Алгоритмы и вычислительные методы оптимизации

Установочная лекция

Информация о преподавателе и дисциплине

Преподаватель дисциплины - Галкина Марина Юрьевна, доцент кафедры прикладной математики и кибернетики (e-mail: gmur7@bk.ru)

По курсу Алгоритмы и вычислительные методы оптимизации в ЭИОС выложены лекции, задания на 3 лабораторные, курсовую работу, темы задач на экзамен.

В ЭИОС выкладываются архивы с отчетами и исходными кодами программ лабораторных и курсовой работ. Защита лабораторных и курсовых работ будет проводиться на занятиях.

После сдачи лабораторных и курсовой работ сдается экзамен.

Курсовая работа состоит из 4-х частей.

- Первая часть – привести поставленную задачу к канонической форме.
- Вторая часть – реализация программы (оценка отлично ставится только при реализации класса простых дробей).
- В третьей части требуется сделать чертеж. Можно воспользоваться ресурсом https://www.desmos.com/calculator?lang=ru, потом доработать чертеж в Paint. На чертеже обязательно отметить точки, соответствующие таблицам из программы.
- В четвертой части требуется составить и решить систему (можно набрать с использованием редактора формул или отсканировать рукописный вариант).

1. Линейное программирование

1.1 Пример задачи линейного программирования (задача использования сырья)

На мебельной фабрике могут выпускать стулья и кресла. Сведения о ресурсах, расходе материалов и прибыли от реализации каждого изделия приведены в таблице.

Ресурсы	Запасы	Расход на единицу продукции	
		стул	кресло
Пиломатериалы (м^3)	10	0.01	0.03
Ткань (м^2)	2000	0.5	2
Рабочее время (ч)	1000	2	5
Прибыль от ед. продукции (у.е.)		10	35

Найти план производства продукции максимизирующий прибыль предприятия.

Для решения задачи построим математическую модель.

x_1 – кол-во выпущенных стульев

x_2 – кол-во выпущенных кресел

Математическая модель:

$$\begin{cases} 0.01x_1 + 0.03x_2 \le 10 \\ 0.5x_1 + 2x_2 \le 2000 \\ 2x_1 + 5x_2 \le 1000 \\ x_1, x_2 \ge 0 \end{cases}$$

Система ограничений

$$Z = 10x_1 + 35x_2 \to \max$$

Целевая функция

Если система ограничений и целевая функция линейны, то модель – задача линейного программирования.

1.2 Метод Жордана-Гаусса

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

Расширенная матрица системы:

$$\overline{A} = \left(\begin{array}{cccc|c} a_{11} & a_{12} & \dots & a_{1n} & b_1 \\ a_{21} & a_{22} & \dots & a_{2n} & b_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} & b_m \end{array} \right)$$

Элементарные преобразования над строками матрицы

- Перестановка строк.
- Умножение всех элементов выбранной строки на число, отличное от 0.
- Сложение соответствующих элементов двух строк.
- Вычеркивание строк, состоящих из нулей.

Пусть имеется система с n неизвестными и m уравнениями, $m \le n$.

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

С помощью элементарных преобразований над строками расширенную матрицу системы можно привести к виду:

$$\overline{A} \sim \left(\begin{array}{cccccccc|c} 1 & 0 & \dots & 0 & a'_{1,r+1} & a'_{1,r+2} & \dots & a'_{1,n} & b'_1 \\ 0 & 1 & \dots & 0 & a'_{2,r+1} & a'_{2,r+2} & \dots & a'_{2,n} & b'_2 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \underbrace{0 \quad 0 \quad \dots \quad 1}_{r} & & & & a'_{r,r+1} & a'_{r,r+2} & \dots & a'_{r,n} & b'_r \end{array} \right) \left. \vphantom{\begin{array}{c} 1 \\ 0 \\ \dots \\ 0 \end{array}} \right\} r$$

где $r \le m$ и первые r столбцов могут занимать произвольные места в произвольном порядке до вертикальной черты.

Размер полученной единичной матрицы называется рангом матрицы.

rang(A)=r

Алгоритм метода Жордана-Гаусса ($m \le n$)

1. Пусть $r=m$
2. $s=1$, $k=1$

3. В s-ом столбце выберем элемент отличный от 0 среди элементов $a_{kk}, a_{k+1,k}, \ldots, a_{m,k}$. Если такого элемента нет, то увеличиваем s на 1 и переходим к п.3. Не теряя общности, можно считать, что $a_{kk} \neq 0$ (этого можно добиться перестановкой строк). Элемент a_{kk} называется разрешающим. Получаем новую матрицу, выполнив следующие преобразования:

– разделим k-ю строку на разрешающий элемент a_{kk};

– все элементы k-го столбца новой матрицы, кроме a_{kk}, равны 0;

– для расчета a'_{ij} – остальных элементов новой матрицы построим прямоугольники с вершинами в элементах a_{ij} и элементе a_{kk}:

$$\begin{array}{cc} \boxed{a_{kk}} & a_{kj} \\ a_{ik} & a_{ij} \end{array}$$

и найдем новые элементы по правилу прямоугольников: $a'_{ij} = a_{ij} - \dfrac{a_{kj} \cdot a_{ik}}{a_{kk}}$.

4. Вычеркнем строки, состоящие из нулей, если такие образовалась. При вычеркивании каждой строки, r уменьшается на 1.

5. Если k меньше r, то увеличиваем k и s на 1 и переходим к п.3, иначе переходим к п.6.

6. Ранг исходной матрицы будет равен r – размеру выделенной единичной матрицы.

