]} \{x_1 \cdot 28\},\ x_1 = 0,1,2$$

C	0-3	4-7	8-10
$W_1(C)$	0	28	56
X_1	0	1	2

Задача о рюкзаке (пример)

Пусть S=10 – грузоподъемность и

C	1	2	3	4
P	4	3	2	1
V	28	20	13	6

$$W_1(C) = \max_{0 \le X_1 \le \left[\frac{10}{4}\right]} \{x_1 \cdot 28\},\ x_1 = 0,1,2$$

C	0-3	4-7	8-10
$W_1(C)$	0	28	56
X_1	0	1	2

$$W_2(S) = \max_{0 \le X_2 \le \left[\frac{C}{3}\right]} \{x_2 \cdot 20 + W_1(C - X_2 \cdot 3)\},\ x_2 = 0,1,2,3$$

Задача о рюкзаке (пример)

Пусть S=10 – грузоподъемность и

C	1	2	3	4
P	4	3	2	1
V	28	20	13	6

$$W_1(C) = \max_{0 \le X_1 \le \left[\frac{10}{4}\right]} \{x_1 \cdot 28\}, x_1 = 0,1,2$$

C	0-3	4-7	8-10
$W_1(C)$	0	28	56
X_1	0	1	2

$$W_2(S) = \max_{0 \le X_2 \le \left[\frac{C}{3}\right]} \{x_2 \cdot 20 + W_1(C - X_2 \cdot 3)\}, x_2 = 0,1,2,3$$

C	0-2	3	4-5	6	7	8	9	10
$W_2(C)$	0	20	28	40	48	56	60	68
x_2	0	1	0	2	1	0	3	2

Задача о рюкзаке (пример)

Пусть S=10 – грузоподъемность и

C	1	2	3	4
P	4	3	2	1
V	28	20	13	6

$$W_3(S) = \max_{0 \le X_3 \le \left[\frac{C}{2}\right]} \{x_3 \cdot 13 + W_2(C - X_3 \cdot 2)\}, x_3 = 0,1,2,3,4,5$$

Задача о рюкзаке (пример)

Пусть S=10 – грузоподъемность и

C	1	2	3	4
P	4	3	2	1
V	28	20	13	6

$$W_3(S) = \max_{0 \le X_3 \le \left[\frac{C}{2}\right]} \{x_3 \cdot 13 + W_2(C - X_3 \cdot 2)\}, x_3 = 0,1,2,3,4,5$$

C	0-1	2	3	4	5	6	7	8	9	10
$W_3(C)$	0	13	20	28	33	41	48	56	61	69
x_3	0	1	0	0	1	1	0	0	1	1

Задача о рюкзаке (пример)

Пусть S=10 – грузоподъемность и

C	1	2	3	4
P	4	3	2	1
V	28	20	13	6

$$W_3(S) = \max_{0 \le X_3 \le \left[\frac{C}{2}\right]} \left\{ x_3 \cdot 13 + W_2(C - X_3 \cdot 2) \right\}, x_3 = 0,1,2,3,4,5$$

C	0-1	2	3	4	5	6	7	8	9	10
$W_3(C)$	0	13	20	28	33	41	48	56	61	69
x_3	0	1	0	0	1	1	0	0	1	1

$$W_4(S) = \max_{0 \le X_4 \le \left[\frac{C}{1}\right]} \left\{ x_4 \cdot 6 + W_3(C - X_4 \cdot 1) \right\}$$

Задача о рюкзаке (пример)

Пусть S=10 – грузоподъемность и

C	1	2	3	4
P	4	3	2	1
V	28	20	13	6

$$W_3(S) = \max_{0 \le X_3 \le \left[\frac{C}{2}\right]} \{x_3 \cdot 13 + W_2(C - X_3 \cdot 2)\}, x_3 = 0,1,2,3,4,5$$

C	0-1	2	3	4	5	6	7	8	9	10
$W_3(C)$	0	13	20	28	33	41	48	56	61	69
x_3	0	1	0	0	1	1	0	0	1	1

$$W_4(S) = \max_{0 \le X_4 \le \left[\frac{C}{1}\right]} \{x_4 \cdot 6 + W_3(C - X_4 \cdot 1)\}$$

C	0	1	2	3	4	5	6	7	8	9	10
$W_4(C)$	0	6	13	20	28	34	41	48	56	62	69
x_4	0	1	0	0	0	1	0	0	0	1	0

Задача о рюкзаке (пример)

Пусть S=10 – грузоподъемность и

C	1	2	3	4
P	4	3	2	1
V	28	20	13	6

$$W_3(S) = \max_{0 \le X_3 \le \left[\frac{C}{2}\right]} \{x_3 \cdot 13 + W_2(C - X_3 \cdot 2)\}, x_3 = 0,1,2,3,4,5$$

C	0-1	2	3	4	5	6	7	8	9	10
$W_3(C)$	0	13	20	28	33	41	48	56	61	69
x_3	0	1	0	0	1	1	0	0	1	1

$$W_4(S) = \max_{0 \le X_4 \le \left[\frac{C}{1}\right]} \{x_4 \cdot 6 + W_3(C - X_4 \cdot 1)\}$$

C	0	1	2	3	4	5	6	7	8	9	10
$W_4(C)$	0	6	13	20	28	34	41	48	56	62	69
x_4	0	1	0	0	0	1	0	0	0	1	0

$x_4 = 0 : W_4(10)$ при $x_4^0 = 0$

$\Rightarrow W_3(10-0) = W_3(10) = 69$ при $x_3^0 = 1$

$\Rightarrow W_2(10 - 2 \cdot 1) = W_2(8) = 56$ при $x_2^0 = 0$

$\Rightarrow W_1(8-0) = W_1(8) = 56$ при $x_1^0 = 2$

$\Rightarrow x^0 = (2,0,1,0)$

Кудряшов М.А.

Слайд 21,22

$$W_k^*(S_{k-1}) = \max_{\{X_{n-1}\}} \left\{ f_k(S_{k-1}, X_k) + W_{k+1}^*(S_k) \right\} \qquad (6)$$

Должно быть X_k

Слайд 23

$$W_k^*(S_{k-1}) = \max_{\{X_{n-1}\}} \left\{ f_k(S_{k-1}, X_k) + W_{k+1}^*(S_k) \right\}, k = n-1, n-2, \ldots, 1$$

Должно быть X_k

Кобыльникова А.О.

Слайд 28

1 шаг состоит в оценки эффективности выделения ресурса на 1-ое предприятие (первое направление);

На одно предприятие, в частном случае на первое предприятие

Слайд 30

Для применения схемы ДП погружают данную задачу в семейство задач с любым числом шагов k≤n и любым запасом ресурса $X_C \le S$.

Должно быть С

Слайд 30

Пусть $W_i(C)$ - максимальный доход при распределении объема С ресурса между i предприятиями, $i = \overline{1, n-1}$

До n

должно быть W_n

Слайд 30

Значение функции $\varphi_n(C)$ вычисляется лишь для данного значения C=S:

$$W_n(S) = \max_{0\le X_{ni}\le S}\{f_n(x_n)+\varphi_{n-1}(C-x_n)\}$$

Должно быть $W_n(S) = \max_{0\le X_n\le S}\{f_n(x_n) + W_{n-1}(S - x_n)\}$

Слайд 33-50 Желательно, чтобы вместо φ было W

Слайд 33

1. $\varphi_1(x) = \max_{0\le x_1\le x}\{f_1(x_1)\}$

$$\varphi_1(x1) = \max_{0\le x_1\le C}\{f_1(x_1)\}$$

Слайд 34-38

2. $\varphi_2(x) = \max_{0\le x2\le x}\{f_2(x_2)+\varphi_1(x-x_2)\}$

$$\varphi_2(\mathrm{C}) = \max_{0\le x_2\le C}\{f_2(x_2) + \varphi_1(C - x_2)\}$$

Слайд 39-43

3. $\varphi_3(x) = \max_{0\le x3\le x}\{f_3(x_3)+\varphi_2(x-x_3)\}$

$$\varphi_3(\mathrm{C}) = \max_{0\le x_3\le C}\{f_3(x_3) + \varphi_2(C - x_3)\}$$

Слайд 44-50

4 $\varphi_4(x) = \max\limits_{0 \le x4 \le x} \{f_4(x_4) + \varphi_4(x - x_4)\}$

$$\varphi_4(\mathrm{C}) = \max_{0 \le x_4 \le C} \{f_4(x_4) + \varphi_3(C - x_4)\}$$

Бектев Данил:

Слайд 50

4-му – X_4^0**=** 1 ⇒ *S=S-1=3* (осталось 3) ⇒ ***φ₃(3)*** при ***X₃⁰=1*** ⇒ 3-му – 1 ⇒ *S=S-1=2* (осталось 2) ⇒ ***φ₃(2)*** при ***X₂⁰=0*** ⇒ *S=S-1=2* (осталось 2) ⇒ ***φ₁(2)*** при ***X₁⁰=2***

$X^0 = (2,0,1,1)$ Прибыль=$8+5+7=20$

ОШИБКА: $\varphi_3(2)$ при $X_2^0 = 0 \quad => S = S - 1 = 2$

Правильно: $\varphi_2(2)$ при $X_2^0 = 0 \quad => S = S - 0 = 2$

Слайд 51

ОШИБКА: X_i − число предметов i −
ого типа, которые будут загружаться на транспортировочное средство

Правильно: X_i − число предметов i − ого типа, которое мы будем брать

Слайд 60-66

ОШИБКИ: $W_2(S) = \cdots;$

$W_3(S) = \cdots;$

$W_4(S) = \max\{\, x_4 * 6 + W_3(C - X_4 * 1)\}$

Правильно: $W_2(\mathrm{C}) = \cdots;$

$W_3(\mathrm{C}) = \cdots;$

$W_4(S) = \max\{\, x_4 * 6 + W_3(S - X_4 * 1)\}$

Методы оптимизации

Нелинейное программирование

Д.В. Домашова

Общая задача НЛП

$$f(x) \to \min \qquad (1)$$

$$g_i(x) \le 0,\ i = \overline{1,m} \qquad (2)$$

$$x \in X \subset R^n \qquad (3)$$

где $f(x),\ g_i(x),\ i = \overline{1,m}$ - нелинейные функции.

Решение: $x^* = \arg\min\limits_{x \in D} f(x)$

Графический метод решения задач нелинейного программирования для функций двух переменных

$$f(x_1, x_2) \to \min$$

$$g_i(x_1, x_2) \le 0, \quad i = \overline{1, m}$$

Функция f(x) называется целевой функцией, а неравенства gj(x)≤0, j =1,...,m называются ограничениями задачи. Множество точек, удовлетворяющих ограничениям задачи, называется допустимым множеством задачи.

Решить задачу нелинейного программирования графически — значит найти такую точку из допустимого множества, через которую проходит линия уровня f(x1,x2) = C, имеющая максимальное значение величины C из всех линий уровня, проходящих через допустимые точки задачи.

Этапы графического решения задач нелинейного программирования

Этап 1. Строится допустимое множество задачи. На плоскости наносятся геометрические места точек, соответствующих каждому уравнению из ограничений задачи $g_j(x_1,x_2)\leq 0$, $j = 1, \ldots, m$. Если допустимое множество задачи пусто, то задача не имеет решения.

Этапы графического решения задач нелинейного программирования

Этап 1. Строится допустимое множество задачи. На плоскости наносятся геометрические места точек, соответствующих каждому уравнению из ограничений задачи $gj(x1,x2) \le 0$, $j = 1, ..., m$. Если допустимое множество задачи пусто, то задача не имеет решения.

Этап 2. Строятся линии уровня целевой функции $f(x1,x2) = C$ при различных значениях параметра С.

Этапы графического решения задач нелинейного программирования

Этап 1. Строится допустимое множество задачи. На плоскости наносятся геометрические места точек, соответствующих каждому уравнению из ограничений задачи $gj(x1,x2) \leq 0$, $j = 1, \ldots, m$. Если допустимое множество задачи пусто, то задача не имеет решения.

