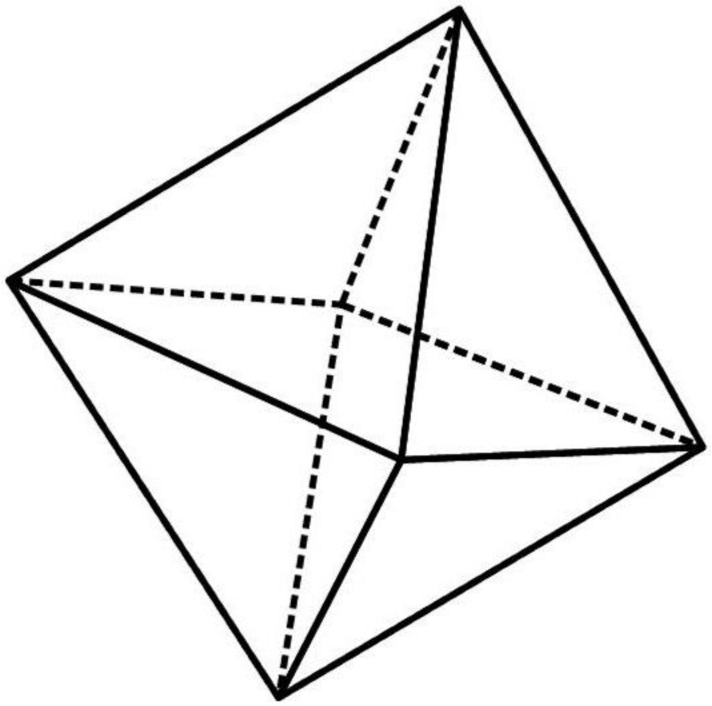
Курс:

ТЕОРИЯ ОПТИМИЗАЦИИ



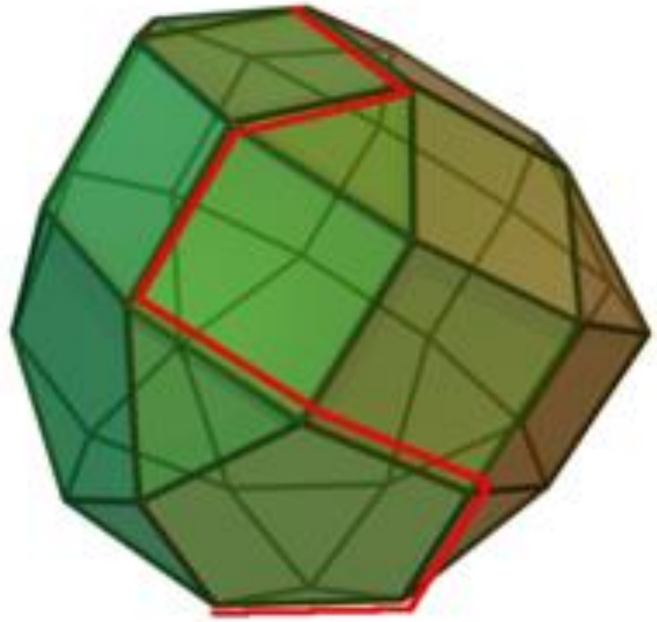
Тема 12:

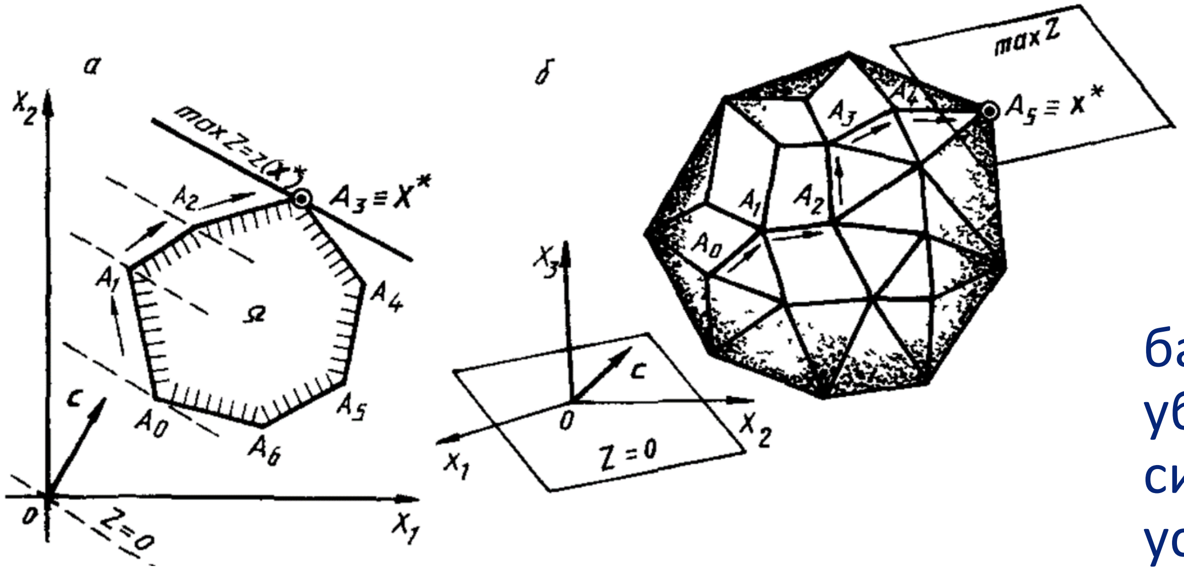
ОПИСАНИЕ СИМПЛЕКС-МЕТОДА.