Указанные выше преобразования матрицы называются *жордановыми исключениями*. В процессе жордановых исключений возможны следующие случаи:

1. Получена строка, состоящая из нулей, кроме последнего коэффициента (правой части уравнения). В этом случае система не имеет решения.

2. Ранг матрицы A равен количеству уравнений m и числу неизвестных n ($r = m = n$). Тогда система имеет единственное решение.

3. Ранг матрицы A не превосходит количества уравнений m ($r \le m$) и $m < n$. Тогда система имеет бесконечно много решений.

Пусть имеет место третий случай, и расширенная матрица системы приведена к виду:

$$
\bar{A} \sim
\begin{array}{c}
\begin{array}{cccccccc}
x_1 & x_2 & & x_r & x_{r+1} & x_{r+2} & & x_n
\end{array}
\\
\left.\left(\begin{array}{cccccccc|c}
1 & 0 & \dots & 0 & a'_{1,r+1} & a'_{1,r+2} & \dots & a'_{1,n} & b'_1 \\
0 & 1 & \dots & 0 & a'_{2,r+1} & a'_{2,r+2} & \dots & a'_{2,n} & b'_2 \\
\dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\
\underbrace{0 \quad 0 \quad \dots \quad 1}_{r} & & & & a'_{r,r+1} & a'_{r,r+2} & \dots & a'_{r,n} & b'_n
\end{array}\right)\right\} r
\end{array}
$$

По преобразованной матрице составим систему:

$$
\begin{cases}
x_1 + a'_{1,r+1}x_{r+1} + \dots + a'_{1n}x_n = b'_1, \\
x_2 + a'_{2,r+1}x_{r+1} + \dots + a'_{2n}x_n = b'_2, \\
\dots\dots\dots\dots\dots\dots\dots\dots\dots, \\
x_r + a'_{r,r+1}x_{r+1} + \dots + a'_{rn}x_n = b'_r.
\end{cases}
$$

Система, приведенная к единичному базису

Переменные $x_1, x_2, \dots, x_r$, соответствующие столбцам единичной матрицы, называются *базисными*, все остальные переменные $x_{r+1}, x_{r+2}, \dots, x_n$ – *свободными*.

$$\begin{cases} x_1 + a'_{1,r+1}x_{r+1} + \cdots + a'_{1n}x_n = b'_1, \\ x_2 + a'_{2,r+1}x_{r+1} + \cdots + a'_{2n}x_n = b'_2, \\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots, \\ x_r + a'_{r,r+1}x_{r+1} + \cdots + a'_{rn}x_n = b'_r. \end{cases}$$

Выразим базисные переменные через свободные:

$$\begin{cases} x_1 = b_1 - a_{1,r+1}x_{r+1} - \ldots - a_{1n}x_n \\ x_2 = b_2 - a_{2,r+1}x_{r+1} - \ldots - a_{2n}x_n \\ \ldots\ \ldots\ \ldots\ \ldots\ \ldots\ \ldots\ \ldots\ \ldots\ \ldots\ \ldots\ \ldots\ \ldots\ \ldots \\ x_r = b_r - a_{r,r+1}x_{r+1} - \ldots - a_{rn}x_n \end{cases}$$

Общее решение системы

Пример 1

Решить систему методом Жордана-Гаусса.

$$\begin{cases} 3x_1 \square\ 2x_2 \square\ 5x_3 \square\ 4x_4 \square\ 3 \\ x_1 \square\ x_2 \square\ x_3 \square\ 4x_4 \square\ \square\ 2 \\ 4x_1 \square\ x_2 \square\ 4x_3 \square\ 2 \end{cases}$$

Пример 2

Решить систему методом Жордана-Гаусса.

$$\begin{cases} 2x_1 \square\ 3x_2 \square\ 5x_3 \square\ 7x_4 \square\ 1 \\ 4x_1 \square\ 6x_2 \square\ 2x_3 \square\ 3x_4 \square\ 2 \\ 2x_1 \square\ 3x_2 \square\ 11x_3 \square\ 15x_4 \square\ 1 \end{cases}$$

Разобранные решения примеров - в файле lecture.pdf.

Метод Жордана-Гаусса с выбором главного элемента в столбце

При решении системы программно может возникнуть деление на очень маленькое число (если диагональный элемент близок к нулю).

Для этого в п. 3 алгоритма в качестве разрешающего выбирают максимальный по модулю элемент среди элементов s-го столбца, расположенных на и ниже k-ой строки и переставляют строки матрицы таким образом, чтобы этот элемент оказался на k-ой строке.

Этот метод надо будет реализовать в лабораторной работе. Можно реализовать класс простых дробей для использования в курсовой работе (отличная оценка за курсовую работу ставится только при работе с простыми дробями).

1.3 Различные формы записи задачи линейного программирования. Переход от одной формы записи к другой

Рассмотрим задачу линейного программирования (ЗЛП) *в общем виде*:

Целевая функция

$$Z = \sum_{j=1}^{n} c_j x_j \to \max(\min)$$

Система ограничений

$$\begin{cases} \sum\limits_{j=1}^{n} a_{ij}x_j \le b_i \quad (i = 1,\ldots,m_1), \\ \sum\limits_{j=1}^{n} a_{ij}x_j \ge b_i \quad (i = m_1+1,\ldots,m_2), \\ \sum\limits_{j=1}^{n} a_{ij}x_j = b_i \quad (i = m_2+1,\ldots,m), \\ x_j \ge 0 \;\; (j = 1,\ldots,k, k \le n). \end{cases}$$

где a_{ij}, b_i, c_j – заданные постоянные величины и $m_1 \le m_2 \le m$.

