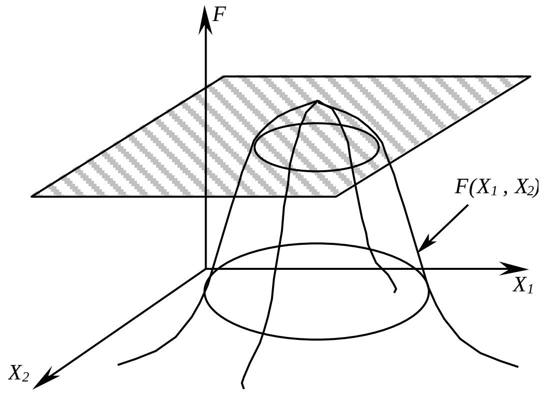
Лекция 12

Задачи нелинейного и квадратичного программирования

Нелинейное программирование (НЛП).



НЛП – это такая задача математического программирования, когда-либо целевая функция, либо ограничения, либо они вместе представляют собой нелинейные функции. Несмотря на то, что ограничения могут быть наложены на сами переменные, термин НЛП применяется только тогда, когда X – действительные числа. Решение задач НЛП намного сложнее решения задач ЛП, при прочих равных условиях. Рассмотрим особенности задач НЛП, а именно то, где может находиться оптимальное решение. Для этого будем рассматривать целевую функцию двух переменных. В НЛП используют геометрический образ целевой функции, как некоторой поверхности, определённый на множестве управляемых переменных.

Методы НЛП разработаны только для выпуклых функций, причём так как используются производные, считается, что функции должны быть дифференцируемы.

Методы оптимизации нелинейных функций без ограничений.

Если ограничения в задаче НЛП отсутствуют, то применяют все методы оптимизации нелинейных функций. Причём, если в задаче с ограничениями, они находятся внутри области, то решение можно найти этими методами. Говорят, что ограничения не являются активными и их можно отбросить. Рассмотрим наиболее широко применяемые методы.

1) **Классический градиентный метод**.

Все эти методы принадлежат к поисковым методам, т. е. оптимальное решение находится не сразу, а последовательным движением от точки к точке и заканчивается в оптимальной точке. Начинаем с начальной точки, её выбор – чтобы была ближе к оптимальной точке. Для её выбора и направления движения используют понятие градиента. Градиент – вектор, показывающий направление наибыстрейшего изменения функции. Градиент направлен по нормали к поверхности целевой функции, а в плоскости X_1OX_2 – по нормали к линиям уровня. Алгоритм: $X^{k+1} = X^k \pm \lambda^k g\left(X^k\right)$.

Каждая последующая точка определяется как предыдущая ± длина градиента в предыдущей точке шага (“+” – задача на максимум; “ – ” – задача на минимум). Если максимуму или минимум гладкий (большой коэффициент «тупизны»), градиент у нас уменьшается (в вершине градиент равен нулю) по модулю по мере приближения к максимуму и шаг уменьшается. Но при выборе λ надо иметь в виду следующее: если $\lambda-$мало, то идти будем долго, а если $\lambda-$большое, то можем перепрыгнуть.

В некоторых алгоритмах $\lambda = const$, в более сложных λ зависит от k. Правила остановки могут быть различными, например следующее:

$$\left|\max_r g_{Xr}\right| \le \varepsilon$$

2) **Покоординатный градиентный метод (метод координат спуска или подъёма)**.

Из некоторой начальной точки движемся вдоль некоторой координаты в сторону, где проекция градиента положительна. Движемся до тех пор, пока производная по этой координате не будет равна нулю. Затем движение идёт по второй координате аналогичным способом, и т. д. (если для двух координат, то опять по первой и по второй).

3) **Метод наискорейшего спуска (подъёма).**

В этом алгоритме из некоторой начальной точки движение осуществляется вдоль направления градиента до тех пор, пока производная по этому направлению не будет равна нулю. Далее из этой точки определяем градиент и т. д. Отличие здесь в том , что длина шага X^k определяется из условия, чтобы обеспечить:

$$\min_{\lambda>0} F\left[X^k \pm \lambda \times g\left(X^k\right)\right]$$

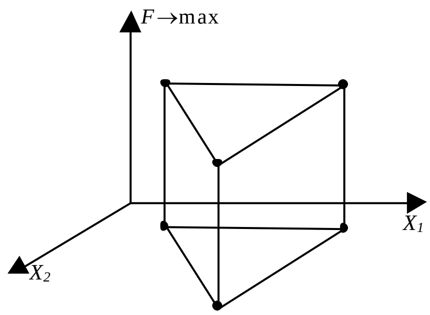
Второй и третий методы называются <u>алгоритмами с длинным шагом</u>. Кроме этого существуют методы, использующие вторую производную, например, метод Ньютона:

$$X^{k+1} = X^k + \left[F''\left(X^k\right)\right]^{-1} g\left(X^k\right)$$

Метод поиска минимальной, не использующий понятие производной.