ТЛЕУЖАНОВА МАНАТЖАН АШИМКУЛОВНА

Симплекс-метод

алгоритм решения оптимизационной задачи линейного программирования путём перебора вершин выпуклого многогранника в многомерном пространстве.





Сущность метода - построение базисных решений, на которых монотонно убывает линейный функционал, до ситуации, когда выполняются необходимые условия локальной оптимальности.

В работе Л.В. Канторовича «Математические методы организации и планирования производства» (1939) были впервые изложены принципы новой отрасли математики, которая позднее получила название линейного программирования.

Исторически общая задача линейного программирования была впервые поставлена в 1947 году Джорджем Бернардом Данцигом, Маршаллом Вудом и их сотрудниками в департаменте военно-воздушных сил США. В то время эта группа занималась исследованием возможности использования математических и смежных с ними методов для военных задач и проблем планирования. В дальнейшем для развития этих идей в ВВС была организована исследовательская группа под названием Project SCOOP. Первое успешное решение задачи линейного программирования на ЭВМ SEAC было проведено в январе 1952 года

Задача линейного программирования

Задача линейного программирования состоит в том, что необходимо максимизировать или минимизировать некоторый линейный функционал на многомерном пространстве при заданных линейных ограничениях.

Заметим, что каждое из линейных неравенств на переменные ограничивает полупространство в соответствующем линейном пространстве. В результате все неравенства ограничивают некоторый выпуклый многогранник (возможно, бесконечный), называемый также полиэдральным комплексом.

Уравнение $W(x) = c$, где $W(x)$ — максимизируемый (или минимизируемый) линейный функционал, порождает гиперплоскость $L(c)$. Зависимость от с порождает семейство параллельных гиперплоскостей.

Тогда экстремальная задача приобретает следующую формулировку — требуется найти такое наибольшее c, что гиперплоскость $L(c)$ пересекает многогранник хотя бы в одной точке.

Заметим, что пересечение оптимальной гиперплоскости и многогранника будет содержать хотя бы одну вершину, причём их будет более одной, если пересечение содержит ребро или k-мерную грань. Поэтому максимум функционала можно искать в вершинах многогранника.

Принцип симплекс-метода состоит в том, что выбирается одна из вершин многогранника, после чего начинается движение по его рёбрам от вершины к вершине в сторону увеличения значения функционала. Когда переход по ребру из текущей вершины в другую вершину с более высоким значением функционала невозможен, считается, что оптимальное значение с найдено.

Последовательность вычислений симплекс-методом можно разделить на две основные фазы:

- нахождение исходной вершины множества допустимых решений,
- последовательный переход от одной вершины к другой, ведущий к оптимизации значения целевой функции.

Усиленная постановка задачи

Рассмотрим следующую задачу линейного программирования:

$$c^T x \to \max, Ax \leqslant b, x \geqslant 0, b \geqslant 0.$$

Теперь поставим эту задачу в эквивалентной усиленной форме. Необходимо максимизировать Z, где:

$$\begin{bmatrix} 1 & -c^T & 0 \\ 0 & A & E \end{bmatrix} \begin{bmatrix} Z \\ x \\ x_s \end{bmatrix} = \begin{bmatrix} 0 \\ b \end{bmatrix}, x, x_s \geqslant 0$$

Здесь x — переменные из исходного линейного функционала, x_s — новые переменные, дополняющие старые таким образом, что неравенство переходит в равенство, c — коэффициенты исходного линейного функционала, Z — переменная, которую необходимо максимизировать.

Полупространства $x \geq 0$ и $x_s \geq 0$ в пересечении образуют многогранник, представляющий множество допустимых решений. Разница между числом переменных и уравнений даёт нам число степеней свободы. Проще говоря, если мы рассматриваем вершину многогранника, то это число рёбер, по которым мы можем продолжать движение. Тогда мы можем присвоить этому числу переменных значение 0 и назвать их «небазисными».