Задача линейного программирования задана *в канонической форме*, если требуется максимизировать целевую функцию, все ограничения системы – уравнения, и на все переменные наложено условие неотрицательности ($m_1 = m_2 = 0, k = n$).

Задача линейного программирования задана *в симметричной форме*, если требуется максимизировать целевую функцию, все ограничения системы – неравенства «≤» (или минимизировать целевую функцию, все ограничения системы – неравенства «≥») и на все переменные наложено условие неотрицательности.

Набор чисел $X = (x_1, x_2, \ldots, x_n)$ называется *допустимым решением (планом)*, если он удовлетворяет системе ограничений

Множество всех допустимых решений называется *областью допустимых решений* (ОДР).

Допустимое решение $X^* = (x_1^*, x_2^*, \ldots, x_n^*)$, для которого достигается максимальное (минимальное) значение функции, называется *оптимальным планом*. Значение целевой функции при плане X будем обозначать $Z(X)$.

Следовательно, если X^* – оптимальный план задачи, то для любого X выполняется неравенство $Z(X) \le Z(X^*)$ при нахождении максимума функции или $Z(X) \ge Z(X^*)$ при нахождении минимума функции.

Все три формы записи ЗЛП являются эквивалентными в том смысле, что имеются алгоритмы перехода от одной формы к другой. Таким образом, если имеется способ решения задачи в одной из форм, то всегда можно определить оптимальный план задачи, заданной в любой другой форме.

Алгоритмы перехода от одной формы к другой

- **Симметричная → каноническая.**

Переход осуществляется путем добавления в левую часть каждого неравенства дополнительной неотрицательной переменной. Если неравенство было «≤», то дополнительная переменная добавляется в левую часть неравенства со знаком «+». Если неравенство было «≥», то дополнительная переменная добавляется в левую часть неравенства со знаком «–». Вводимые дополнительные переменные называются *балансовыми*. Задачу минимизации функции Z заменяют на задачу максимизации функции $(-Z)$ и используют, что $\min Z = -\max(-Z)$.

- **Каноническая → симметричная.**

Для осуществления такого перехода находится общее решение системы уравнений – ограничений, целевая функция выражается через свободные переменные. Далее, воспользовавшись неотрицательностью базисных переменных, можно исключить их из задачи. Симметричная форма задачи будет содержать неравенства, связывающие только свободные переменные, и целевую функцию, зависящую только от свободных переменных. Значения базисных переменных находятся из общего решения исходной системы уравнений.

- **Общая → каноническая.**

Каждая переменная, на которую не было наложено условие неотрицательности, представляется в виде разности двух новых неотрицательных переменных. Неравенства преобразуются в уравнения путем введения в левую часть каждого неравенства балансовой переменной таким же образом, как это было описано при переходе от симметричной к канонической форме. Задачу минимизации функции Z заменяют на задачу максимизации функции $(–Z)$ таким же образом, как это было описано при переходе от симметричной к канонической форме.

1.4 Графический способ метод решения ЗЛП, заданной в симметричной форме, в случае двух переменных

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

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

Для решения этой задачи можно перебрать все вершины многоугольника, определяемого системой ограничений, и выбрать из них ту, в которой значение функции больше.

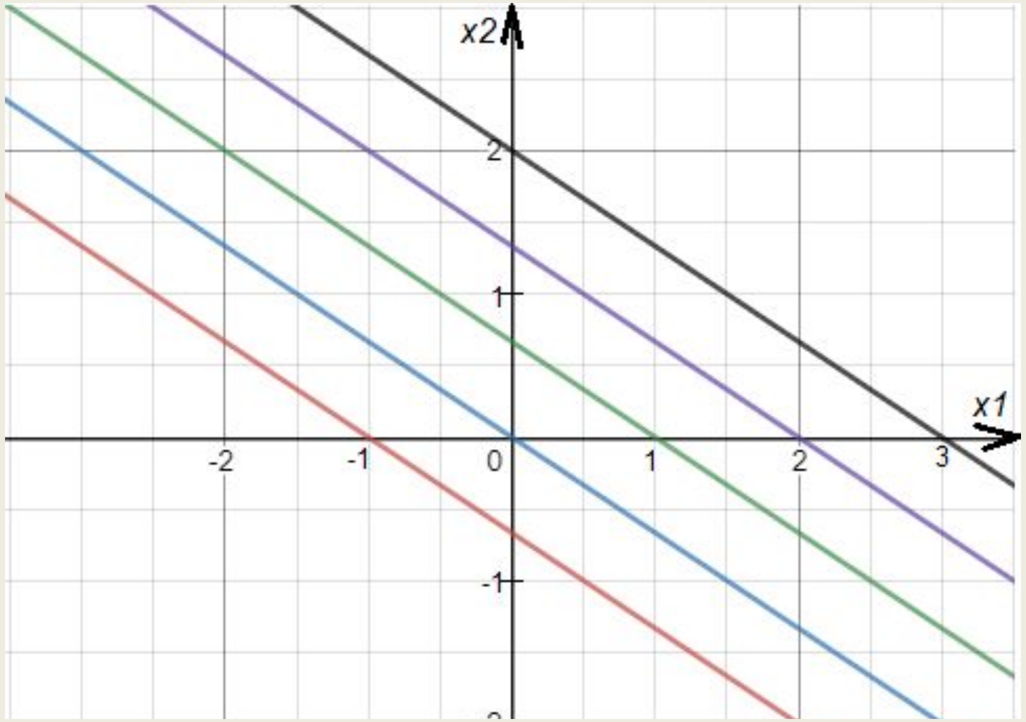
Линией уровня функции Z называется множество точек, удовлетворяющее уравнению $Z = c$, где c – произвольная константа. В двумерном случае, это уравнение определяет прямую. Линии уровня линейной функции образуют семейство параллельных прямых.