Метод Нелдера-Мида.

Современные методы поиска минимальной (максимальной) нелинейной функции чрезвычайно разнообразны. Одним из наиболее эффективных является метод Нелдера-Мида. Идея метода состоит в том, что, вычисляется значение целевой функции в вершинах сначала правильного, а затем деформируемого многогранника. Многогранник – некоторое правильное тело в $n-$мерном пространстве.



Эта правильная фигура называется <u>симплексом</u>. В задачах для двух переменных, это правильный треугольник. Затем сравниваются значения целевых функций в вершине, а затем выполняются операции:

1) <u>Отражение</u> – поворот симплекса через одну из сторон;
2) <u>Растяжение</u> – если идём в правильном направлении;
3) <u>Редукция</u> – возврат назад, если перескочили максимум;

4) <u>Сжатие</u> – уменьшение сторон, чтобы движение было с более мелким шагом.

Алгоритм:

1) Обозначим $X_h^{(k)}$ – самая худшая точка; $X_l^{(k)}$ – самая лучшая точка.

$$F\left(X_h^{(k)}\right)=F_{\max}^{(k)};\quad F\left(X_l^{(k)}\right)F_{\min}^{(k)}$$

$X_0^{(k)}$ – центр тяжести всех вершин, исключая X_n. Координаты центра тяжести определяются:

$$X_0^{(k)}=\frac{1}{n}\left[\left(\sum_{i=1}^{n+1}X_{ij}^{(k)}\right)-X_{nj}^{(k)}\right];\quad j=\overline{1,n}$$

2) Затем отражаем, и получаем точку: $X_\alpha^{(k)}=X_0^{(k)}+\alpha\left(X_0^{(k)}-X_n^{(k)}\right)$.

Здесь α – коэффициент отражения; в обычном случае он равен единице.

- Если $F\left(X_\alpha^{(k)}\right)\le F\left(X_l^{(k)}\right)$, то вектор $X_\alpha^{(k)}-X_0^{(k)}$ растягивается в γ раз и получаем:

$$X_b^{(k)}=X_0^{(k)}+\gamma\left(X_n^{(k)}-X_n^{(k)}\right);\quad \gamma>\alpha$$

Если для полученной точки X_b $F\left(X_b^{(k)}\right)<F\left(X_l^{(k)}\right)$, то $X_n^{(k)}$ заменяется на $X_b^{(k)}$, и переходим к пункту 2

- Если $F\left(X_\alpha^{(k)}\right)>F\left(X_i^{(k)}\right);\ i\ne h$, то вектор $X_n^{(k)}-X_0^{(k)}$ сжимается в β раз и получаем точку:

$$X_i^{(k)}=X_0^{(k)}+\beta\left(X_0^{(k)}-X_n^{(k)}\right);\quad 0<\beta<1$$

Затем точка $X_n^{(k)}$ заменяется на точку $X_l^{(k)}$, и так же переходим к пункту 2

- Если $F\left(X_\alpha^{(k)}\right)>F\left(X_n^{(k)}\right)$, то все векторы $X_i^{(k)}-X_l^{(k)}$ уменьшаются в λ раз:

$$X_i^{(k)}=X_l+\frac{1}{\lambda}\left(X_i^{(k)}-X_l^{(k)}\right);\quad i=\overline{1,n+1}.\quad i-\text{номер вершины.}$$

Затем возвращаемся к пункту 1.

Правило остановки:

$$\sqrt{\frac{1}{n+1}\sum_{i=1}^{n+1}\left[F\left(X_i^{(r)}\right)-F\left(X_0^{(r)}\right)\right]^2}\le\varepsilon$$

Задачи НЛП с ограничениями-равенствами.