Этап 2. Строятся линии уровня целевой функции $f(x1,x2) = C$ при различных значениях параметра С.

Этап 3. Определяется направление возрастания (для задачи максимизации) или убывания (для задачи минимизации) линий уровня целевой функции.

Этапы графического решения задач нелинейного программирования

Этап 1. Строится допустимое множество задачи. На плоскости наносятся геометрические места точек, соответствующих каждому уравнению из ограничений задачи $gj(x1,x2) \leq 0$, $j = 1, \ldots, m$. Если допустимое множество задачи пусто, то задача не имеет решения.

Этап 2. Строятся линии уровня целевой функции $f(x1,x2) = C$ при различных значениях параметра С.

Этап 3. Определяется направление возрастания (для задачи максимизации) или убывания (для задачи минимизации) линий уровня целевой функции.

Этап 4. Определяется точка допустимого множества, через которую проходит линия уровня с максимальным (для задачи максимизации) или минимальным (для задачи минимизации) значением параметра С. Если целевая функция не ограничена сверху (для задачи максимизации) или не ограничена снизу (для задачи минимизации) на допустимом множестве, то задача не имеет решения.

Этапы графического решения задач нелинейного программирования

Этап 1. Строится допустимое множество задачи. На плоскости наносятся геометрические места точек, соответствующих каждому уравнению из ограничений задачи $gj(x1,x2) \leq 0$, $j = 1, \ldots, m$. Если допустимое множество задачи пусто, то задача не имеет решения.

Этап 2. Строятся линии уровня целевой функции $f(x1,x2) = C$ при различных значениях параметра С.

Этап 3. Определяется направление возрастания (для задачи максимизации) или убывания (для задачи минимизации) линий уровня целевой функции.

Этап 4. Определяется точка допустимого множества, через которую проходит линия уровня с максимальным (для задачи максимизации) или минимальным (для задачи минимизации) значением параметра С. Если целевая функция не ограничена сверху (для задачи максимизации) или не ограничена снизу (для задачи минимизации) на допустимом множестве, то задача не имеет решения.

Этап 5. Для найденной точки определяют ее координаты x=(x1,x2) и значение целевой функции в данной точке

Графический метод решения задач нелинейного программирования для функций двух переменных

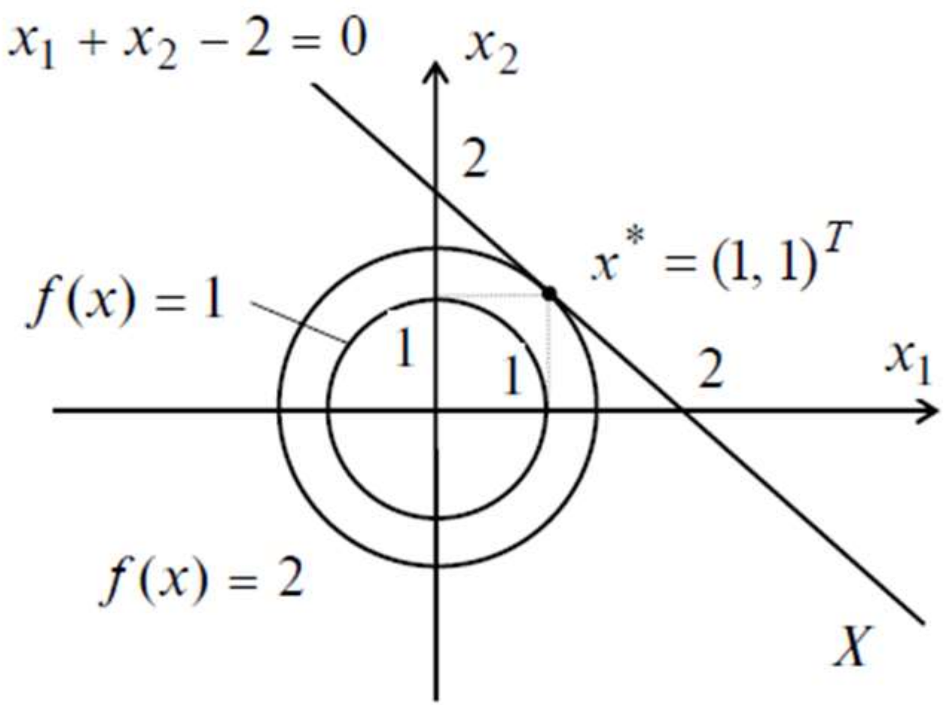
$$f(x_1, x_2) = x_1^2 + x_2^2 \to \min$$

$$x_1 + x_2 - 2 = 0$$

Графический метод решения задач нелинейного программирования для функций двух переменных

$$f(x_1, x_2) = x_1^2 + x_2^2 \to \min$$

$$x_1 + x_2 - 2 = 0$$



Графический метод решения задач нелинейного программирования для функций двух переменных

Пример. Найти оптимальное решение задачи

$$f = (x_1 - 2)^2 + (x_2 - 3)^2 \to \min/\max,$$

$$\begin{cases} x_1 + 2x_2 \leq 12 \\ x_1 + x_2 \leq 9, \end{cases}$$

$$x_1, x_2 \geq 0.$$

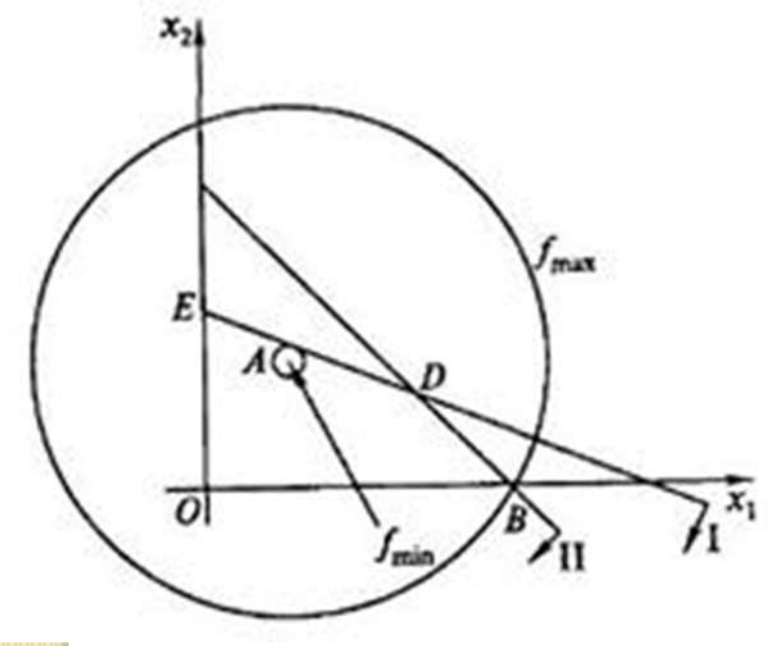
Графический метод решения задач нелинейного программирования для функций двух переменных

Пример. Найти оптимальное решение задачи

$$f=(x_1-2)^2+(x_2-3)^2 \to \min/\max,$$

$$\begin{cases} x_1+2x_2 \le 12 \\ x_1+x_2 \le 9, \end{cases}$$

$$x_1, x_2 \ge 0.$$



Решение. Множество допустимых решений задачи — четырехугольник OEDB.
Линии одного уровня функции цели - концентрические окружности с центром в точке A(2, 3) - точка внутри области.
В этой же точке достигается минимум функции цели, равный fmin=0.
Максимальное значение функции достигается в точке B(9, 0): fmax=58.

Графический метод решения задач нелинейного программирования для функций двух переменных

- Вывод:
- Наиболее существенное отличие задачи нелинейного программирования от линейных задач заключается в том, что оптимальное решение может находиться как на границе допустимого множества, так и являться его внутренней точкой.

Классическая задача на условный экстремум
Постановка задачи

$$f(x) \to \min \qquad (4)$$

$$g_i(x) = 0,\ i = \overline{1,m} \qquad (5)$$

Эта задача – частный случай задачи (1-3), т.к. каждое $g_i(x) = 0$ можно заменить двумя неравенствами:

$g_i(x) \le 0,\ \ g_i(x) \ge 0,\ i = \overline{1,m}$.

В свою очередь ограничения (2) можно представить $g_i(x) + Z_i^2 = 0,\ i = \overline{1,m}$.

Классическая задача на условный экстремум
Метод множителей Лагранжа

Задача условной оптимизации сводится к задаче безусловной оптимизации функции Лагранжа, составленной для этой задачи.

Обобщенная функция Лагранжа для задачи (4-5):

$$L(x, y_0, y) = y_0 f(x) + \sum_{i=1}^{m} y_i g_i(x), \quad y_i \in R, \; i = \overline{1, m+1}$$

y_i – множители Лагранжа.

Классическая функция Лагранжа для задачи (4-5):

$$L(x, y) = f(x) + \sum_{i=1}^{m} y_i g_i(x), \quad y_i \in R, \; i = \overline{1, m}$$

Классическая задача на условный экстремум
Метод множителей Лагранжа

Теорема 1 (Необходимое условие оптимальности)

Пусть функции f, $g_1,\ldots,g_m$ – непрерывно дифференцируемые в некоторой окрестности т. $x^* \in R^n$. Если x^* - точка локального экстремума задачи (4-5), то $\exists$ числа $y_0^*, y_1^*, \ldots, y_m^*$, не равные нулю одновременно и такие, что выполнены следующие условия:

1) условие стационарности обобщенной функции Лагранжа по x:

$$\frac{\partial L}{\partial x_j}\left(x^*, y_0^*, y^*\right) = 0, \quad j = \overline{1,n}$$

2) условие допустимости решения:

$$g_i\left(x^*\right) = 0, \ i = \overline{1,m}$$

Если при этом градиенты $g_i'\left(x^*\right)$, $i = \overline{1,m}$ в точке x^* линейно независимы (выполняется условие регулярности), то $y_0^* \neq 0$

Классическая задача на условный экстремум

Метод множителей Лагранжа

Замечания

1) При решении задач проверка условия регулярности затруднена, так как точка x_* заранее неизвестна.

Поэтому, как правило, рассматриваются два случая: $y_0^* \neq 0$; $y_0^* = 0$

Если выполняется условие регулярности, то берут

$$y_0^* = 1, \ y_i^* = \frac{y_i^*}{y_0^*}, \ i = \overline{1, m}$$

и точку x^* называют регулярной.

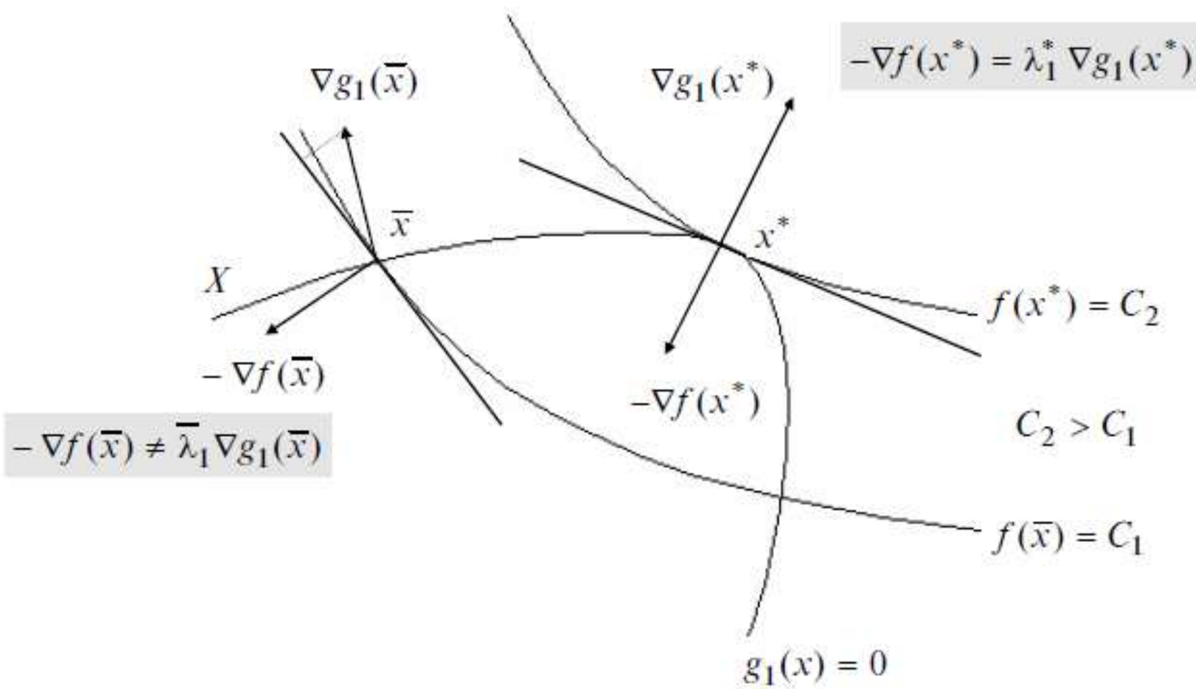
Классическая задача на условный экстремум

Метод множителей Лагранжа

Замечания

2) Необходимые условия оптимальности означают, что градиенты $f'(x^*),\ g_1'(x^*),\dots,\ g_m'(x^*)$ - линейно зависимы.