Пример 3

Построим линии уровня функции $Z=2x_1+3x_2$.

$2x_1+3x_2=6$

x_1	0	3
x_2	2	0



На рисунке приведены линии уровня функции Z. Первая линия (черная) $2x_1+3x_2=6$.

Вектор $grad \ Z = \left(\frac{\partial Z}{\partial x_1}, \frac{\partial Z}{\partial x_2} \right)$ называется *градиентом функции Z*. Вектор $grad \ Z$ в каждой точке перпендикулярен линии уровня, проходящей через эту точку, и указывает направление наибольшего возрастания функции Z, $grad \ Z = (c_1, c_2)$.

Наибольшее значение Z достигается в вершине, через которую проходит линия уровня, соответствующая наибольшему значению Z.

Алгоритм графического решения задачи линейного программирования

1. Записывают уравнения граничных прямых $a_{i1}x_1+a_{i2}x_2=b_i$ (i = 1,…,m) и строят их на плоскости X_1OX_2 по двум точкам.
2. Отмечают полуплоскости, соответствующие ограничениям - неравенствам. Для этого берут «пробную» точку, через которую не проходит граница полуплоскости (часто берут, если прямая не проходит через начало координат, точку (0,0)), и ее координаты подставляют в соответствующее ограничение-неравенство. Если полученное неравенство верное, то искомой будет полуплоскость, содержащая «пробную» точку; в противном случае, искомой будет полуплоскость, которой данная точка не принадлежит.
3. Заштриховывают многоугольник, определяемый системой ограничений. Для этого определяют общую часть ранее отмеченных m полуплоскостей, лежащую в первой четверти (следует из условий неотрицательности переменных).

4. Строят вектор $grad\ \ Z = (c_1, c_2)$ и одну из прямых семейства $Z = c$ (чаще всего $Z = 0$). Если выбранный для построения многоугольника масштаб не позволяет построить $grad\ \ Z$, то вместо него строят вектор $\overline{N} = k \cdot grad\ \ Z$ (в случае, если длина $grad\, Z$ слишком мала для построения) или $\overline{N} = \dfrac{grad\ \ Z}{k}$ (в случае, если длина $grad\ \ Z$ слишком велика для построения), где $k > 1$.

5. В случае необходимости параллельным переносом линию уровня следует расположить таким образом, чтобы многоугольник находился впереди линии уровня по направлению $grad\, Z$.

6. Определяют экстремальную точку, соответствующую вершине многоугольника, путем параллельного перемещения прямой $Z = c$ в направлении вектора $grad\, Z$. Это будет наиболее удаленная вершина многоугольника, в которой линия уровня пересекается с многоугольником.

7. Вычисляют координаты оптимальной точки и значение функции Z в этой точке.

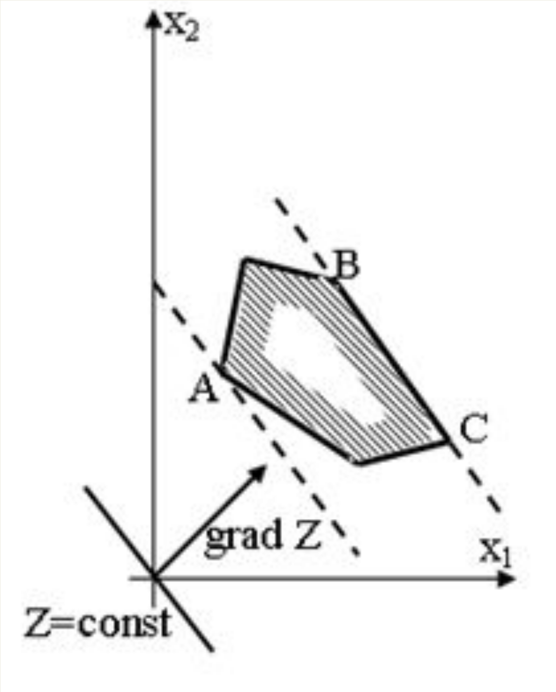
Замечание:

1. Если у функции требуется найти минимальное значение, то линию уровня $Z=c$ перемещают в направлении вектора $-grad\,Z$ (или перемещают в направлении $grad\,Z$, но находят первую вершину пересечения линии уровня с многоугольником).

2. Если масштаб по осям выбран одинаковым, то линия уровня перпендикулярна $grad\,Z$

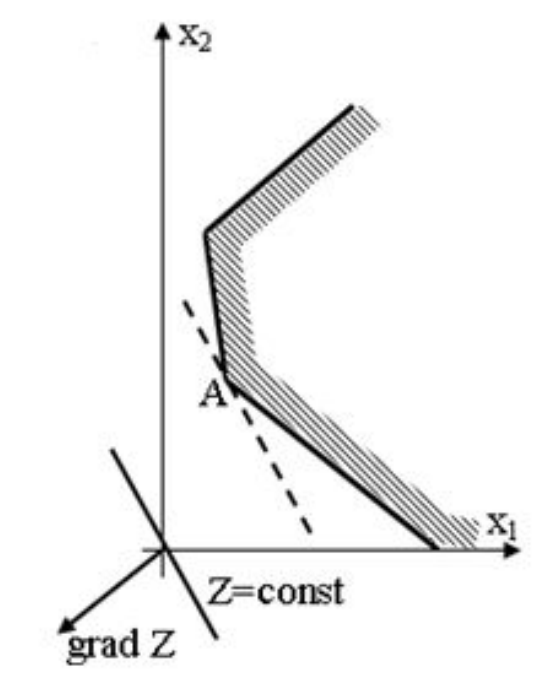
В зависимости от особенностей области допустимых решений и взаимного расположения области и вектора $grad\,Z$ при решении задачи линейного программирования возможны различные случаи.