В НЛП особо рассматриваются задачи с ограничениями-равенствами. Это так называемые задачи на условный экстремум. Решение такой задачи производится с использованием <u>функции Лагранжа</u>. Это функция позволяет построить новую целевую функцию, которая бы в виде <u>штрафа</u> учитывала ограничения:

$$\Phi(X,\lambda)=F(X)+\sum_{i=1}^{m}\lambda_i(\varphi_i(X)-b_i)\quad(1)$$

$F(X)$ – целевая функция.

Таким образом, задача сводится к минимизации функции Лагранжа без ограничений. Для её решения используется классический приём записывается <u>условие существования экстремума</u>:

$$\begin{cases} \dfrac{\partial \Phi}{\partial X_j} = \dfrac{\partial F}{\partial X_j} + \sum\limits_{i=1}^{m} \left(\lambda_i \dfrac{\partial \varphi_i}{\partial X_j} \right) = 0, \quad j = \overline{1,n}; \quad (2) \\ \dfrac{\partial \Phi}{\partial \lambda_i} = \varphi_i(X) - b_i = 0, \quad i = \overline{1,m}; \quad (3) \end{cases}$$

λ_i – множитель Лагранжа.

Выражения (2) и (3) дают точки подозрительные на экстремум и их ещё надо проверить с помощью достаточных признаков экстремума. Рассмотрим некоторые из этих признаков.

1) Пусть X^* – критическая точка подозрительная на экстремум полученная с помощью выражений (2) или (3). Рассмотрим матрицу Гесса – матрицу вторых производных:

$$H = \frac{\partial^2 \Phi}{\partial X_i \partial X_j}$$

Если эта матрица определена положительно, то точка X^* – минимум, а если определена отрицательно, то максимум. Если же матрица Гесса знаконеопределена, то в этой точке экстремума нет, а если полуопределена, то это признак не даёт ответа на вопрос.

2) Пусть D_k – главный определитель матрицы Гесса k – го порядка.

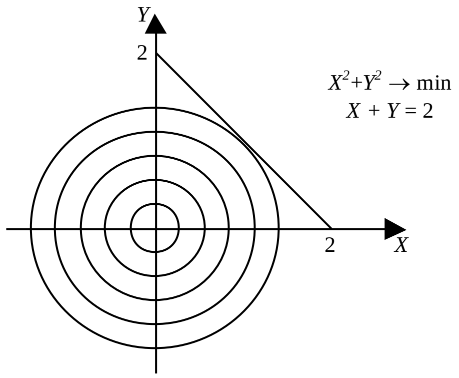
- Если $D_1 > 0, D_2 > 0, D_3 > 0, \ldots$ – это достаточный признак минимума;
- Если $D_1 < 0, D_2 > 0, D_3 < 0, \ldots$ – достаточный признак минимума;
- Если же в определителях знаки $\geq$ и $\leq$, то это необходимый, а не достаточный признак и вопрос об экстремуме не решается.

3) Наиболее сильным достаточным признаком является следующий. Составим расширенную матрицу Гесса:

$$H^{\beta} = \begin{pmatrix} 0 & Q \\ Q' & H \end{pmatrix}; \; Q = \begin{pmatrix} \dfrac{\partial \varphi_1}{\partial X_1} & \dfrac{\partial \varphi_1}{\partial X_2} & \cdots & \dfrac{\partial \varphi_1}{\partial X_n} \\ \dfrac{\partial \varphi_2}{\partial X_1} & \dfrac{\partial \varphi_2}{\partial X_2} & \cdots & \dfrac{\partial \varphi_2}{\partial X_n} \\ \cdots & \cdots & \cdots & \cdots \\ \dfrac{\partial \varphi_n}{\partial X_1} & \dfrac{\partial \varphi_n}{\partial X_2} & \cdots & \dfrac{\partial \varphi_n}{\partial X_n} \end{pmatrix}$$

Q' – транспонированная матрица.

Признак: точка X^* соответствует максимуму если начиная с главного определителя порядка $m+1$, последние $n-m$ определителей матрицы H образуют знакопеременный ряд, а знак определителя стоящего на $n-m$ – ом месте от конца должен быть:



$$(-1)^{m+1} \to \max$$
$$(-1)^{m} \to \min$$

Геометрически задача на условный экстремум сводится к тому, что решение находится в той точке, где линия (поверхность) определяющая функцию (систему) ограничений, касается какой-нибудь линии уровня.

