ЛЕКЦИЯ № 5

Математические основы оптимизации

(продолжение)

3. Задачи выпуклого программирования.

3.1. Формулировка задачи

Пусть f: $X \to Y; X \subset E^n, Y \subset E^1$,

g: $X \to Z; X \subset E^n, Z \subset E^m$,

$f(x) \in C^1(X),\ g(x) \in C^1(X)$,

$$x^e = Arg\min f(x), x_i \ge 0, i = 1..n$$

.

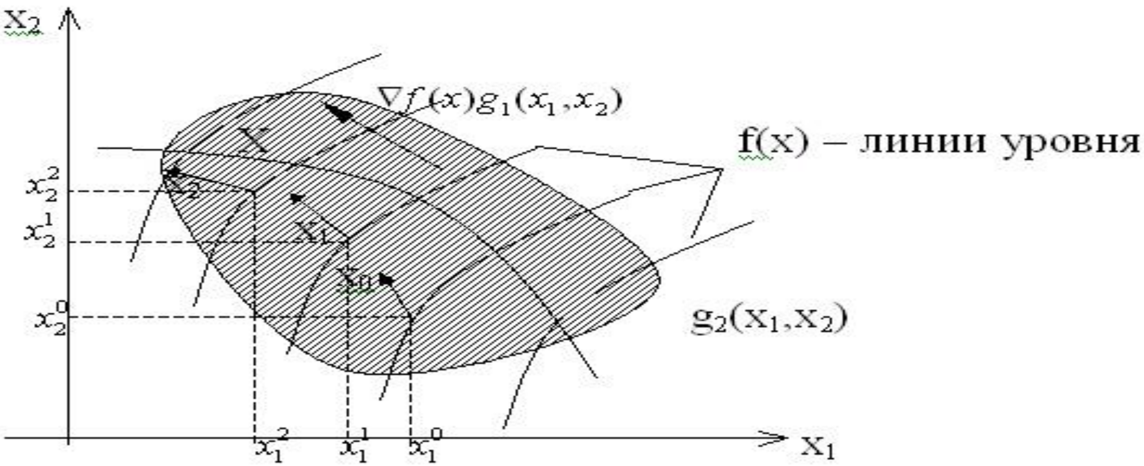
$g_j(x) = 0,\ j = 1..\ell$ – ограничения типа равенств.

$g_j(x) \le 0, k = (\ell + 1)..m$ – ограничения типа неравенств.

Обозначим множество индексов: $K = \{l+1, m\}, k \in K$.

3. Задачи выпуклого программирования.

3.1. Активные ограничения.



В процессе решения задачи происходит перемещение текущей точки из заданной начальной точки в конечную точку. При этом возможен выход на границу области допустимых решений X. Такие нарушенные ограничения носят название активных.

Определение множества индексов активных ограничений:

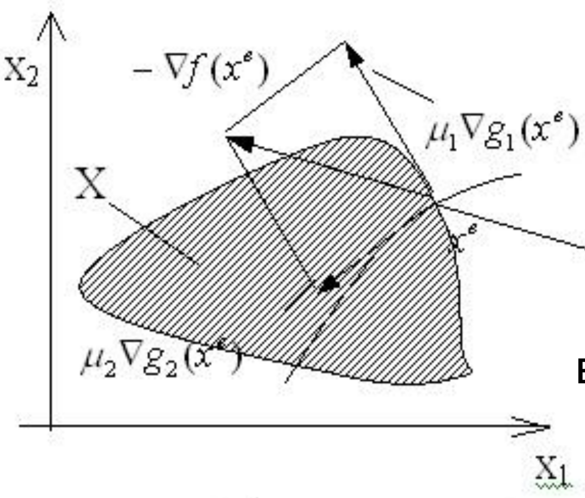
$$K_A = \left\{ k \in K \middle| (\forall g_k(x) \le 0) \wedge (\exists g_{kA}(x) = 0) \Rightarrow k_A \in K_A \subset K \right\}$$

3.2. Функция Куна-Таккера (функция Лагранжа) для задач выпуклого программирования (1951г.).

$$L(x,\lambda,\mu) = f(x) + \sum_{j=1}^{l} \lambda_j g_j(x) + \sum_{k=l+1}^{m} \mu_k g_k(x)$$

Продифференцируем функцию L(x,λ,μ) по x и приравняем 0.
Если продифференцировать по $\mu_к$, то получим неравенства $g_k(x) \leq 0$, что не позволяет получить единственное решение. Противоречие состоит в том, что для ЗВП имеется единственное решение.

$$\left.\frac{\partial L(x,\lambda,\mu)}{\partial x_i}\right|_{x^e} = 0 \Rightarrow \nabla_x f(x^e) = -\sum_{j=1}^{l} \lambda_j \nabla g_j(x) - \sum_{k=l+1}^{m} \mu_j \nabla g_j(x)$$



<u>Лемма</u>: Градиент целевой функции $\nabla f(x^e)$ равен линейной комбинации градиентов функций ограничений в точке условного экстремума x^e: $-\nabla f(x^e) = \sum_{j=1}^{l} \lambda_j \nabla g_j(x^e) + \sum_{k=l+1}^{m} \mu_k \nabla g_k(x^e)$

Множество всех векторов

$$\sum_{j=1}^{l} \lambda_j \nabla g_j(x) + \sum_{k=l+1}^{m} \mu_k \nabla g_k(x)$$

называется множеством ***возможных направлений.***

3.3. Теорема Куна-Таккера.