Если область допустимых решений – ограниченный многоугольник, то может быть либо единственное решение, либо бесконечно много решений – все точки отрезка, соединяющего две вершины многоугольника (альтернативный оптимум). В случае альтернативного оптимума оптимальный план представляется выражением координат произвольной точки отрезка через координаты ее концов.

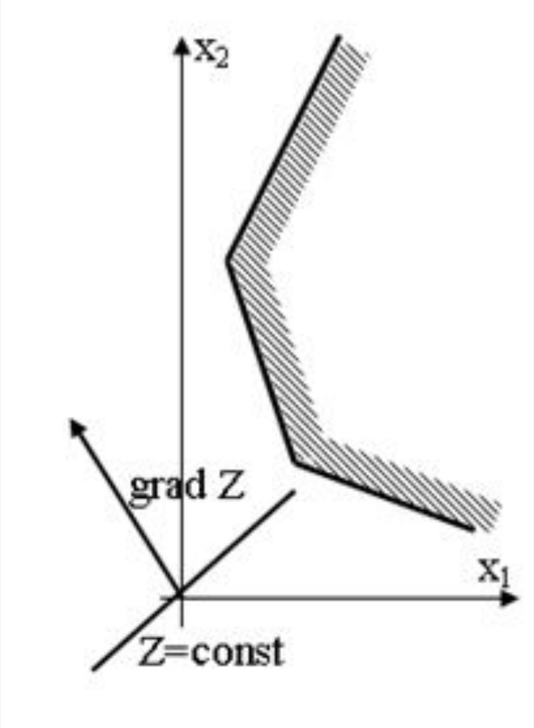


Функция достигает минимума в точке *A*,
а максимума – в любой точке отрезка *BC* (альтернативный оптимум).

Если область допустимых решений – неограниченный многоугольник, то в зависимости от направления $grad\,Z$ задача может иметь или не иметь решения. В этом случае так же может оказаться альтернативный оптимум.



Максимум функции достигается в точке A, минимума функция не имеет.



Функция не имеет ни минимума, ни максимума.

Пример 4

Решить графически задачу линейного программирования

$$Z = x_1 + x_2 \to \max$$

$$\begin{cases} x_1 + 2x_2 \le 14 \\ -5x_1 + 3x_2 \le 15 \\ 4x_1 + 6x_2 \ge 24 \\ x_1, x_2 \ge 0 \end{cases}$$

Разобранное решение примера - в файле lecture.pdf.

1.5 Симплекс-метод

Симплекс-метод (метод последовательного улучшения плана) позволяет решить любую ЗЛП, заданную в канонической форме.

Симплекс – простейший многогранник данного числа измерений, например, треугольник в R^2, тетраэдр – в R^3 и т.д.). Название метода связано с тем историческим обстоятельством, что ограничения одной из первых задач, решенных этим методом, задавали симплекс в пространстве соответствующей размерности.

Пусть задача линейного программирования представлена в канонической форме:

$$Z = c_1 x_1 + c_2 x_2 + \dots + c_n x_n \to \max \qquad (1.15)$$

$$\begin{cases} a_{11} x_1 + a_{12} x_2 + \dots + a_{1n} x_n = b_1 \\ \dots \\ a_{m1} x_1 + a_{m2} x_2 + \dots + a_{mn} x_n = b_m \\ x_1, x_2, \dots, x_n \ge 0 \end{cases} \qquad (1.16)$$

Прежде, чем описать алгоритм симплекс-метода, сформулируем ряд теорем, которые дают этот алгоритм.

Пусть система (1.16) приведена к единичному базису с неотрицательными правыми частями $(b_i \ge 0, i = 1, 2, \ldots, m)$:

$$\begin{cases} x_1 + a'_{1,r+1}x_{r+1} + \ldots + a'_{1n}x_n = b'_1 \\ x_2 + a'_{2,r+1}x_{r+1} + \ldots + a'_{2n}x_n = b'_2 \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots , \\ x_r + a'_{r,r+1}x_{r+1} + \ldots + a'_{rn}x_n = b'_r \\ x_i \ge 0, i = 1, 2, \ldots, n \end{cases} \qquad (1.17)$$

Выразим целевую функцию через свободные переменные.

$$Z = c'_0 + c'_1 x_{r+1} + \ldots + c'_n x_n \to \max \qquad (1.18)$$

Системе (1.17) соответствует опорное решение $(b'_1, \ldots, b'_r, 0, \ldots, 0)$, при этом значение функции $Z = c'_0$.

Из (1.17), (1.18) можно перейти к симметричной форме ЗЛП и для нее построить область допустимых решений – многогранник.

Теорема 8 Каждому опорному решению системы (1.16) соответствует вершина многогранника планов эквивалентной симметричной задачи и наоборот.

Из теоремы 8 следует, что максимального значения функция достигает для одного из опорных решений системы (1.16).