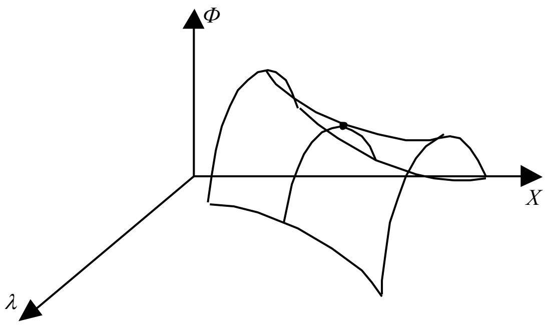
Квадратичное программирование.

Задачей КП называется такая задача НЛП, когда целевая функция представляет собой сумму линейной и квадратичной формы (переменные не старше второй степени), а все ограничения линейные, т. е. эта задача на одну ступень выше ЛП и для решения таких задач используется симплекс-метод.

В нелинейном программировании до конца изучен вопрос существования решения только для задачи выпуклого программирования. Фундаментом является теорема Куна-Такера: если функции $F(x)$ и $\varphi(x)$ выпуклые, то решение задачи НЛП находится в седловой точке для функции Лагранжа, т. е.:

$$F(x) \to \min; \qquad \varphi_i(x) \le 0, \;\; i=\overline{1,m}; \qquad \Phi(x,\lambda) = F(x) + \sum_{i=1}^{m} \lambda_i \varphi_i(x)$$

Поэтому, составляя соответствующую функцию Лагранжа необходимо найти её седловую точку. Геометрически это можно нарисовать если X – одна координата и λ – один коэффициент:



$$\Phi(X^*,\lambda) \le \Phi(X^*,\lambda^*) \le \Phi(X,\lambda)$$

Эта точка должна удовлетворять системе следующих неравенств:

$$\left(\frac{\partial \Phi}{\partial X_j}\right)_{X^*,\lambda^*} \le 0; \qquad \left.\begin{matrix} X_j^*\left(\dfrac{\partial \Phi}{\partial X_j}\right)_{X^*,\lambda^*} = 0 \\ X_j^* \ge 0, \;\; j=\overline{1,n} \end{matrix}\right\}; \qquad \left.\begin{matrix} \lambda_i^*\left(\dfrac{\partial \Phi}{\partial \lambda_j}\right) = 0 \\ \lambda_i^* \ge 0, \;\; i=\overline{1,m} \end{matrix}\right\}; \qquad \left(\frac{\partial \Phi}{\partial \lambda_i}\right)_{X^*,\lambda^*} \ge 0$$

Итак, точка Куна-Такера позволяет записать условие Куна-Такера и найти седловую точку.

Решение задач квадратичного программирования методом Баранкина-Дорфмана.

Задачей КП называется следующая задача:

$$F(X)=\sum_{j=1}^{n} p_j X_j+\sum_{k=1}^{n}\sum_{j=1}^{n}\left(C_{kj} X_k X_j\right)\rightarrow \min$$

$$\varphi_i(X)=\sum_{j=1}^{n} a_{ij} X_j-b_i \leq 0,\quad i=\overline{1,m},\quad j=\overline{1,n},\quad X_j \geq 0$$

В матричном виде: пусть $P, X, B-$ векторы-столбцы:

$$F=p'X+X'CX\rightarrow \min,\quad AX<b,\quad X\geq 0$$

$C-$ матрица квадратичной формы, которая должна быть симметрична и положительно-полуопределена.

Метод Баранкина-Дорфмана непосредственно основан на применении теоремы Куна-Такера, поэтому условие Куна-Такера в матричной форме для КП выглядит следующим образом:

$$\Phi(X\lambda)=\underbrace{p'X+X'CX}_{F(X)}+\underbrace{\lambda(AX-b)}_{штраф};\qquad \frac{\partial\Phi}{\partial X}=V=p'+2CX+A'\lambda;\qquad -\frac{\partial\Phi}{\partial\lambda}=Y=-AX+b$$

Тогда условие Куна-Такера можно записать в следующем виде:

$$\left.\begin{array}{c}\left.\begin{array}{r} AX+V=b \\ 2CX-V+A'\lambda=-p \\ X\geq 0; Y\geq 0; V\geq 0; \lambda\geq 0\end{array}\right\}\ (*) \\ XV+Y\lambda=0\ \ (\#)\end{array}\right\}-\textit{условие Куна-Такера}$$

Неизвестными являются (X, Y, V, λ).