Если m=1 $\Rightarrow$ $f'(x^*)$ и $g_1'(x^*)$ - коллинеарны.



Точка x∗ условного экстремума (максимума) является точкой касания линии уровня целевой функции и кривой, описывающей ограничение. В точке $\bar{x}$ возможно движение вдоль ограничения, связанное с увеличением функции.

Классическая задача на условный экстремум
Метод множителей Лагранжа
Замечания

3) условия $g_i\left(x^*\right)=0,\ i=\overline{1,m}$ эквивалентны условиям

$$\frac{\partial L\left(x^*, y_0^*, y^*\right)}{\partial y_i}=g_i\left(x^*\right)=0$$

Классическая задача на условный экстремум
Метод множителей Лагранжа

Теорема 2 (Необходимое условие экстремума 2-го порядка)

Пусть функции $f(x), g_1(x), \ldots, g_m(x)$ - дважды непрерывно дифференцируемы в точке $x^* \in R^n$ и непрерывно дифференцируемы в ее некоторой окрестности, причем градиенты $g_1'(x^*), \ldots, g_m'(x^*)$ - линейно независимы.

Если x^* - локальное решение задачи (4-5) (регулярная точка экстремума), то $< L''_{xx}(x^*, y^*)h,\ h >\geq 0 \quad \forall\ y^*$, удовлетворяющих

$$L'_x(x^*, y^*) = 0,\ g_i(x^*) = 0,\ i = \overline{1,m} \text{ и } \forall\ h \in R^n:$$

$$< g_i'(x^*), h >= 0,\ i = \overline{1,m}$$

Классическая задача на условный экстремум
Метод множителей Лагранжа

Теорема 3 (Достаточные условия экстремума)

Пусть функции $f, g_1, \dots, g_m$ дважды дифференцируемы в точке $x^* \in R^n$, удовлетворяющей условиям допустимости $g_i(x^*) = 0,\ i = \overline{1, m}$ и $\exists\ y^*$:

1) $L'_x(x^*, y^*) = 0$

2) $< L''_{xx}(x^*, y^*)h,\ h > > 0$ при $\forall\ h \neq 0 \in R^n$ для которых

$< g'_i(x^*), h >= 0,\ i = \overline{1, m}$

то x^* - строгое локальное решение задачи (4-5).

Классическая задача на условный экстремум
Пример

$$f(x_1, x_2) = x_1^2 + x_2^2 \to \min$$

$$x_1 + x_2 - 2 = 0$$

Классическая задача на условный экстремум
Пример

$$f(x_1, x_2) = x_1^2 + x_2^2 \to \min$$

$$x_1 + x_2 - 2 = 0$$

Проверим условие регулярности.

Так как $\nabla g1(1,1)^T \neq 0$, то условие выполняется

Поэтому будем пользоваться классической функцией Лагранжа

Классическая задача на условный экстремум
Пример

$$f(x_1, x_2) = x_1^2 + x_2^2 \to \min$$

$$x_1 + x_2 - 2 = 0$$

$$L(x_1, x_2, y) = f(x_1, x_2) + yg_1(x_1, x_2)$$

Классическая задача на условный экстремум

Пример

$$f(x_1, x_2) = x_1^2 + x_2^2 \to \min$$

$$x_1 + x_2 - 2 = 0$$

$$L(x_1, x_2, y) = f(x_1, x_2) + y g_1(x_1, x_2)$$

$$L(x_1, x_2, y) = x_1^2 + x_2^2 + y(x_1 + x_2 - 2)$$

Классическая задача на условный экстремум
Пример

$$f\left(x_1, x_2\right) = x_1^2 + x_2^2 \rightarrow \min$$

$$x_1 + x_2 - 2 = 0$$

$$L\left(x_1, x_2, y\right) = f\left(x_1, x_2\right) + y g_1\left(x_1, x_2\right)$$

$$L\left(x_1, x_2, y\right) = x_1^2 + x_2^2 + y(x_1 + x_2 - 2)$$

$$\frac{\partial L(x_1, x_2, y)}{\partial x_1} = 2x_1 + y = 0$$

$$\frac{\partial L(x_1, x_2, y)}{\partial x_2} = 2x_2 + y = 0$$

Классическая задача на условный экстремум
Пример

$$f\left(x_1, x_2\right) = x_1^2 + x_2^2 \to \min$$

$$x_1 + x_2 - 2 = 0$$

$$L\left(x_1, x_2, y\right) = f\left(x_1, x_2\right) + y g_1\left(x_1, x_2\right)$$

$$L\left(x_1, x_2, y\right) = x_1^2 + x_2^2 + y(x_1 + x_2 - 2)$$

$$\frac{\partial L(x_1, x_2, y)}{\partial x_1} = 2x_1 + y = 0$$

$$\frac{\partial L(x_1, x_2, y)}{\partial x_2} = 2x_2 + y = 0$$

$$\frac{\partial L(x_1, x_2, y)}{\partial y} = x_1 + x_2 - 2 = 0$$

Классическая задача на условный экстремум
Пример

$$f(x_1, x_2) = x_1^2 + x_2^2 \to \min$$

$$x_1 + x_2 - 2 = 0$$

$$L(x_1, x_2, y) = f(x_1, x_2) + y g_1(x_1, x_2)$$

$$L(x_1, x_2, y) = x_1^2 + x_2^2 + y(x_1 + x_2 - 2)$$

$$\frac{\partial L(x_1, x_2, y)}{\partial x_1} = 2x_1 + y = 0$$

$$\frac{\partial L(x_1, x_2, y)}{\partial x_2} = 2x_2 + y = 0$$

$$\frac{\partial L(x_1, x_2, y)}{\partial y} = x_1 + x_2 - 2 = 0$$

$$x^* = (1,1)$$

$$y^* = -1$$

Классическая задача на условный экстремум
Пример

$$f\left(x_1, x_2\right) = x_1^2 + x_2^2 \to \min$$

$$x_1 + x_2 - 2 = 0$$

$$L\left(x_1, x_2, y\right) = f\left(x_1, x_2\right) + y g_1\left(x_1, x_2\right)$$

$$L\left(x_1, x_2, y\right) = x_1^2 + x_2^2 + y(x_1 + x_2 - 2)$$

$$\frac{\partial L(x_1, x_2, y)}{\partial x_1} = 2x_1 + y = 0$$

$$\frac{\partial L(x_1, x_2, y)}{\partial x_2} = 2x_2 + y = 0$$

$$\frac{\partial L(x_1, x_2, y)}{\partial y} = x_1 + x_2 - 2 = 0$$

$$x^* = (1,1)$$

$$y^* = -1$$

$$L''_{xx}\left(x^*, y^*\right) = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}$$

Классическая задача на условный экстремум
Пример

$$f\left(x_1, x_2\right) = x_1^2 + x_2^2 \to \min$$

$$x_1 + x_2 - 2 = 0$$

$$L\left(x_1, x_2, y\right) = f\left(x_1, x_2\right) + y g_1\left(x_1, x_2\right)$$

$$L\left(x_1, x_2, y\right) = x_1^2 + x_2^2 + y(x_1 + x_2 - 2)$$

$$\frac{\partial L(x_1, x_2, y)}{\partial x_1} = 2x_1 + y = 0$$

$$\frac{\partial L(x_1, x_2, y)}{\partial x_2} = 2x_2 + y = 0$$

$$\frac{\partial L(x_1, x_2, y)}{\partial y} = x_1 + x_2 - 2 = 0$$

$$x^* = (1,1)$$

$$y^* = -1$$

$$L''_{xx}\left(x^*, y^*\right) = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}$$

$\Delta_1 = 2 > 0, \Delta_2 = 4$

Выполнено достаточное условие экстремума, следовательно, точка $x^* = (1,1)$ – точка минимума

Классическая задача на условный экстремум
Пример 2

Пример 3.6. Найти условный экстремум в задаче

$$f(x) = x_1 + x_2 \to \text{extr},$$

$$g_1(x) = x_1^2 + x_2^2 - 2 = 0.$$

☐ Проверим условие регулярности. Так как $\nabla g_1(x) = (2x_1, 2x_2)^T \neq 0$ для всех $x \in X$, то условие выполняется (см. определение 3.6). Поэтому будем пользоваться классической функцией Лагранжа.

1. Составим функцию Лагранжа:

$$L(x, \lambda_1) = x_1 + x_2 + \lambda_1 (x_1^2 + x_2^2 - 2).$$

2. Выпишем необходимые условия экстремума первого порядка:

а) $$\frac{\partial L(x, \lambda_1)}{\partial x_1} = 1 + 2\lambda_1 x_1 = 0 \;\Rightarrow\; x_1 = -\frac{1}{2\lambda_1},$$

$$\frac{\partial L(x, \lambda_1)}{\partial x_2} = 1 + 2\lambda_1 x_2 = 0 \;\Rightarrow\; x_2 = -\frac{1}{2\lambda_1};$$

б) $g_1(x) = x_1^2 + x_2^2 - 2 = 0$.