Симплекс-метод позволяет найти этот опорный план в результате упорядоченного перебора опорных планов. При этом упорядоченность понимается в том смысле, что переход от одного опорного решения к другому значения функции не убывают.

$$\begin{cases} x_1 + a'_{1,r+1}x_{r+1} + \ldots + a'_{1n}x_n = b'_1 \\ x_2 + a'_{2,r+1}x_{r+1} + \ldots + a'_{2n}x_n = b'_2 \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ x_r + a'_{r,r+1}x_{r+1} + \ldots + a'_{rn}x_n = b'_r \\ x_i \geq 0, i = 1, 2, \ldots, n \end{cases}, \tag{1.17}$$

$$Z = c'_0 + c'_1 x_{r+1} + \cdots + c'_n x_n \to \max \tag{1.18}$$

Средством хранения информации о решаемой задаче линейного программирования являются симплекс-таблицы, начальная таблица составляется по (1.17), (1.18) и имеет следующий вид:

$$\begin{array}{c|c|ccccccc} б.п. & 1 & x_1 & x_2 & \cdots & x_r & x_{r+1} & \cdots & x_n \\ \hline x_1 & b'_1 & 1 & 0 & \cdots & 0 & a'_{1,r+1} & \cdots & a'_{1,n} \\ x_2 & b'_2 & 0 & 1 & \cdots & 0 & a'_{2,r+1} & \cdots & a'_{2,n} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_r & b'_r & 0 & 0 & \cdots & 1 & a'_{r,r+1} & \cdots & a'_{r,n} \\ \hline Z & c'_0 & 0 & 0 & \cdots & 0 & -c'_{r+1} & \cdots & -c'_n \end{array} \tag{1.19}$$

Последнюю строку называют *строкой целевой функции* или Z-строкой. Столбец с заголовком 1 содержит правые части уравнений (1.17), свободный член целевой функции и называется *столбцом свободных членов*.

б.п.	1	x_1	x_2	$\ldots$	x_r	x_{r+1}	$\ldots$	x_n
x_1	b'_1	1	0	$\ldots$	0	$a'_{1,r+1}$	$\ldots$	$a'_{1,n}$
x_2	b'_2	0	1	$\ldots$	0	$a'_{2,r+1}$	$\ldots$	$a'_{2,n}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
x_r	b'_r	0	0	$\ldots$	1	$a'_{r,r+1}$	$\ldots$	$a'_{r,n}$
Z	c'_0	0	0	$\ldots$	0	$-c'_{r+1}$	$\ldots$	$-c'_n$

(1.19)

Таблице (1.19) соответствует опорное решение $(b'_1,\ldots,b'_r,0,\ldots,0)$, при этом значение функции $Z = c'_0$.

Теорема 9 (признак оптимальности опорного решения) Если в таблице (1.19) коэффициенты $-c'_j \geq 0$ $(j = r+1,\ldots,n)$, то опорное решение $X' = (b'_1,\ldots,b'_r,0,\ldots,0)$ является оптимальным и при этом $Z_{\max} = Z(X') = c'_0$.

Теорема 10 (признак неограниченности функции) Если в таблице (1.19) существует хотя бы один коэффициент $-c'_j < 0$ такой, что среди элементов a'_{ij} $(i = 1,2,\ldots,r)$ нет положительных, то целевая функция (1.18) при ограничениях (1.17), не ограничена сверху.

б.п.	1	x_1	x_2	$\cdots$	x_r	x_{r+1}	$\cdots$	x_n
x_1	b'_1	1	0	$\cdots$	0	$a'_{1,r+1}$	$\cdots$	$a'_{1,n}$
x_2	b'_2	0	1	$\cdots$	0	$a'_{2,r+1}$	$\cdots$	$a'_{2,n}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
x_r	b'_r	0	0	$\cdots$	1	$a'_{r,r+1}$	$\cdots$	$a'_{r,n}$
Z	c'_0	0	0	$\cdots$	0	$-c'_{r+1}$	$\cdots$	$-c'_n$

(1.19)

Теорема 11 (о возможности улучшения плана) Если в таблице (1.19) существует хотя бы один коэффициент $-c'_j < 0\,(j = r+1,\ldots,n)$, а в каждом столбце с таким элементом среди $a'_{ij}\,(i = 1,2,\ldots,r)$ есть хотя бы один положительный, то поменяв базис, можно перейти к другому опорному плану X'' такому, что $Z(X'') \geq Z(X')$.

Теорема 12 (признак альтернативного оптимума) Если в Z-строке симплексной таблицы (1.19), содержащей оптимальный план, среди чисел $-c'_j \geq 0\,(j = r+1,\ldots,n)$ есть равные 0, то множество оптимальных решений бесконечно.

Сформулированные теоремы позволяют проверить, является ли найденное опорное решение оптимальным, и выявить целесообразность перехода к новому опорному решению, но не дают самого алгоритма перехода.

Идея симплекс-метода: осуществлять переход от одной таблицы к другой, заменяя одну из базисных переменных на свободную до тех пор, пока не получим оптимальное решение или не установим, что функция не ограничена (теоремы 9, 10).

Если в Z-строке есть отрицательные элементы, то введение в базис соответствующих свободных переменных позволит не уменьшить значение целевой функции (теорема 11). Так как количество переменных в базисе должно остаться неизменным, то одна из переменных должна быть выведена из базиса. Таким образом, на каждом шаге симплекс-метода осуществляется преобразование базиса: одна из свободных переменных вводится в базис, а одна из базисных – выводится.

Алгоритм симплексного преобразования

б.п.	1	x_1	x_2	$\dots$	x_r	x_{r+1}	$\dots$	x_n
x_1	b'_1	1	0	$\dots$	0	$a'_{1,r+1}$	$\dots$	$a'_{1,n}$
x_2	b'_2	0	1	$\dots$	0	$a'_{2,r+1}$	$\dots$	$a'_{2,n}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
x_r	b'_r	0	0	$\dots$	1	$a'_{r,r+1}$	$\dots$	$a'_{r,n}$
Z	c'_0	0	0	$\dots$	0	$-c'_{r+1}$	$\dots$	$-c'_n$

1. Для перехода к новому опорному решению в Z-строке среди отрицательных коэффициентов $-c'_j \ (j = r+1, \dots, n)$ выбирают наибольший по абсолютной величине, пусть это будет $-c'_{j_0}$. Тем самым выбрана свободная переменная x_{j_0}, которая будет вводиться в базис. Столбец с заголовком x_{j_0} называется *разрешающим*.