$(*)-$ система линейных уравнений относительно неизвестных; решить её, значит найти решение задачи ЛП. $(\#)-$ нелинейное условие, и поэтому задача сводится к нахождению точки в допустимой области, чтобы выполнилось $(\#)$. Введём новый вектор $Z=(X, Y, V, \lambda)$ и $\tilde{Z}=(V, \lambda, X, Y)$.

$$XV+Y\lambda=\frac{1}{2} Z\times\tilde{Z}$$

Окончательно условие Куна-Такера будет выглядеть так:

$$\begin{pmatrix} A & E & 0 & 0 \\ 2C & 0 & -E & A'\end{pmatrix}\times Z'=\begin{pmatrix} b \\ -p\end{pmatrix},\quad Z\geq 0\qquad (1);\qquad\qquad T(Z)=Z\times\tilde{Z}=0\qquad (2)$$

Метод Баранкина-Дорфа заключается в следующем: находится допустимый вектор Z, удовлетворяющая выражению (1) и проверяется условие (2). Далее выбирается новое базисное решение, причём оно выбирается так, чтобы величина $T(Z)$ всё время уменьшалась. Для этого используется модифицированная симплекс-таблица, в которой генеральный элемент находится минимизацией выпуклой функции $T(Z)$:

$$T(Z)=Z\times\tilde{Z}\rightarrow \min\qquad (3)$$

т. е. мы решаем задачи (1) и (3), а не (1) и (2). В симплекс-таблице записывается в качестве базисных переменных, все переменные. Запишем строку базисных переменных:

$$Z_g=d_{g_0}+\sum_{h=1}^{n} d_{g_h}\times t_h,\quad g=\overline{1,2N}$$

где $2N=2(n+m)$ и t_h- свободные переменные.

Если строка соответствует свободным переменным, то в строке одна единица и остальные нули. Для выбора генерального элемента используются следующие элементы, которые записываются в дополнительные строки:

$$\alpha_j = \sum_{i=1}^{N} \left(d_{ij} \times d_{i+N,0} + d_{i+N,j} \times d_{i0} \right) \qquad \beta_j = 2\sum_{i=1}^{N} d_{ij} \times d_{i+N,j} \qquad \Theta_j = \min_g \frac{d_{g_0}}{\left| d_{g_j} \right|}, \text{при } d_{g_j} < 0 \quad (\arg\min = r)$$

$$K_j = 2\alpha_j + \Theta_j \beta_j$$

Можно показать, что новое значение $T_j = T + \Theta_j K_j$: По определению $\Theta_j > 0$, поэтому K_j должно быть меньше нуля, так как хотим, что бы оно было меньше T_j. Рассмотрим величину K_j: величина β_j –вторая производная функции $T(Z)$, которая выпукла вниз, и поэтому $\beta_j > 0$, значит, надо выбрать тот столбец для которого $\alpha_j < 0$, а строку надо выбрать ту, для которой вычисляется величина Θ.

Пример:

$$F(x) = -6x_1 + 2x_1^2 - 2x_1x_2 + 2x_2^2 \to \min$$

$$x_1 + x_2 \le 2$$

$$x_1 \ge 0$$

$$x_2 \ge 0$$

Запишем все матрицы, которые нам нужны:

$$A = (1,1); \quad B = 2; \quad C = \begin{pmatrix} 2 & -1 \\ -1 & 2 \end{pmatrix}; \quad p = \begin{pmatrix} -6 \\ 0 \end{pmatrix}; \quad n = 2; \quad m = 1; \quad N = 3;$$

Запишем условие Куна-Такера, определяемое выражением (1):

$$\begin{pmatrix} \begin{pmatrix} 1 & 1 \end{pmatrix} & 1 & \begin{pmatrix} 0 & 0 \end{pmatrix} & 0 \\ \begin{pmatrix} 4 & -2 \\ -2 & 4 \end{pmatrix} & \begin{pmatrix} 0 \\ 0 \end{pmatrix} & \begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix} & \begin{pmatrix} 1 \\ 1 \end{pmatrix} \end{pmatrix} \times \begin{pmatrix} x_1 \\ x_2 \\ y_1 \\ V_1 \\ V_2 \\ \lambda_1 \end{pmatrix} = \begin{pmatrix} 2 \\ 6 \\ 0 \end{pmatrix}$$

$$x_1 + x_2 + y_1 = 2$$
$$4x_1 - 2x_2 - V_1 + \lambda_1 = 6$$
$$-2x_1 + 4x_2 - V_2 + \lambda_1 = 0$$