Остальные переменные при этом будут вычисляться однозначно и называться «базисными». Сам же набор этих переменных называется базисом. Полученная точка будет вершиной в пересечении соответствующих небазисным переменным гиперплоскостей.

Для того, чтобы найти т.н. начальное допустимое решение (вершину, из которой мы начнём движение), присвоим всем изначальным переменным x значение 0 и будем их считать небазисными, а все новые будем считать базисными.

При этом начальное допустимое решение вычисляется однозначно :

$$x_{si} = b_i$$

Алгоритм

Теперь приведём шаги алгоритма. На каждом шаге мы будем менять множества базисных и небазисных векторов (двигаться по рёбрам), и матрица будет иметь следующий вид:

$$\begin{bmatrix} 1 & c_B^T B^{-1} A - c^T & c_B^T B^{-1} \\ 0 & B^{-1} A & B^{-1} \end{bmatrix} \begin{bmatrix} Z \\ x \\ x_s \end{bmatrix} = \begin{bmatrix} c_B^T B^{-1} b \\ B^{-1} b \end{bmatrix},$$

где c_B — коэффициенты вектора c, соответствующие базисным переменным (переменным x_s соответствуют 0), B — столбцы $[\mathbf{AE}]$, соответствующие базисным переменным. Матрицу, образованную оставшимися столбцами обозначим D. Почему матрица будет иметь такой вид поясним в описании шагов алгоритма.

Первый шаг.

Выбираем начальное допустимое значение, как указано выше. На первом шаге B — единичная матрица, так как базисными переменными являются x_s. c_B — нулевой вектор по тем же причинам.

Второй шаг

Покажем, что в выражении $(c_B^T B^{-1} A - c^T)x + (c_B^T B^{-1})x_s$ только небазисные переменные имеют ненулевой коэффициент. Заметим, что из выражения $Ax + x_s = b$ базисные переменные однозначно выражаются через небазисные, так как число базисных переменных равно числу уравнений. Пусть x' — базисные, а x'' — небазисные переменные на данной итерации. Уравнение $Ax + x_s = b$ можно переписать, как $Bx' + Dx'' = b$. Умножим его на B^{-1} слева: $x' + B^{-1}Dx'' = B^{-1}b$. Таким образом мы выразили базисные переменные через небазисные, и в выражении $B^{-1}Ax + B^{-1}x_s$, эквивалентному левой части равенства, все базисные переменные имеют единичные коэффициенты. Поэтому, если прибавить к равенству $Z - c^T x = 0$ равенство $c_B^T B^{-1}Ax + c_B^T B^{-1}x_s$, то в полученном равенстве все базисные переменные будут иметь нулевой коэффициент — все базисные переменные вида x сократятся, а базисные переменные вида x_s не войдут в выражение $c_B^T B^{-1}x_s$.

Выберем ребро, по которому мы будем перемещаться. Поскольку мы хотим максимизировать Z, то необходимо выбрать переменную, которая будет более всех уменьшать выражение $(c_B^T B^{-1} A - c^T)x + (c_B^T B^{-1})x_s$.

Для этого выберем переменную, которая имеет наибольший по модулю отрицательный коэффициент[3]. Если таких переменных нет, то есть все коэффициенты этого выражения неотрицательны, то мы пришли в искомую вершину и нашли оптимальное решение. В противном случае начнём увеличивать эту небазисную переменную, то есть перемещаться по соответствующему ей ребру. Эту переменную назовём *входящей*.

Третий шаг

Теперь необходимо понять, какая базисная переменная первой обратится в ноль по мере увеличения входящей переменной. Для этого достаточно рассмотреть систему:

$$\begin{bmatrix} B^{-1}A & B^{-1} \end{bmatrix} \begin{bmatrix} x \\ x_s \end{bmatrix} = B^{-1}b$$

При фиксированных значениях небазисных переменных система однозначно разрешима относительно базисных, поэтому мы можем определить, какая из базисных переменных первой достигнет нуля при увеличении входящей. Эту переменную назовем *выходящей*. Это будет означать, что мы натолкнулись на новую вершину. Теперь входящую и выходящую переменную поменяем местами — входящая «войдёт» в базисную, а выходящая из них «выйдет» в небазисные. Теперь перепишем матрицу B и вектор c_B в соответствии с новыми наборами базисных и небазисных переменных, после чего вернёмся ко второму шагу. x''

Поскольку число вершин конечно, то алгоритм однажды закончится. Найденная вершина будет являться оптимальным решением.