3. Решением системы являются две условно-стационарные точки:

$$A\colon\ x_1^* = 1,\ \ x_2^* = 1,\ \ \lambda_1^* = -\frac{1}{2}; \qquad B\colon\ x_1^* = -1,\ \ x_2^* = -1,\ \ \lambda_1^* = \frac{1}{2}.$$

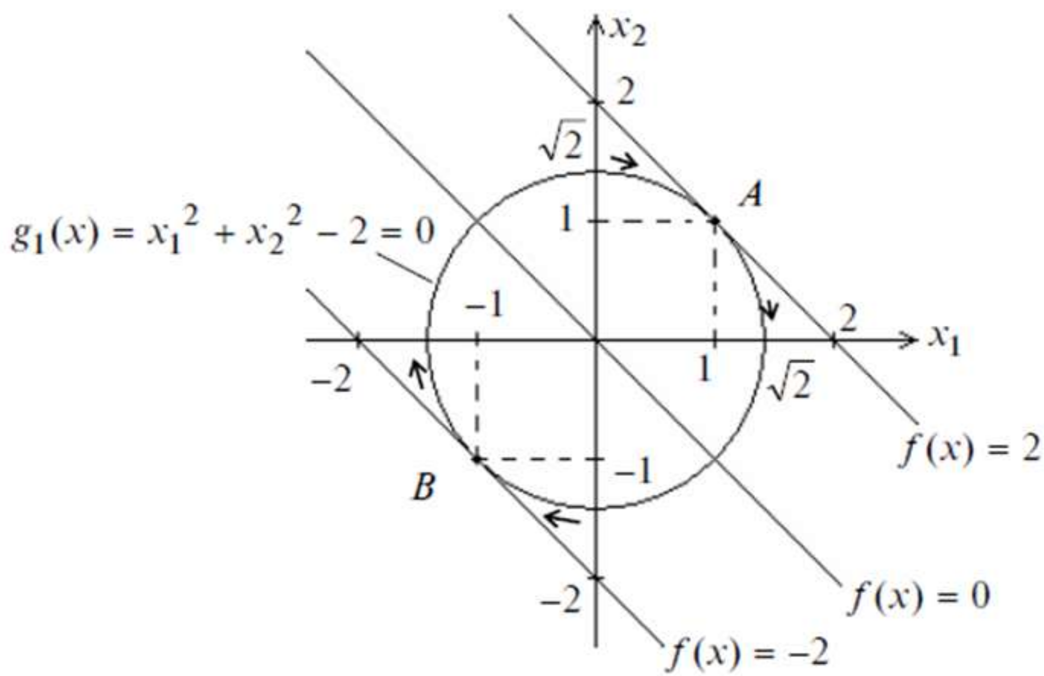
Классическая задача на условный экстремум
Пример 2

Проверим выполнение достаточных условий экстремума:

$$L''_{xx} = \begin{pmatrix} 2\lambda_1 & 0 \\ 0 & 2\lambda_1 \end{pmatrix}$$

$L''_{xx}(1{,}1) = \begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix}$ $\Delta_1 = -1 < 0,\ \Delta_2 = 1 > 0$ – матрица отрицательно определена, следовательно x_1=(1,1) – точка максимума

$L''_{xx}(-1,-1) = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ $\Delta_1 = 1 > 0,\ \Delta_2 = 1 > 0$ – матрица положительно определена, следовательно x_1=(-1,-1) – точка минимума



Общая задача НЛП

$$f(x) \rightarrow \min \qquad (1)$$

$$g_i(x) \leq 0,\ i = \overline{1,m} \qquad (2)$$

$$x \in X \subset R^n \qquad (3)$$

Общая задача НЛП

$$f(x) \to \min \qquad (1)$$

$$g_i(x) \le 0,\ i = \overline{1,m} \qquad (2)$$

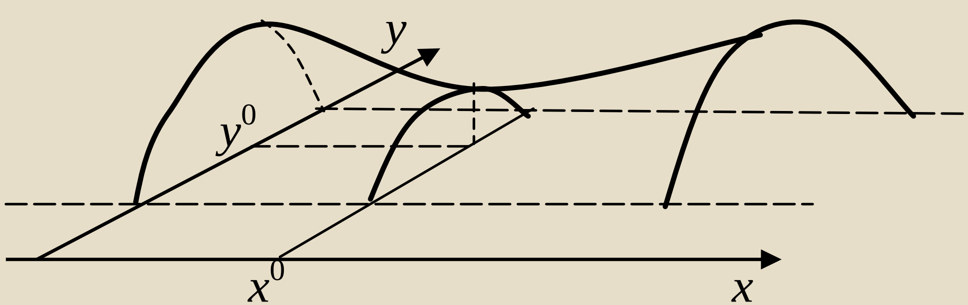
$$x \in X \subset R^n \qquad (3)$$

Классическая функция Лагранжа:

$$L(x,y) = f(x) + \sum_{i=1}^{m} y_i g_i(x)$$

Определение. Пара векторов $\left(x^0, y^0\right)$ является седловой точкой функции $L(x,y)$ в области $x \in X,\ \ y \ge 0$, если

$$L\left(x^0, y\right) \le L\left(x^0, y^0\right) \le L(x, y), \qquad \forall x \in X,\ \ y \ge 0$$



Общая задача НЛП

Теорема (достаточное условие локального экстремума). Пусть $\left(x^0, y^0\right)$ - седловая точка функции Лагранжа в области $x \in X,\ y \geq 0$, тогда x^0 – является решением задачи (1)-(3), причем, справедливо правило дополняющей нежесткости:

$$\sum_{i=1}^{m} y_i^0 g_i\left(x^0\right) = 0$$ (или $y_i^0 g_i\left(x^0\right) = 0,\quad i = \overline{1,m}$, т.к. $y_i^0 \geq 0,\ g_i(x) \leq 0$)

Общая задача НЛП

Теорема Куна-Таккера (необходимое и достаточное условия для задачи выпуклого НЛП)

Пусть $f(x),\ g_i(x),\ i=\overline{1,m}$ выпуклые функции, множество X – выпукло и выполняется условие регулярности Слейтера: $\exists \bar{x} \in X : g_i(\bar{x}) < 0,\ \forall i=\overline{1,m}$,то x^0 – решение задачи (1)-(3) тогда и только тогда, когда $\exists y^0 \geq 0 : \left(x^0, y^0\right)$ - седловая точка функции Лагранжа.

Общая задача НЛП

Теорема Куна-Таккера (обобщенное правило множителей Лагранжа).

Для того чтобы т.x^* была решением задачи (1)-(2) необходимо, а в том случае, когда (1)-(2) –задача выпуклого НЛП и область D удовлетворяет условию Слейтера $(\exists\tilde{x} : g_i(\tilde{x}) \le 0,\ i = \overline{1,m})$, то и достаточно, чтобы нашлись такие $y_i^* \ge 0,\ i = \overline{1,m}$, не равные нулю одновременно, что:

$$L'_x(x^*, y^*) = f'(x^*) + \sum_{i=1}^{m} y_i^* g'_i(x^*) = 0,$$

при этом выполняются условия дополняющей нежесткости:

$$y_i^* g_i(x^*) = 0,\ i = \overline{1,m}$$

Задача (1)-(2) называется задачей выпуклого НЛП, если

f(x) и g_i(x), $i = \overline{1,m}$ – выпуклые функции

Выпуклые множества

Пусть L – конечное линейное пространство. x^1, x^2 Є L.

Опр.: Множество $E \subset L$: $E = \left\{ x \in L : x = \lambda x^1 + (1-\lambda)x^2, \lambda \in [0,1] \in R \right\}$ называется выпуклым множеством, если для любых x^1, x^2 Є X => $\lambda x^1 + (1-\lambda)x^2$ є X, λ Є [0,1].

Выпуклые множества

Пусть L – конечное линейное пространство. x^1, x^2 Є L.

Опр.: Множество $E \subset L$: $E = \{x \in L : x = \lambda x^1 + (1-\lambda) x^2, \lambda \in [0,1] \in R\}$ называется выпуклым множеством, если для любых x^1, x^2 Є X => $\lambda x^1 + (1-\lambda) x^2$ є X, λ Є [0,1].

В пространстве R^n выпуклыми множествами являются: само пространство, любое его линейное подпространство, точка, шар, отрезок, а также прямая, проходящая через т. x° в направлении вектора h:

$$l\, x^\circ h = \{x \in R^n : x = x^\circ + \alpha h,\ \alpha \in R\}$$

луч, выходящий из т. x° в направлении вектора h:

$$l^+ x^\circ h = \{ x \in R^n : x = x^\circ + \alpha h,\ \alpha \geq 0\}$$

гиперплоскость с нормалью p:

$$Hp\beta = \{ x \in R^n : \langle p_1 x \rangle = \beta\}$$

порождаемые гиперплоскостью полупространства:

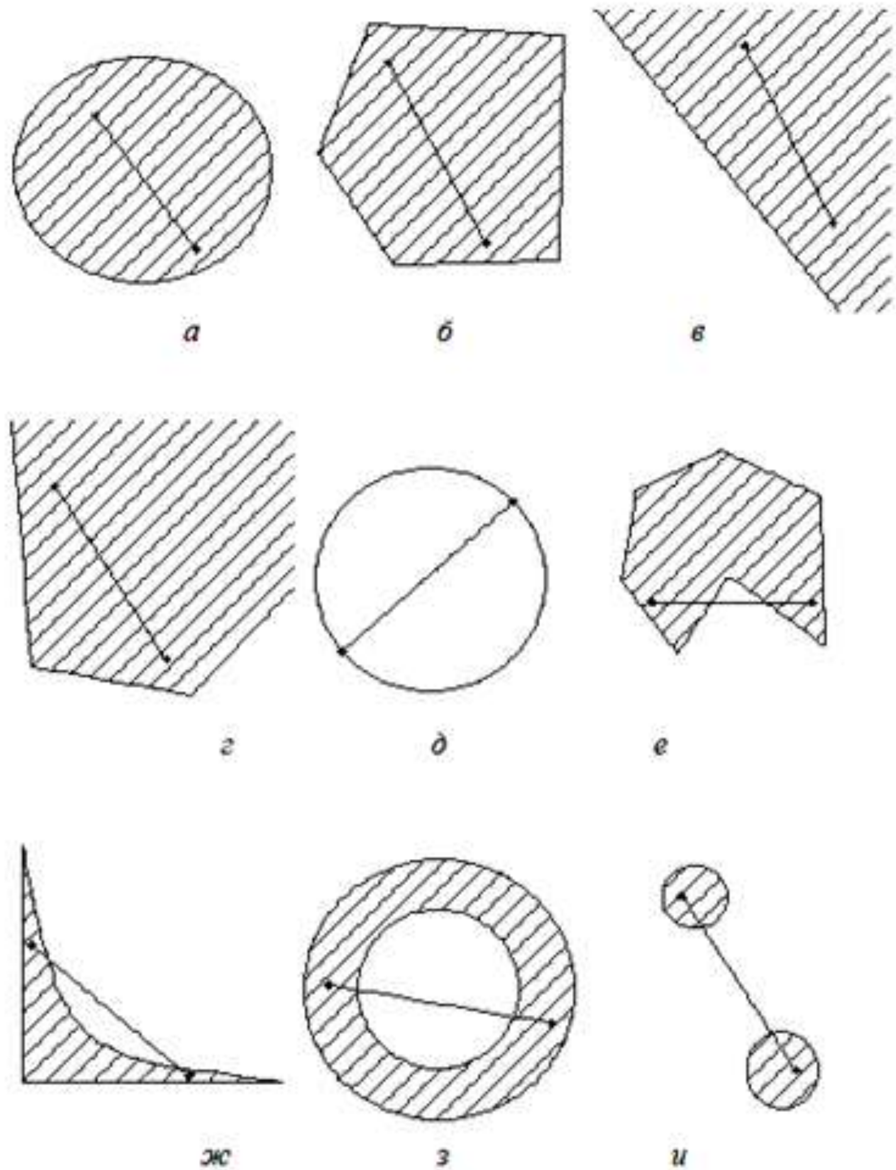
$$Hp\beta^+ = \{ x \in R^n : \langle p_1 x \rangle \geq \beta\}$$

$$Hp\beta^- = \{ x \in R^n : \langle p_1 x \rangle \leq \beta\}$$

Все перечисленные множества (кроме шара) являются частными случаями выпуклого множества вида:

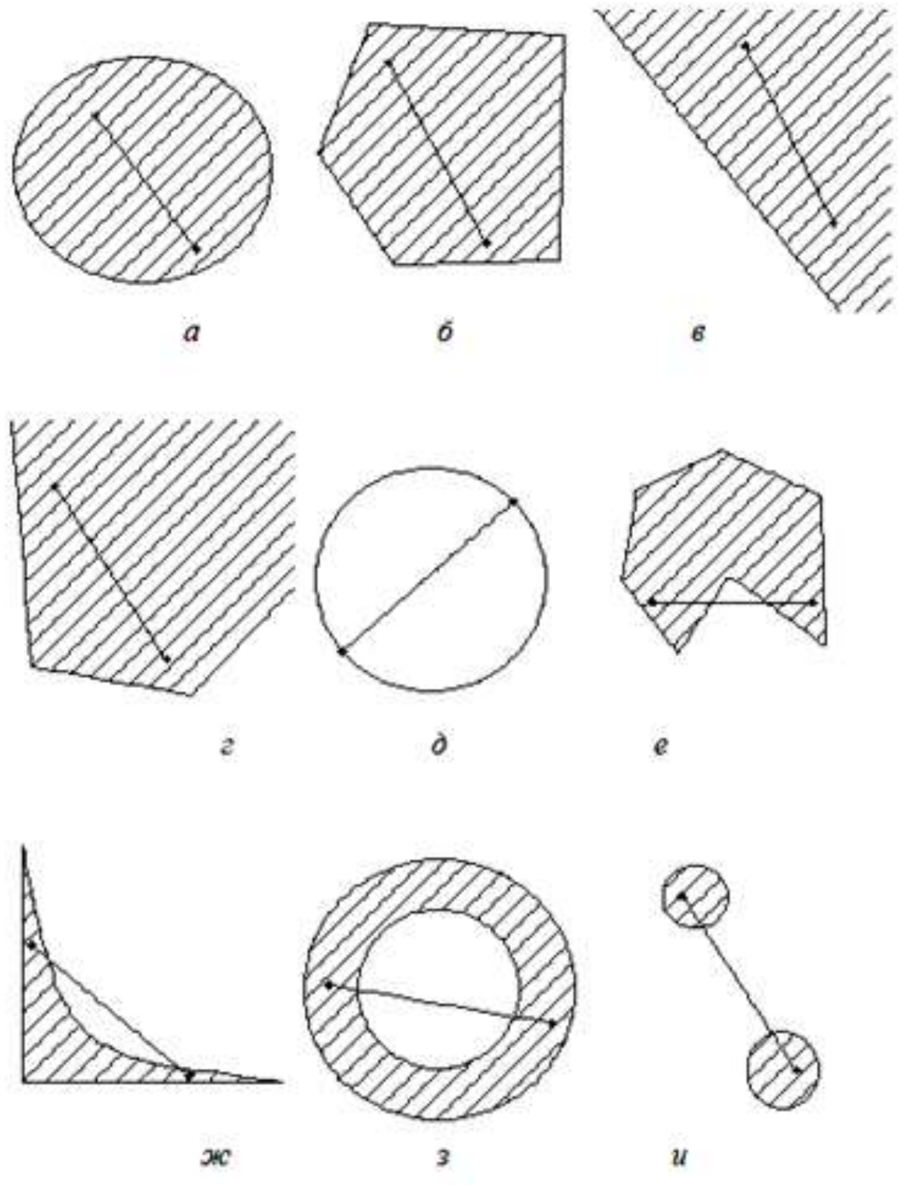
$$X = \{x \in R^n : Ax \leq b\} = \{ x \in R^n : \langle a_i, x \rangle \leq b_i\},\ i = \overline{1, m}$$

Выпуклые множества



Какие множества выпуклые?

Выпуклые множества



Выпуклые множества: а, б, в, г

Выпуклые функции

Определение. Функция f (x) , определенная на выпуклом множестве X, называется выпуклой, если
$f(\lambda x^1 + (1-\lambda)x^2) \le \lambda f(x^1) + (1-\lambda) f(x^2)\ \forall x^1, x^2 \in X,\ 0 \le \lambda \le 1$.

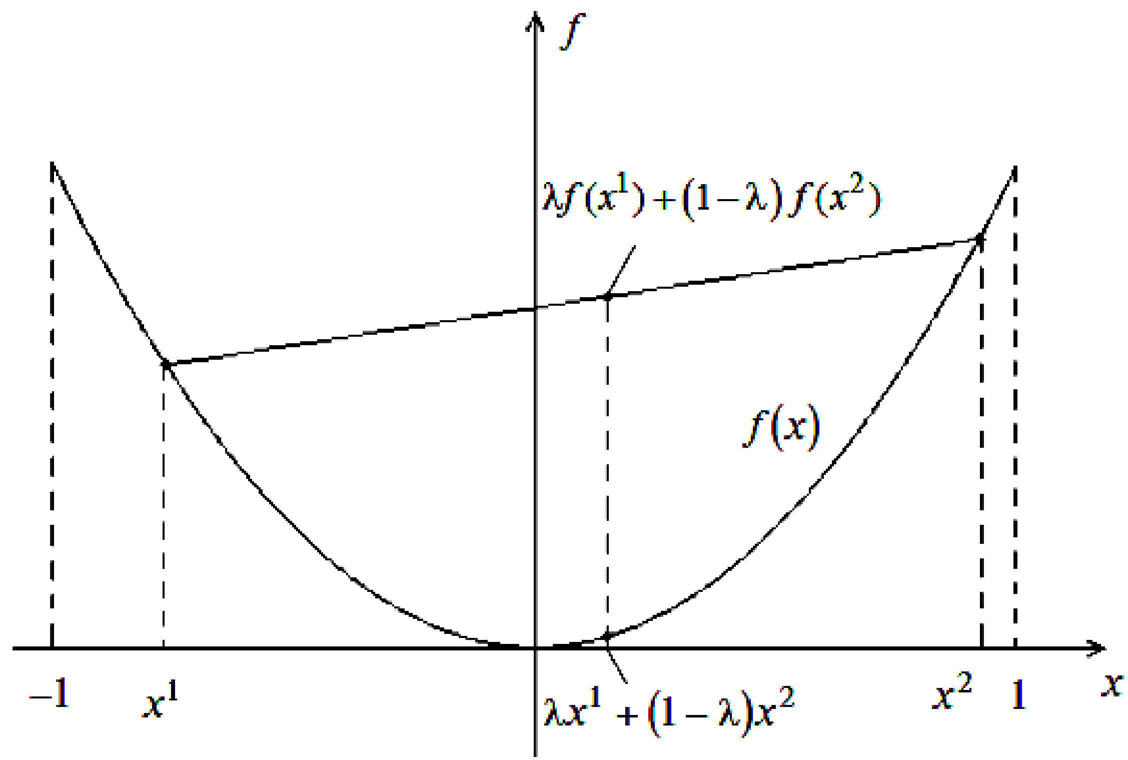
Определение. Функция *f* (*x*) , определенная на выпуклом множестве *X*, называется *строго выпуклой*, если
$f(\lambda x^1 + (1-\lambda)x^2) \le \lambda f(x^1) + (1-\lambda) f(x^2)\ \forall x^1, x^2 \in X,\ x1 \ne x2,\ 0 < \lambda < 1.$

Функцию *f* (*x*) называют *выпуклой*, если ее график целиком лежит не выше отрезка, соединяющего две произвольные точки графика.
Функцию называют *строго выпуклой*, если ее график целиком лежит ниже отрезка, соединяющего две произвольные, но не совпадающие точки графика.

Выпуклость функции можно определить по матрице Гессе:
- если $H(x) \ge 0\ \forall x \in R^n$, то функция выпуклая;
- если $H(x) > 0\ \forall x \in R^n$, то функция строго выпуклая.

Выпуклые функции

Пример. Дана функция $f(x) = x^2$. Исследуем ее на выпуклость на отрезке $X = \{x: -1 \le x \le 1\}$
Функция является строго выпуклой, так как ее график целиком лежит ниже отрезка, соединяющего две произвольные, но не совпадающие точки графика.
Более того, функция одновременно является строго выпуклой, так как выполняется условие $H(x) = f''(x) = 2 > 0$.



Выпуклые функции

1. Если f (x) выпуклая функция на выпуклом множестве X, то всякая точка локального минимума является точкой ее глобального минимума на X.

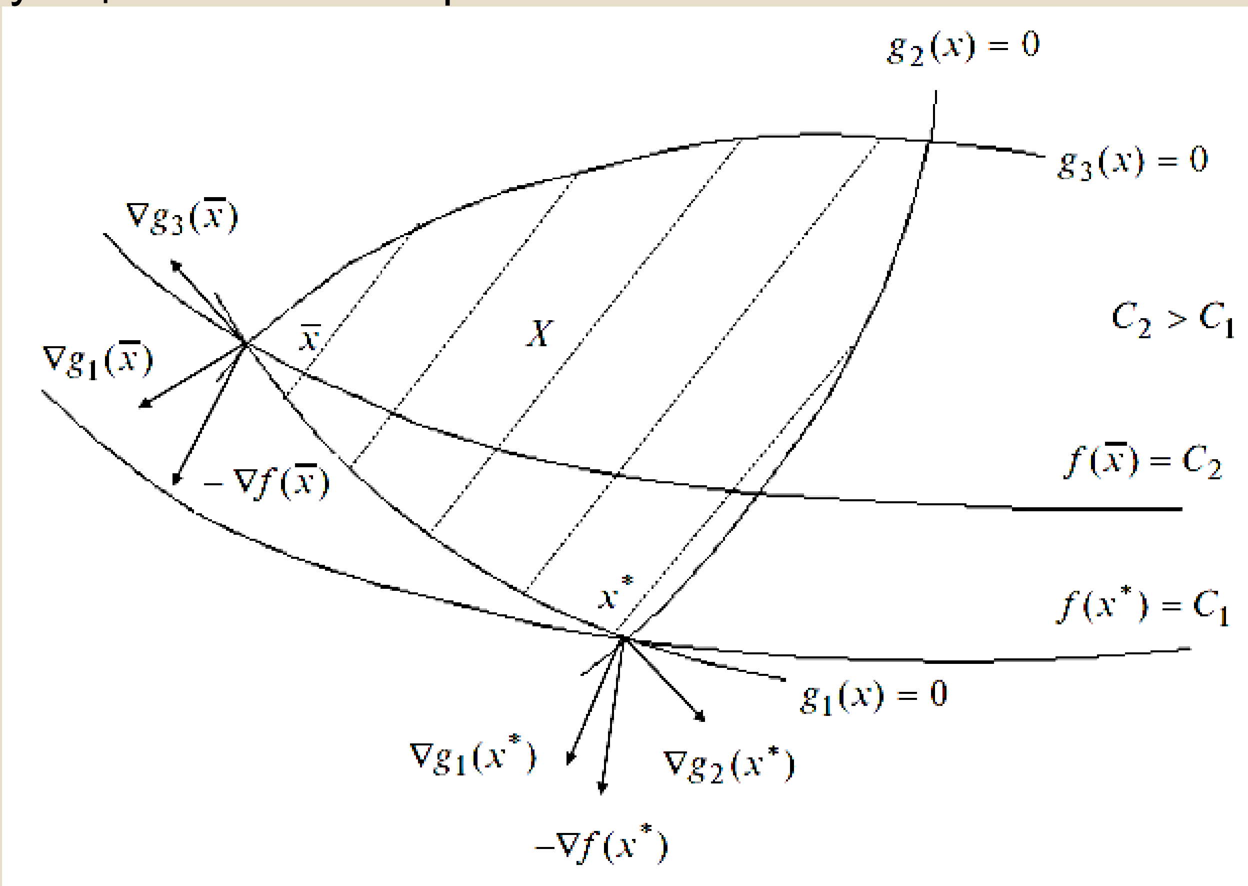
2. Если выпуклая функция достигает своего минимума в двух различных точках, то она достигает минимума во всех точках отрезка, соединяющего эти две точки.

3. Если f (x) строго выпуклая функция на выпуклом множестве X, то она может достигать своего глобального минимума на X не более чем в одной точке.

Общая задача НЛП

Замечание 1.

Геометрический смысл: антиградиент целевой функции в т.x^* является неотрицательной линейной комбинацией градиентов функций, образующих активные ограничения в т.x



Общая задача НЛП

Замечание 2.

$$y_i^* g_i\left(x^*\right) = 0,\ i = \overline{1,m}$$

Из условий дополняющей нежесткости следует,

что если ограничение в т.x^* - пассивное, т.е. $g_i\left(x^*\right) < 0$, то $y_i^* = 0$,

если ограничение в т.x^* - активное, т.е. $g_i\left(x^*\right) = 0$, то

$y_i^* \geq 0$ (min) $y_i^* \leq 0$ (max) .