Таблица 1

	1	X_1	X_2	V_1
X_1		1		
X_2			1	
Y_1	2	-1	-1	
V_1				1
V_2	6	-6	6	1
λ_1	6	-4	2	1
α_j	24	-14	4	2
β_j		8		
Θ_j		1		
K_j		-20		

Чтобы записать симплекс-таблицу надо выделить базис. Для получения первого базисного решения используется любой метод, например, метод Гаусса. Можно использовать метод, который использует симплекс-процедуры. Для любых задач можно и подобрать первое базисное решение. Мы подберём такое, чтобы сразу получить опорное решение:

$$Y_1 = 2 - X_1 - X_2$$
$$X_1 = 6 - 4X_1 + 2X_2 + V_1$$
$$V_2 = 6 - 6X_1 + 6X_2 + V_1$$

Запишем симплекс-таблицу по методу Баранкина-Дорфмана – *таблица 1*.

Таблица 2

	1	V_2	X_2	V_1
X_1	1	-1/6	1	1/6
X_2			1	
Y_1	1	1/6	-2	-1/6
V_1				1
V_2		1		
λ_1	2	2/3	-2	1/3
α_j	4	1	-6	1
β_j			8	
Θ_j			1/2	
K_j			-8	

<u>Симплекс преобразования</u>: строку умножаем на $-\lambda$, а столбец на λ. Порядок строк нарушать нельзя.

<u>Недостаток метода</u>: иногда встречаются задачи, когда все $K_j > 0$. Значит мы будем переходить в новую вершину и значение будет увеличиваться, т .е. в соседних вершинах значение больше (мы идём по соседним вершинам в симплекс-методе). В этом случае идём на временное ухудшение T, т .е. идём через «**мёртвую зону**». В алгоритме Франца-Вольфа это учтено.

Метод проекции градиента для решения задач НЛП.

Этот метод предусматривает движение от точки к точке внутри области допустимых решений одним из градиентных методов. Но при выходе точки за пределы ограничений производится процедура возврата очередного приближения на допустимое множество D. Т. е. если была задача

$$X^{k+1} = X^k - \alpha_k g\left(X^k\right), \quad X \in D, \quad X^{k+1} \notin D$$

то **проекция** – ближайшая точка множества D, лежит на границе области D. Точка $Z_D = P_D(Z)$ называется проекцией точки Z на область D, если расстояние $\rho(Z_D, Z) = \min\limits_{Z \in D} \rho(X, Z)$. Очевидно, что если точка $Z \in D$, то проекция совпадает с Z. Таким образом, в методе проекции градиента любая последующая точка вычисляется как:

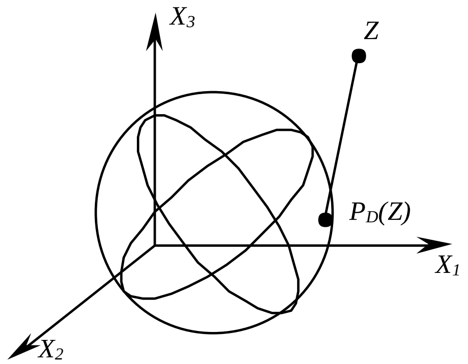
$$X^{(k+1)} = P_D\left[X^k - \alpha_k g\left(X^k\right)\right]$$

α_k – коэффициент определяющий длину шага, который определяется так же, как и в методе наискорейшего спуска. Такую задачу можно решить методом «Золотого сечения Ньютона».

Если $g(X)$ на множестве D удовлетворяет условию Липтица, означающего ограничения но крутизну изменения функции, т. е. $\|g(X') - g(X'')\| \le L\|X' - X''\|$, тогда полагают , что $\alpha_k = \alpha$, которая выбирается как любое число из интервала $(0, 2/L)$ если известна минимальная константа Липтица, то α берут $1/L$.