Отметим разрешающий столбец.

б.п.	1	x_1	x_2	...	x_r	x_{r+1}	...	x_{j_0}	...	x_n	СО
x_1	b'_1	1	0	...	0	$a'_{1,r+1}$	...	a'_{1,j_0}	...	$a'_{1,n}$	
x_2	b'_2	0	1	...	0	$a'_{2,r+1}$	...	a'_{2,j_0}	...	$a'_{2,n}$	
...	...	...	...	...	...	...	...	...	...	...	
x_{i_0}	b'_{i_0}	0	0	...	0	$a'_{i_0,r+1}$	...	a'_{i_0,j_0}	...	$a'_{i_0,n}$	
...	...	...	...	...	...	...	...	...	...	...	
x_r	b'_r	0	0	...	1	$a'_{r,r+1}$	...	a'_{r,j_0}	...	$a'_{r,n}$	
Z	c'_0	0	0	...	0	$-c'_{r+1}$	...	$-c'_{j_0}$	...	$-c'_n$	

$$-c'_{j_0} < 0$$

2. Для выбора базисной переменной, выводимой из базиса, находятся *симплексные отношения* – отношения элементов столбца свободных членов к положительным элементам разрешающего столбца. И записывают эти отношения в специальный столбец.

Переменная, выводимая из базиса, определяется минимальным симплексным соотношением, соответствующая ей строка называется *разрешающей*. Отметим ее.

Элемент, находящийся на пересечении разрешающей строки и разрешающего столбца, называется *разрешающим элементом*, выделим его.

б.п.	1	x_1	x_2	...	x_r	x_{r+1}	...	x_{j_0}	...	x_n	CO
x_1	b'_1	1	0	...	0	$a'_{1,r+1}$	...	a'_{1,j_0}	...	$a'_{1,n}$	
x_2	b'_2	0	1	...	0	$a'_{2,r+1}$	...	a'_{2,j_0}	...	$a'_{2,n}$	
...	...	...	...	...	...	...	...	...	...	...	
x_{i_0}	b'_{i_0}	0	0	...	0	$a'_{i_0,r+1}$	...	a'_{i_0,j_0}	...	$a'_{i_0,n}$	
...	...	...	...	...	...	...	...	...	...	...	
x_r	b'_r	0	0	...	1	$a'_{r,r+1}$	...	a'_{r,j_0}	...	$a'_{r,n}$	
Z	c'_0	0	0	...	0	$-c'_{r+1}$	...	$-c'_{j_0}$	...	$-c'_n$	

3. Формируется новая симплексная таблица по следующим правилам:

– в базис вместо переменной x_{i_0} вводим переменную x_{j_0};

б.п.	1	x_1	x_2	...	x_r	x_{r+1}	...	x_{j_0}	...	x_n	CO
x_1				...			...		...		
x_2				...			...		...		
...	...	...	...	...	...	...	...	...	...	...	
x_{j_0}				...			...		...		
...	...	...	...	...	...	...	...	...	...	...	
x_r				...			...		...		
Z				...			...		...		

б.п.	1	x_1	x_2	…	x_r	x_{r+1}	…	x_{j_0}	…	x_n	CO
x_1	b'_1	1	0	…	0	$a'_{1,r+1}$	…	a'_{1,j_0}	…	$a'_{1,n}$	
x_2	b'_2	0	1	…	0	$a'_{2,r+1}$	…	a'_{2,j_0}	…	$a'_{2,n}$	
…	…	…	…	…	…	…	…	…	…	…	
x_{i_0}	b'_{i_0}	0	0	…	0	$a'_{i_0,r+1}$	…	a'_{i_0,j_0}	…	$a'_{i_0,n}$	
…	…	…	…	…	…	…	…	…	…	…	
x_r	b'_r	0	0	…	1	$a'_{r,r+1}$	…	a'_{r,j_0}	…	$a'_{r,n}$	
Z	c'_0	0	0	…	0	$-c'_{r+1}$	…	$-c'_{j_0}$	…	$-c'_n$	

– элементы разрешающей строки делятся на разрешающий элемент;

б.п.	1	x_1	x_2	…	x_r	x_{r+1}	…	x_{j_0}	…	x_n	CO
x_1				…			…		…		
x_2				…			…		…		
…	…	…	…	…	…	…	…	…	…	…	
x_{j_0}	$\dfrac{b'_{i_0}}{a'_{i_0,j_0}}$	0	0	…	0	$\dfrac{a'_{i_0,r+1}}{a'_{i_0,j_0}}$	…	1	…	$\dfrac{a'_{i_0,n}}{a'_{i_0,j_0}}$	
…	…	…	…	…	…	…	…	…	…	…	
x_r				…			…		…		
Z				…			…		…		

б.п.	1	x_1	x_2	…	x_r	x_{r+1}	…	x_{j_0}	…	x_n	СО
x_1	b'_1	1	0	…	0	$a'_{1,r+1}$	…	a'_{1,j_0}	…	$a'_{1,n}$	
x_2	b'_2	0	1	…	0	$a'_{2,r+1}$	…	a'_{2,j_0}	…	$a'_{2,n}$	
…	…	…	…	…	…	…	…	…	…	…	
x_{i_0}	b'_{i_0}	0	0	…	0	$a'_{i_0,r+1}$	…	a'_{i_0,j_0}	…	$a'_{i_0,n}$	
…	…	…	…	…	…	…	…	…	…	…	
x_r	b'_r	0	0	…	1	$a'_{r,r+1}$	…	a'_{r,j_0}	…	$a'_{r,n}$	
Z	c'_0	0	0	…	0	$-c'_{r+1}$	…	$-c'_{j_0}$	…	$-c'_n$	