Общая задача НЛП

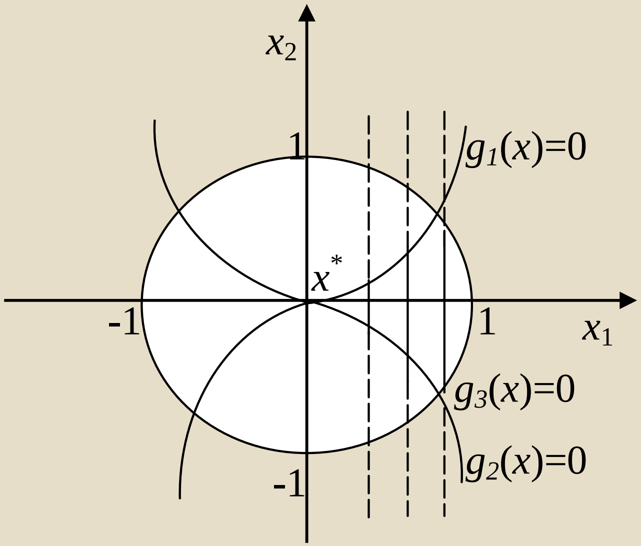
Замечание 3.

Случай нерегулярности $y_0^* = 0$ (отражает вырожденность ограничений).

Рассмотрим пример.

$$f(x) = x_1 \to \min$$

$$\begin{cases} -x_1^3 + x_2 \le 0 \\ -x_1^3 - x_2 \le 0 \\ x_1^2 + x_2^2 \le 1 \end{cases}$$



x^*=(0,0)

Активными в т. x^* являются $g_1(x)$ и $g_2(x)$, при этом $f'(x^*) = (1,0)$

$g_1'(x^*) = (0,1)$, $g_2'(x^*) = (0,-1)$, $g_1'(x) = (-3x_1^2, 1)$, $g_2'(x) = (-3x_1^2, -1)$

Видно, что $f'(x^*)$ нельзя представить в виде линейной комбинации векторов $g_1'(x^*)$ и $g_2'(x^*)$. Условия теоремы здесь может выполняться лишь при $y_0^* = 0,\ y_1^* = \lambda,\ y_2^* = -\lambda,$ где $\lambda > 0$.

Суть дела тут в том, что сами градиенты $g_1'(x^*)$ и $g_2'(x^*)$ линейно зависимы.

Общая задача НЛП

Теорема (достаточные условия экстремума второго порядка).

Пусть имеется точка (x^*, y^*), удовлетворяющая системе $\dfrac{\partial L\left(x^*, y_0^*, y^*\right)}{\partial x_j} = 0, \quad j = \overline{1,n}$ при $y_0^* \neq 0$.

Если в этой точке $< L''_{xx}\left(x^*, y^*\right)h,\ h >> 0$ ($< L''_{xx}\left(x^*, y^*\right)h,\ h >< 0$) для всех $\forall\ h \neq 0 \in R^n$ таких, что

$$< g_i'\left(x^*\right), h >= 0, i \in I(x^*) \quad y_i^* > 0 \; (y_i^* < 0)$$

$$< g_i'\left(x^*\right), h > \leq 0, i \in I(x^*) \,,\; y_i^* = 0$$

то точка x∗ является точкой локального минимума (максимума) в задаче (1-2).

Общая задача НЛП
Пример

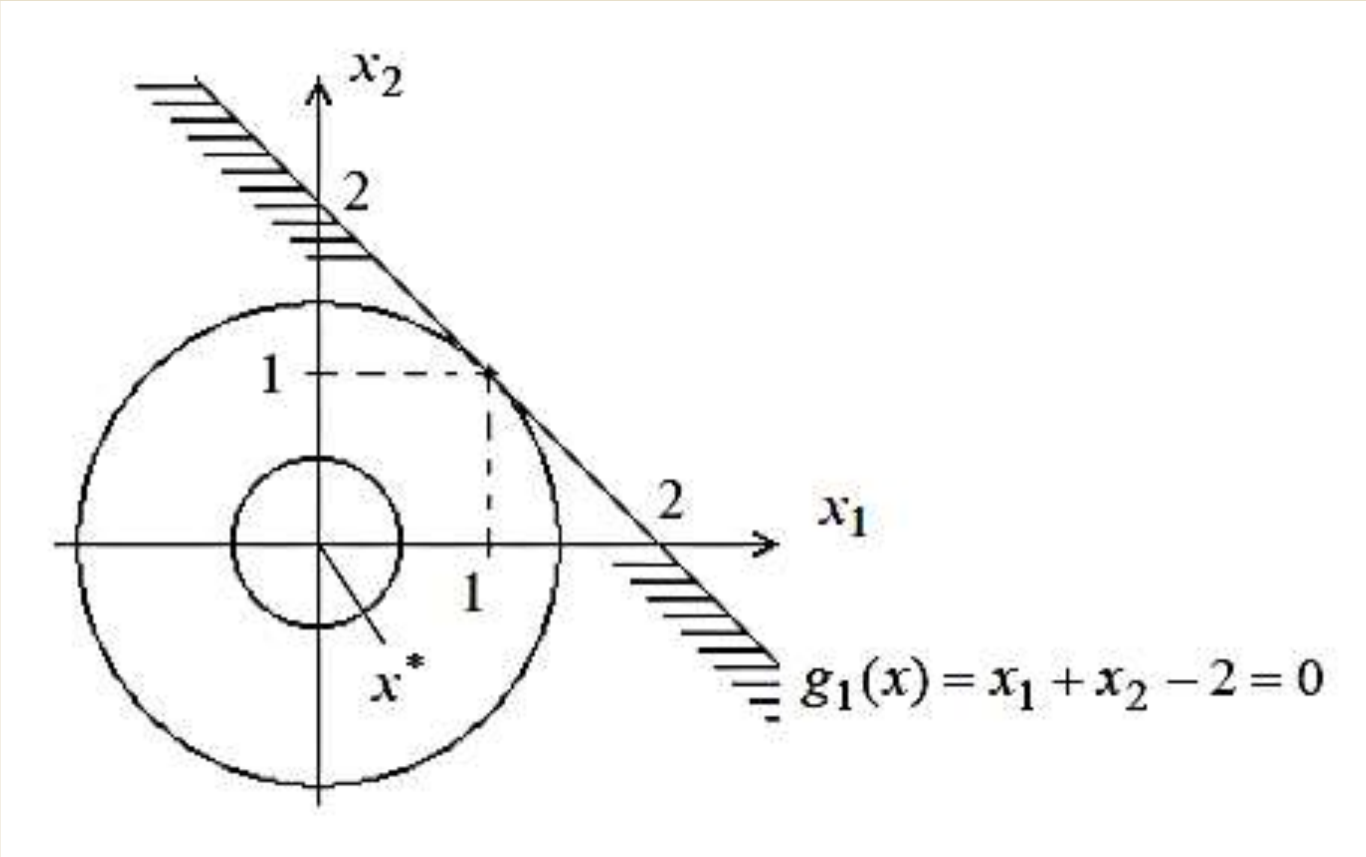
$$f\left(x_1, x_2\right) = x_1^2 + x_2^2 \to \min$$

$$x_1 + x_2 - 2 \leq 0$$

Общая задача НЛП
Пример

$$f(x_1, x_2) = x_1^2 + x_2^2 \to \min$$

$$x_1 + x_2 - 2 \le 0$$



Общая задача НЛП
Пример

$$L(x_1, x_2, y) = y_0 f(x_1, x_2) + y_1 g_1(x_1, x_2)$$

$$L(x_1, x_2, y) = y_0 x_1^2 + x_2^2 + y_1 (x_1 + x_2 - 2)$$

$$\frac{\partial L(x_1, x_2, y)}{\partial x_1} = 2y_0 x_1 + y_1 = 0$$

$$\frac{\partial L(x_1, x_2, y)}{\partial x_2} = 2y_0 x_2 + y_1 = 0$$

$$x_1 + x_2 - 2 \le 0$$

$y_1 \ge 0$ - для минимума, $y_1 \le 0$ - для максимума

$$y_1 (x_1 + x_2 - 2) = 0$$

Общая задача НЛП
Пример

Решим систему для двух случаев.

Первый случай: $y_0 = 0$. Тогда $y_1 = 0$, что противоречит требованию теоремы о существовании ненулевого вектора y/

Общая задача НЛП
Пример

Решим систему для двух случаев.

Первый случай: $y_0 = 0$. Тогда $y_1 = 0$, что противоречит требованию теоремы о существовании ненулевого вектора y/

Второй случай: $y_0 \neq 0$. Тогда заменим обобщенную функцию Лагранжа классической.

$$L(x_1, x_2, y) = x_1^2 + x_2^2 + y_1(x_1 + x_2 - 2)$$

$$\frac{\partial L(x_1, x_2, y)}{\partial x_1} = 2x_1 + y_1 = 0$$

$$\frac{\partial L(x_1, x_2, y)}{\partial x_2} = 2x_2 + y_1 = 0$$

$$x_1 + x_2 - 2 \leq 0$$

$y_1 \geq 0$ - для минимума, $y_1 \leq 0$ - для максимума

$$y_1(x_1 + x_2 - 2) = 0$$

Общая задача НЛП
Пример

Из условия дополняющей нежесткости:

1) $y_1 = 0$

Фактически решается задача поиска безусловного экстремума.

Решаем систему, получаем $x^* = (0{,}0)$.

Условие допустимости решения выполняется.

Выполняются необходимые условия и для минимума, и для максимума.

Общая задача НЛП
Пример

Из условия дополняющей нежесткости:

1) $y_1 = 0$

Фактически решается задача поиска безусловного экстремума.

Решаем систему, получаем $x^* = (0{,}0)$.

Условие допустимости решения выполняется.

Выполняются необходимые условия и для минимума, и для максимума.

2) $y_1 \neq 0$

Решаем систему, получаем $x^* = (1{,}1)$, $y^* = -2$.

Так как $y^* < 0$, то выполняется необходимое условие максимума. Таким образом, имеем две условно-стационарные точки.

Общая задача НЛП
Пример

Проверим достаточные условия экстремума.

$$L''_{xx}\left(x^*, y^*\right) = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}$$

$\Delta_1 = 2 > 0, \Delta_2 = 4 > 0,\ y_1 = 0$

Выполнено достаточное условие экстремума, следовательно, точка $x^* = (0{,}0)$ – точка минимума.

Общая задача НЛП
Пример

Проверим достаточные условия экстремума.

$$L''_{xx}\left(x^*, y^*\right) = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}$$

$\Delta_1 = 2 > 0, \Delta_2 = 4 > 0,\ y_1 = 0$

Выполнено достаточное условие экстремума, следовательно, точка $x^* = (0,0)$ – точка минимума.

Так как для точки $x^* = (1,1)$ $y^* = -2$, то достаточные условия для максимума не выполнены (матрица в стационарной точке не является отрицательно определенной).

Общая задача НЛП
Пример

Проверим достаточные условия экстремума.

$$L''_{xx}\left(x^*, y^*\right) = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}$$

$\Delta_1 = 2 > 0, \Delta_2 = 4 > 0,\ y_1 = 0$

Выполнено достаточное условие экстремума, следовательно, точка $x^* = (0,0)$ – точка минимума.

Так как для точки $x^* = (1,1)\quad y^* = -2$, то достаточные условия для максимума не выполнены (матрица в стационарной точке не является отрицательно определенной).

С другой стороны, функция *f* (*x*) выпуклая и множество допустимых решений также выпуклое. Поэтому в точке $x^* = (0,0)$ достигается глобальный условный минимум, а достаточные условия можно было и не проверять. Вычислим значение функции в точке условного минимума: f (x*) = 0.