Определение проекции точки $Z \notin D$ является самостоятельной задачей НЛП.

$$\sum_{i=1}^{n}\left(X_j - Z_j\right)^2 \to \min, \quad X \in D$$



Если D – область определённая линейными ограничениями, то это будет задачей квадратичного программирования. Во многих случаях определение проекции возможно из практических соображений, например, если D – шар.

Иногда, как в примере с шаром, нахождение кратчайшего расстояния приводит к задаче на условный экстремум:

$$\Phi(X, \lambda) = \sum_{j=1}^{n}\left(X_j - Z_j\right)^2 + \lambda\sum_{j=1}^{n}\left(X_j - R_0\right)^2; \qquad \frac{\partial \Phi}{\partial X_j} = 0; \qquad \frac{\partial \Phi}{\partial \lambda_j} = 0$$

В случае с шаром:

$$P_D(Z) = \begin{cases} Z, \quad \sum\limits_{j=1}^{n} Z_j^2 \le R_0^2; \ (Z \in D) \\ \dfrac{R_0 Z}{\sqrt{\sum\limits_{j=1}^{n} z^2}}, \ \sum\limits_{j=1}^{n} Z_j^2 > R_0^2; \ (Z \notin D) \end{cases}$$

Методы возможных направлений Гаус-Зойтендейка.

В методах возможных направлений переход от точки X^k к точке X^{k+1} осуществляется по направлению S_k, необязательно вдоль градиента. При этом движение вдоль S_k должно быть таким, что бы новая точка принадлежала области D. Направление, удовлетворяющее этому условию, называется **возможным** или **допустимым**:

$$X^{k+1} = \left(X^k - \lambda S^k\right) \in D \quad (1)$$

Таких направлений множество; среди возможных направлений выбирают такое, для которого скалярное произведение градиентов в точке k и этого направления меньше нуля. Такое направление называется **подходящим**. Таких направлений в общем случае так же множество. Зойтендейк предложил выбирать то, которое максимально уменьшает значение целевой функции:

$$\left(g\left(X^k\right), S^k\right) \to \min$$

$$\varphi_i\left(X^k + S^k\right) \le 0, \quad i = I\left(X^k\right) \text{– ограничение нашей задачи}$$

В общем случае, на каждом шаге анализируется не только значение градиентов целевой функции, но и значение градиентов активных ограничений. **Активными** называются те ограничения, которые находятся вблизи k – ой точки.

В какой-то начальной точке X^0 определяем множество индексов $I(X)$, таких что $\varphi_i(X) = 0$. Пусть $\vec{g}_0(X)$ – градиент целевой функции, $\vec{g}_i(X)$ – градиент функции ограничений, i – номер ограничения. Тогда ищут $\vec{S}$, такое, что $\left(g_i\left(X^0\right), S\right) < 0$. Тогда надо найти S^0, такое, что $X^0 + \lambda S^0 \in D$. Для этого сначала определяют $\lambda' = \min\limits_{i \notin I\left(X^0\right)} d_i$, $\varphi_i\left(X^0 + d_i S\right) = 0$, где d_i – корень уравнения $\varphi_i\left(X^0 + d_i S\right) = 0$. λ' выбирается из условия прохода до противоположной границы. Это разумно, когда точка вне области. Затем вычисляется λ'', где $\lambda'' = \min\limits_{i \in I\left(X^0\right)} C$, здесь C – корень уравнения $g_i\left(X^0 + C_i S\right) = 0 \quad (*)$. Т. е. λ'' определяет длину шага до точки, где градиент равен нулю. Если нет решения выражения (*), то $\lambda'' = \infty$, и λ' выбирается из условия не выхода из области, а λ'' выбирается из условия попадания в экстремум.

Кроме этого, при решении вспомогательной задачи $\left(g\left(X^k\right) \times S\right) \to \min$, $\varphi_i\left(X^k + S\right) \le 0$, $i \in I$, используются различные условия нормировки:

$N1$: $S' \times S \le 1$

$N2$: $-1 \le S_j \le 1$, $j = \overline{1, n}$

$N5$: $\varphi_i\left(X^k + S\right) \le 0$, $\forall i$ – самое распространённое.

Метод Зойтендейка для выпуклой задачи не гарантирует сходимость за конечное число шагов; для квадратичной задачи гарантируется сходимость за конечное число шагов с применением условия сопряжённости градиентов.