– на месте остальных элементов разрешающего столбца будут стоять нули;

б.п.	1	x_1	x_2	…	x_r	x_{r+1}	…	x_{j_0}	…	x_n	СО
x_1				…			…	0	…		
x_2				…			…	0	…		
…	…	…	…	…	…	…	…	…	…	…	
x_{j_0}	$\dfrac{b'_{i_0}}{a'_{i_0,j_0}}$	0	0	…	0	$\dfrac{a'_{i_0,r+1}}{a'_{i_0,j_0}}$	…	1	…	$\dfrac{a'_{i_0,n}}{a'_{i_0,j_0}}$	
…	…	…	…	…	…	…	…	…	…	…	
x_r				…			…	0	…		
Z				…			…	0	…		

б.п.	1	x_1	x_2	...	x_r	x_{r+1}	...	x_{j_0}	...	x_n	CO
x_1	b'_1	1	0	...	0	$a'_{1,r+1}$	...	a'_{1,j_0}	...	$a'_{1,n}$	
x_2	b'_2	0	1	...	0	$a'_{2,r+1}$	...	a'_{2,j_0}	...	$a'_{2,n}$	
...	...	...	...	...	...	...	...	...	...	...	
x_{i_0}	b'_{i_0}	0	0	...	0	$a'_{i_0,r+1}$	...	a'_{i_0,j_0}	...	$a'_{i_0,n}$	
...	...	...	...	...	...	...	...	...	...	...	
x_r	b'_r	0	0	...	1	$a'_{r,r+1}$	...	a'_{r,j_0}	...	$a'_{r,n}$	
Z	c'_0	0	0	...	0	$-c'_{r+1}$	...	$-c'_{j_0}$	...	$-c'_n$	

– столбцы таблицы с заголовками базисных переменных (кроме x_{i_0}), не изменяются;

б.п.	1	x_1	x_2	...	x_r	x_{r+1}	...	x_{j_0}	...	x_n	CO
x_1		1	0	...	0		...	0	...		
x_2		0	1	...	0		...	0	...		
...	...	...	...	...	...	...	...	...	...	...	
x_{j_0}	$\dfrac{b'_{i_0}}{a'_{i_0,j_0}}$	0	0	...	0	$\dfrac{a'_{i_0,r+1}}{a'_{i_0,j_0}}$	...	1	...	$\dfrac{a'_{i_0,n}}{a'_{i_0,j_0}}$	
...	...	...	...	...	...	...	...	...	...	...	
x_r		0	0	...	1		...	0	...		
Z		0	0	...	0		...	0	...		

– все остальные элементы пересчитываются по правилу прямоугольника:

$$\begin{array}{cc} \boxed{a'_{i_0 j_0}} & a'_{i_0 j} \\ a'_{ij_0} & a'_{ij} \end{array} \qquad a''_{ij} = a'_{ij} - \frac{a'_{i_0 j} \cdot a'_{ij_0}}{a'_{i_0 j_0}}$$

Правило 3 распространяется на коэффициенты столбца свободных членов и Z-строки.

Симплексные преобразования проводят до тех пор, пока не будет получено оптимальное решение или установлена неразрешимость задачи (теоремы 9, 10).

Геометрически симплекс-метод можно проинтерпретировать следующим образом. Начальному опорному плану соответствует угловая точка многогранника решений (1.17). Шагу симплексного преобразования соответствует переход в соседнюю вершину таким образом, чтобы значение целевой функции не уменьшилось

б.п.	1	x_1	x_2	$\cdots$	x_r	x_{r+1}	$\cdots$	x_n
x_1	b'_1	1	0	$\cdots$	0	$a'_{1,r+1}$	$\cdots$	$a'_{1,n}$
x_2	b'_2	0	1	$\cdots$	0	$a'_{2,r+1}$	$\cdots$	$a'_{2,n}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
x_r	b'_r	0	0	$\cdots$	1	$a'_{r,r+1}$	$\cdots$	$a'_{r,n}$
Z	c'_0	0	0	$\cdots$	0	$-c'_{r+1}$	$\cdots$	$-c'_n$

Замечания:

1. Если таблица (1.19) соответствует оптимальному решению, а среди чисел $-c'_j$ $(j = r+1, \ldots, n)$ имеется коэффициент $-c'_{j_0}$ равный нулю, то задача имеет бесконечно много решений. Вторую вершину, соответствующую оптимальному решению, можно найти, выбрав в качестве разрешающего столбец, содержащий $-c'_{j_0}$.

2. Задачу минимизации Z можно формально заменить задачей максимизации функции (-Z). Но можно этого не делать. Признаком оптимальности опорного плана задачи минимизации является отсутствие положительных элементов в Z-строке таблицы (1.19). Для выбора разрешающего столбца выбирают максимальный среди положительных коэффициентов $-c'_j$ $(j = r+1, \ldots, n)$, а вся остальная вычислительная процедура остается прежней.

Пример 5

Решить задачу симплекс-методом и дать геометрическую интерпретацию процесса поиска оптимального решения.

$$Z = 8x_1 + 6x_2 \to \max$$
$$\begin{cases} 5x_1 + 2x_2 \le 20 \\ 6x_1 + 12x_2 \le 72 \\ x_1, x_2 \ge 0 \end{cases}$$

Решение примера 5 – в файле lecture.pdf.