Общая задача НЛП

Рассмотрим задачу.

$$f(x) \to \min \qquad (1)$$

$$g_i(x) \le 0,\ i = \overline{1,m} \qquad (2)$$

$$x_j \ge 0,\ \ j = \overline{1,n} \qquad (3)$$

Теорема (Куна-Таккера)

Если $f(x),\ g_i(x)$ - выпуклы, множество *D* регулярно по Слейтеру, то $x^* \in D$ - оптимальное решение задачи тогда и только тогда, когда $\exists y_i^* \ge 0,\ i = \overline{1,m},\ V_j^* \ge 0,\ \ j = \overline{1,n}$

$$f'(x^*) + \sum_{i=1}^{m} y_i^* g_i'(x^*) - \sum_{j=1}^{n} V_j^* l_j = 0,$$

при этом выполнены условия дополняющей нежесткости:

$$y_i^* g_i(x^*) = 0,\ i = \overline{1,m}, \qquad V_j^* x_j^* = 0,\ \ j = \overline{1,n},$$

где $l_j,\ \ j = \overline{1,n}$ - *j*-ый координатный вектор l_j=(0,…,1,…,0)

Общая задача НЛП
Условия существования седловой точки

Пусть $L(x,y)$ выпукла по x для $\forall x \geq 0$, вогнута по y $\forall y \geq 0$, непрерывно дифференцируема по x и y.

Теорема. Пара (x^*, y^*), $x^* \geq 0,\ y^* \geq 0$ - седловая точка функции Лагранжа $L(x,y)$ в области $x \geq 0,\ y \geq 0$, тогда и только тогда, когда

$$\frac{\partial L^*}{\partial x_j} \geq 0, \quad x_j^* \frac{\partial L^*}{\partial x_j} = 0, \quad x_j^* \geq 0, \quad j = \overline{1,n}$$

$$\frac{\partial L^*}{\partial y_i} \leq 0, \quad y_i^* \frac{\partial L^*}{\partial y_i} = 0, \quad y_i^* \geq 0, \quad i = \overline{1,m}$$

$$L^* = L(x^*, y^*)$$

Общая задача НЛП
Условия существования седловой точки

Теорема Куна-Таккера и Условия существования седловой точки функции Лагранжа эквивалентны

$$V = \frac{\partial L^*}{\partial x} = f'(x^*) + \sum_{i=1}^{m} y_i^* g_i'(x*), \qquad g_i(x^*) = \frac{\partial L^*}{\partial y_i}$$

1) $\frac{\partial L^*}{\partial x_j} \geq 0$ эквивалентно $V_j^* \geq 0,\ \ j = \overline{1,n}$

2) $x_j^* \frac{\partial L^*}{\partial x_j} = 0$ эквивалентно $V_j^* x_j^* = 0$

3) $y_i^* \frac{\partial L^*}{\partial y_i} = 0$ эквивалентно $y_i^* g_i(x^*) = 0$

4) $y_i^* \geq 0$ эквивалентно $y_i^* \geq 0$

5) $\frac{\partial L^*}{\partial y_i} \leq 0$ эквивалентно $g_i(x) \leq 0$ - условие допустимости

6) $x_j^* \geq 0$ эквивалентно $x_j^* \geq 0$ - условие допустимости

Общая задача НЛП

В случае, если задача имеет вид:

$$f(x) \to \max \qquad (1')$$

$$g_i(x) \le 0,\ i = \overline{1,m} \qquad (2)$$

$$x_j \ge 0,\ \ j = \overline{1,n} \qquad (3)$$

Теорема (Куна-Таккера) для задачи на максимум

Если $f(x)$- вогнутая, $g_i(x)$ - выпуклы, множество *D* регулярно по Слейтеру, то $x^* \in D$ - оптимальное решение задачи тогда и только тогда, когда $\exists y_i^* \ge 0,\ \ i = \overline{1,m},\ \ V_j^* \ge 0,\ \ j = \overline{1,n}$

$$-f'(x^*) + \sum_{i=1}^{m} y_i^* g_i'(x^*) - \sum_{j=1}^{n} V_j^* l_j = 0,$$

при этом выполнены условия дополняющей нежесткости:

$$y_i^* g_i(x^*) = 0,\ \ i = \overline{1,m}, \qquad V_j^* x_j^* = 0,\ \ j = \overline{1,n},$$

где $l_j,\ \ j = \overline{1,n}$ - *j*-ый координатный вектор l_j=(0,…,1,…,0)

Задача квадратичного программирования

Частным случаем задачи нелинейного программирования является **задача квадратичного программирования**, в которой целевая функция представляет собой сумму линейной и квадратичной функции (квадратичной формы):

$$f(x_1, x_2, \ldots, x_n) = \sum_{j=1}^{n} c_1 x_1 + \sum_{k=1}^{n} \sum_{j=1}^{n} d_{kj} x_k x_j =$$

$$= c_1 x_1 + c_2 x_2 + \ldots + c_n x_n + d_{11} x_1^2 + d_{22} x_2^2 + \ldots + d_{nn} x_n^2 +$$

$$+ d_{12} x_1 x_2 + d_{13} x_1 x_3 + \ldots + d_{n-1,n} x_{n-1} x_n$$

а ограничения являются линейными:

$$\begin{cases} a_{11} x_1 + \ldots + a_{1n} x_n \le b_1 \\ \ldots \\ a_{m1} x_1 + \ldots + a_{mn} x_n \le b_m \end{cases}$$

$$x_j \ge 0, \; j = \overline{1, n}$$

Задача квадратичного программирования

- Как и в общем случае решения задач нелинейного программирования, для определения глобального экстремума задачи квадратичного программирования не существует эффективного вычислительного метода, если не известно, что любой локальный экстремум является одновременно и глобальным.
- Так как в задаче квадратичного программирования множество допустимых решений выпукло, то, если целевая функция вогнута, любой локальный максимум является глобальным; если же целевая функция - выпуклая, то любой локальный минимум также и глобальный.
- Целевая функция представляет собой сумму линейной функции (которая является и выпуклой, и вогнутой) и квадратичной формы.
- Если квадратичная форма является вогнутой (выпуклой), то задачи отыскания максимума (минимума) целевой функции могут быть успешно решены.
- Вопрос о том, будет ли квадратичная форма вогнутой или выпуклой, зависит от того, является ли она отрицательно-определенной, отрицательно-полуопределенной, положительно-определенной, положительно-полуопределенной или вообще неопределенной.

Задача квадратичного программирования

Для решения задачи воспользуемся условиями существования седловой точки функции Лагранжа.

1) $\dfrac{\partial L^*}{\partial x_j} \geq 0$ 2) $x_j^* \dfrac{\partial L^*}{\partial x_j} = 0$ 3) $x_j^* \geq 0$, $j = \overline{1,n}$

4) $\dfrac{\partial L^*}{\partial y_i} \leq 0$ 5) $y_i^* \dfrac{\partial L^*}{\partial y_i} = 0$ 6) $y_i^* \geq 0$, $i = \overline{1,m}$

Если область допустимых решений не пустая, то условие регулярности Слейтера выполнено.

Задача квадратичного программирования

Для решения задачи воспользуемся условиями существования седловой точки функции Лагранжа.

1) $\dfrac{\partial L^*}{\partial x_j} \geq 0$ 2) $x_j^* \dfrac{\partial L^*}{\partial x_j} = 0$ 3) $x_j^* \geq 0$, $j = \overline{1, n}$

4) $\dfrac{\partial L^*}{\partial y_i} \leq 0$ 5) $y_i^* \dfrac{\partial L^*}{\partial y_i} = 0$ 6) $y_i^* \geq 0$, $i = \overline{1, m}$

Если область допустимых решений не пустая, то условие регулярности Слейтера выполнено.

Функция Лагранжа для задачи квадратичного программирования на минимум:

$$L(x, y) = \sum_{j=1}^{n} c_j x_j + \sum_{k=1}^{n} \sum_{j=1}^{n} d_{kj} x_k x_j + \sum_{i=1}^{m} y_i \left(\sum_{j=1}^{n} a_{ij} x_j - b_j\right)$$

Задача квадратичного программирования

Функция Лагранжа для задачи квадратичного программирования на минимум:

$$L(x, y) = \sum_{j=1}^{n} c_j x_j + \sum_{k=1}^{n} \sum_{j=1}^{n} d_{kj} x_k x_j + \sum_{i=1}^{m} y_i \left(\sum_{j=1}^{n} a_{ij} x_j - b_j\right)$$

Частные производные функции Лагранжа по переменным x будут линейными функциями.

Преобразуем неравенства (1) и (4) в равенства:

$$\frac{\partial L^*}{\partial x_j} - s_j = 0$$

$$\frac{\partial L^*}{\partial y_i} + w_i = 0$$

Задача квадратичного программирования

Функция Лагранжа для задачи квадратичного программирования на минимум:

$$L(x,y)=\sum_{j=1}^{n}c_j x_j+\sum_{k=1}^{n}\sum_{j=1}^{n}d_{kj}x_k x_j+\sum_{i=1}^{m}y_i\Big(\sum_{j=1}^{n}a_{ij}x_j-b_j\Big)$$

Частные производные функции Лагранжа по переменным x будут линейными функциями.

Преобразуем неравенства (1) и (4) в равенства:

$$\frac{\partial L^*}{\partial x_j}-s_j=0$$

$$\frac{\partial L^*}{\partial y_i}+w_i=0$$

Тогда условия (3) и (5) можно записать в следующем виде:

$$\frac{\partial L^*}{\partial x_j}-s_j=0 \qquad x_j^* s_j=0 \quad x_j^*\ge 0 \quad s_j\ge 0\,, \quad j=\overline{1,n}$$

$$\frac{\partial L^*}{\partial y_i}+w_i=0 \qquad y_i^* w_i=0 \quad y_i^*\ge 0 \quad w_i\ge 0\,, \quad i=\overline{1,m}$$

Задача квадратичного программирования

- Таким образом, чтобы найти решение задачи квадратичного программирования, нужно найти **неотрицательное** решение системы уравнений, удовлетворяющее условиям (3)(5).
- Данное решение можно найти, применив метод искусственного базиса для решения задачи максимизации функции:

$$-\sum z_j \to max,$$

где z_j – искусственные переменные, введенные в ограничения (1) с преобразованными ограничениями (1)(2),удовлетворяющее условиям (3)(5).

Задача квадратичного программирования

Пример

$$f(x_1, x_1) = 2x_1 + 4x_2 - x_1^2 - 2x_2^2 \to max$$

$$\begin{cases} x_1 + 2x_2 \leq 8 \\ 2x_1 - x_2 \leq 12 \end{cases}$$

$$x_1 \geq 0, x_2 \geq 0$$

Задача квадратичного программирования
Пример

Поставим задачу на минимум, возьмем противоположную функцию:

$$f(x_1, x_1) = -2x_1 - 4x_2 + x_1^2 + 2x_2^2 \to min$$

$$\begin{cases} x_1 + 2x_2 \leq 8 \\ 2x_1 - x_2 \leq 12 \end{cases}$$

$$x_1 \geq 0, x_2 \geq 0$$

Задача квадратичного программирования
Пример

Поставим задачу на минимум, возьмем противоположную функцию:

$$f(x_1, x_1) = -2x_1 - 4x_2 + x_1^2 + 2x_2^2 \rightarrow min$$

$$\begin{cases} x_1 + 2x_2 \leq 8 \\ 2x_1 - x_2 \leq 12 \end{cases}$$

$$x_1 \geq 0, x_2 \geq 0$$

$f''_{xx} = \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix}$ - положительно-определенная, следовательно, целевая функция – выпуклая функция, и любое локальное решение задачи будет являться ее глобальным решением.

Задача квадратичного программирования
Пример

Поставим задачу на минимум, возьмем противоположную функцию:

$$f(x_1, x_1) = -2x_1 - 4x_2 + x_1^2 + 2x_2^2 \to min$$

$$\begin{cases} x_1 + 2x_2 \leq 8 \\ 2x_1 - x_2 \leq 12 \end{cases}$$
$$x_1 \geq 0, x_2 \geq 0$$

$f''_{xx} = \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix}$ - положительно-определенная, следовательно, целевая функция – выпуклая функция, и любое локальное решение задачи будет являться ее глобальным решением.

Функция Лагранжа:

$$L(x, y) = -2x_1 - 4x_2 + x_1^2 + 2x_2^2 + y_1(x_1 + 2x_2 - 8) + y_2(2x_1 - x_2 - 12)$$

Задача квадратичного программирования
Пример

Функция Лагранжа:

$$L(x,y) = -2x_1 - 4x_2 + x_1^2 + 2x_2^2 + y_1(x_1 + 2x_2 - 8) + y_2(2x_1 - x_2 - 12)$$

Запишем условия существования седловой точки:

$$\frac{\partial L}{\partial x_1} = -2 + 2x_1 + y_1 + 2y_2 \geq 0$$

$$\frac{\partial L}{\partial x_2} = -4 + 4x_2 + 2y_1 - y_2 \geq 0$$

Задача квадратичного программирования
Пример

Функция Лагранжа:

$$L(x, y) = -2x_1 - 4x_2 + x_1^2 + 2x_2^2 + y_1(x_1 + 2x_2 - 8) + y_2(2x_1 - x_2 - 12)$$

Запишем условия существования седловой точки:

$$\frac{\partial L}{\partial x_1} = -2 + 2x_1 + y_1 + 2y_2 \geq 0$$

$$\frac{\partial L}{\partial x_2} = -4 + 4x_2 + 2y_1 - y_2 \geq 0$$

$$\frac{\partial L}{\partial y_1} = x_1 + 2x_2 - 8 \leq 0$$

$$\frac{\partial L}{\partial y_2} = 2x_1 - x_2 - 12 \leq 0$$

Задача квадратичного программирования
Пример

Условия дополняющей нежесткости:

$$x_1 \frac{\partial L}{\partial x_1} = x_1\left(-2 + 2x_1 + y_1 + 2y_2\right) = 0$$

$$x_2 \frac{\partial L}{\partial x_2} = x_2\left(-4 + 4x_2 + 2y_1 - y_2\right) = 0$$

$$y_1 \frac{\partial L}{\partial y_1} = y_1(x_1 + 2x_2 - 8) = 0$$

$$y_2 \frac{\partial L}{\partial y_2} = y_2(2x_1 - x_2 - 12) = 0$$

Требования неотрицательности:

$$x_1, x_2 \geq 0, y_1, y_2 \geq 0$$

Задача квадратичного программирования
Пример

Перепишем неравенства в следующем виде:

$$\begin{cases} 2x_1 + y_1 + 2y_2 \geq 2 \\ 4x_2 + 2y_1 - y_2 \geq 4 \\ x_1 + 2x_2 \leq 8 \\ 2x_1 - x_2 \leq 12 \end{cases}$$

Задача квадратичного программирования
Пример

Перепишем неравенства в следующем виде:
Преобразуем неравенства в равенства:

$$\begin{cases} 2x_1 + y_1 + 2y_2 \geq 2 \\ 4x_2 + 2y_1 - y_2 \geq 4 \\ x_1 + 2x_2 \leq 8 \\ 2x_1 - x_2 \leq 12 \end{cases} \qquad \begin{cases} 2x_1 + y_1 + 2y_2 - w_1 = 2 \\ 4x_2 + 2y_1 - y_2 - w_2 = 4 \\ x_1 + 2x_2 + w_3 = 8 \\ 2x_1 - x_2 + w_4 = 12 \end{cases}$$

Задача квадратичного программирования
Пример

Поставим задачу линейного программирования для решения методом искусственного базиса:

$$g = -z_1 - z_2 \to max$$

$$\begin{cases} 2x_1 + y_1 + 2y_2 - w_1 + z_1 = 2 \\ 4x_1 + 2y_1 - y_2 - w_2 + z_2 = 4 \\ x_1 + 2x_2 + w_3 = 8 \\ 2x_1 - x_2 + w_4 = 12 \end{cases}$$

$$x_1, x_2 \geq 0, y_1, y_2 \geq 0, w_1, w_2, w_3, w_4 \geq 0, z_1, z_2 \geq 0$$

Задача квадратичного программирования
Пример

Поставим задачу линейного программирования для решения методом искусственного базиса:

$$g = -z_1 - z_2 \to max$$

$$\begin{cases} 2x_1 + y_1 + 2y_2 - w_1 + z_1 = 2 \\ 4x_1 + 2y_1 - y_2 - w_2 + z_2 = 4 \\ x_1 + 2x_2 + w_3 = 8 \\ 2x_1 - x_2 + w_4 = 12 \end{cases}$$

$$x_1, x_2 \geq 0, y_1, y_2 \geq 0, w_1, w_2, w_3, w_4 \geq 0, z_1, z_2 \geq 0$$

Условия дополняющей нежесткости:

$x_1 w_1 = 0,\ x_2 w_2 = 0,\ y_1 w_3 = 0,\ y_2 w_4 = 0$

Задача квадратичного программирования
Пример

Решим задачу линейного программирования симплекс-методом.

Полученный ответ:
$x_1 = 1, x_2 = 1, w_3 = 5, w_4 = 11$, все остальные переменные равны 0.

Задача квадратичного программирования
Пример

Решим задачу линейного программирования симплекс-методом.

Полученный ответ:
$x_1 = 1, x_2 = 1, w_3 = 5, w_4 = 11$, все остальные переменные равны 0.

Проверим условия дополняющей нежесткости:
$x_1 w_1 = 1 \cdot 0 = 0$, $x_2 w_2 = 1 \cdot 0 = 0$,
$y_1 w_3 = 0 \cdot 5 = 0$, $y_2 w_4 = 0 \cdot 11 = 0$

Таким образом, условия дополняющей нежесткости выполнены.

Задача квадратичного программирования
Пример

Решим задачу линейного программирования симплекс-методом.

Полученный ответ:
$x_1 = 1, x_2 = 1, w_3 = 5, w_4 = 11$, все остальные переменные равны 0.

Проверим условия дополняющей нежесткости:
$x_1 w_1 = 1 \cdot 0 = 0,\ x_2 w_2 = 1 \cdot 0 = 0,$
$y_1 w_3 = 0 \cdot 5 = 0,\ y_2 w_4 = 0 \cdot 11 = 0$

Таким образом, условия дополняющей нежесткости выполнены.

Так как решаемая задача – задача выпуклого нелинейного программирования, то необходимые условия экстремума являются и достаточными, следовательно, полученное решение является глобальным минимумом:

$$x_{min} = (1,1),\ \ f_{min} = -3\ \Rightarrow\ f_{max} = 3$$

Общая задача НЛП
Условный экстремум при смешанных ограничениях

$$f(x) \to \min$$

$$D\begin{cases} g_i(x) = 0, \quad i = \overline{1,k}, \quad k < n \\ g_i(x) \le 0, \quad i = \overline{k+1,m} \end{cases}$$

$f(x), g_i(x)$ - дважды непрерывно дифференцируемы.

Общая задача НЛП

Условный экстремум при смешанных ограничениях

Теорема (Необходимые условия экстремума 1-го порядка)

Пусть x^* - точка локального минимума, следовательно $\exists y_0^*$ и $y^* = (y_1^*, ..., y_m^*)$ не равные нулю одновременно, что выполняются следующие условия:

1) $\dfrac{\partial L(x^*, y_0^*, y^*)}{\partial x_j} = 0, \quad j = \overline{1, n}$ - условия стационарности обобщенной функции Лагранжа

2) $g_i(x^*) = 0, \ i = \overline{1, k}, \ g_i(x^*) \le 0, \ i = \overline{k+1, m}$ - условия допустимости решения

3) $y_i^* \ge 0, \ i = \overline{k+1, m}$ - условия неотрицательности в случае минимума ($y_i^* \le 0, \ i = \overline{k+1, m}$ в случае максимума).

4) $y_i^* g_i(x^*) = 0, \ i = \overline{k+1, m}$ - условия дополняющей нежесткости.

Если при этом градиенты активных ограничений-неравенств и ограничений-равенств в точке x* линейно независимы (выполняется условие регулярности), то $y_0^* \ne 0$

Общая задача НЛП
Условный экстремум при смешанных ограничениях

Теорема (достаточные условия минимума (максимума) первого порядка).

Пусть имеется точка (x^*,y^*), удовлетворяющая системе

$$\frac{\partial L\left(x^*, y_0^*, y^*\right)}{\partial x_j} = 0, \quad j = \overline{1, n}$$ при $y_0^* \neq 0$,

суммарное число активных ограничений-неравенств в точке x^* и ограничений-равенств совпадает с числом n переменных (при этом условие регулярности выполняется).

Если $y_i^* \geq 0, \ \forall i \in I\left(x^*\right)$, то x^* – точка условного локального минимума в задаче (1-3).

Если $y_i^* \leq 0, \ \forall i \in I\left(x^*\right)$, то x^* – точка условного локального максимума в задаче (1-3).

Общая задача НЛП
Условный экстремум при смешанных ограничениях

Теорема (необходимые условия минимума (максимума) второго порядка).

Пусть x^* – регулярная точка минимума (максимума) в задаче (1-3) и имеется решение (x^*, y^*) системы $\frac{\partial L(x^*, y_0^*, y^*)}{\partial x_j} = 0, \quad j = \overline{1, n}$.

Тогда $< L''_{xx}(x^*, y^*)h, \ h > \geq 0$ $(< L''_{xx}(x^*, y^*)h, \ h > \leq 0)$ для $\forall \ h \neq 0 \in R^n$, при которых

$$< g_i'(x^*), h >= 0, \ i = \overline{1, k}, \quad y_i^* > 0 \ (y_i^* < 0)$$

$$< g_i'(x^*), h > \leq 0, \ i = \overline{k+1, m}, i \in I(x^*), \ y_i^* = 0$$

Общая задача НЛП
Условный экстремум при смешанных ограничениях

Теорема (достаточные условия экстремума второго порядка).

Пусть имеется точка (x^*, y^*), удовлетворяющая системе $\frac{\partial L(x^*, y_0^*, y^*)}{\partial x_j} = 0, \quad j = \overline{1,n}$ при $y_0^* \neq 0$.

Если в этой точке $< L''_{xx}(x^*, y^*)h,\ h >> 0$ ($< L''_{xx}(x^*, y^*)h,\ h >< 0$) для всех $\forall\ h \neq 0 \in R^n$ таких, что

$$< g_i'(x^*),\ h >= 0,\ \ i = \overline{1,k},\quad y_i^* > 0\ (\ y_i^* < 0\ )$$

$$< g_i'(x^*),\ h >\leq 0,\ \ i = \overline{k+1,m},\ i \in I(x^*),\ \ y_i^* = 0$$

то точка x∗ является точкой локального минимума (максимума) в задаче (1-3).

Страница с ошибкой и исправление

1. Страницы 28-30: $y^* = -2$
2. Страница 34: $L(x^0, y) \leq L(x^0, y^0) \leq L(x, y^0)$
3. Страница 37: $\exists \tilde{x}: g_i(\tilde{x}) < 0, i = \overline{1, m}$
4. Страница 42: Определение строго выпуклой функции –
$$f(\lambda x^1 + (1-\lambda)x^2) < \lambda f(x^1) + (1-\lambda) f(x^2)$$
5. Страница 51: $L(x_1, x_2, y) = y_0(x_1^2 + x_2^2) + y_1(x_1 + x_2 - 2)$
6. Страница 59:

Если $f(x)$, $g_i(x)$ - выпуклы, множество D регулярно по Слейтеру, то $x^* \in D$ - оптимальное решение задачи тогда и только тогда, когда $\exists y_i^* \geq 0,\ i = \overline{1,m},\ V_j^* \geq 0,\ j = \overline{1,n}$

Надо дописать, что при этом y_i^* и V_j^* одновременно не равны нулю

7. Страница 63: $f(x_1, x_2, \ldots, x_n) = \sum_{j=1}^{n} c_1 x_1 + \sum_{k=1}^{n}\sum_{j=1}^{n} d_{kj} x_k x_j =$

Надо написать: $f(x_1, x_2, \ldots, x_n) = \sum_{j=1}^{n} c_j x_j +$ то что в лекции

8. Страницы 70-73: $f(x_1, x_2)$
9. Страница 72-73: $f''_{xx} = \begin{pmatrix} 2 & 0 \\ 0 & 4 \end{pmatrix}$

Выполнили: Титов, Кубалов, Петрищев