Рассмотрим задачу выпуклого программирования

Пусть f: $X \to Y; X \subset E^n, Y \subset E^1; f(x) \in C^1(X)$

g: $X \to Z; X \subset E^n, Z \subset E^m; g(x) \in C^1(X)$

Требуется найти: $x^e = Arg\min f(x)$, $x \in X$

$x_i \geq 0, i = 1..n$,

$g_j(x) = 0, j = 1..l$,

$g_k(x) \leq 0, k = l+1..m$ · Для этого решение ЗВП должно удовлетворять следующим условиям:

$x^e \in X$,

$\mu_k g_k(x^e) = 0, \lambda_j g_j(x^e) = 0.$

$\nabla_x f(x^e) + \sum_{j=1}^{l} \lambda_j \nabla g_j(x^e) + \sum_{k=l+1}^{m} \mu_k \nabla g_k(x^e) = 0.$

$$\mu_k \begin{cases} = 0, g_k(x) < 0 \\ > 0, g_k(x) = 0 \end{cases}$$

Если эти условия выполнены, то x^e – решение ЗВП.

Условия: $\mu_k g_k(x^e) = 0, \lambda_j g_j(x^e) = 0.$

- называются ***условиями дополняющей нежесткости***.

3.4. Теория двойственности.

3.4.1.Седловая точка функции Лагранжа. Двойственность.

Рассмотрим упрощенную задачу выпуклого программирования.
Пусть заданы условия на f(x) и g (x) как и ранее.

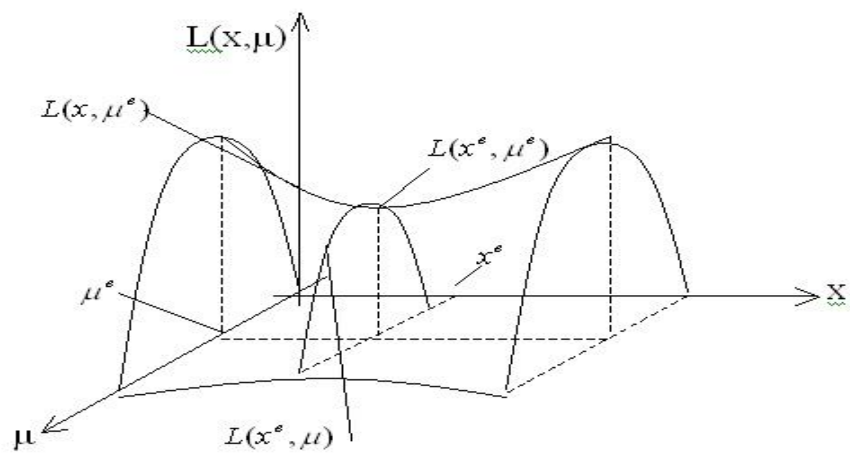
$$\begin{cases} x^e = Arg\min f(x), \\ x_i \ge 0, i = 1..n, \\ g_j(x) \le 0, j = 1..m. \end{cases}$$

Тогда функция Лагранжа может быть записана:

$$L(x,\mu) = f(x) + \sum_{j=1}^{m} \mu_j g_j(x) .$$

Рассмотрим случай, когда x = x1, μ = μ1 :

$$L(x,\mu^e) \ge L(x^e,\mu^e) \ge L(x^e,\mu) \quad .$$



Таким образом седловая точка функции Лагранжа (функции Куна-Таккера) является $\left[x^e, \mu^e\right]^T$.

Именно эта точка является решением ЗВП.

3.4. Теория двойственности.

Задача может быть поставлена как минимаксная и может быть разбита на решение двух задач:

1. **Прямая задача** – $x^e = Arg\min_{x\in X}\max_{\mu\in M} L(x,\mu)$

   при $\mu=\mu(x)$–множитель Лагранжа (Куна-Таккера) .

2. **Двойственная задача** – $\mu^e = Arg\max_{\mu\in M}\min_{x\in X} L(x,\mu)$

   при $x=x(\mu)$ – переменная проектирования.

Пример: Задача линейного программирования.

Прямая задача: $\min c^T x$; $Ax=b$; $x_i \geq 0$.

Двойственная задача:

$$L(x,\mu)=c^T x+(A^T\mu\ x-b^T\mu)=(c^T+A^T\mu)x-b^T\mu\ ;$$

$$\max \mu^T b\ ;\ \ \mu^T A=-c\ ;\ \ \mu_j \leq 0.$$

4. Алгоритм и сходимость алгоритмов.

4.1. Понятие алгоритма. Алгоритмические отображения.

Определение. Алгоритмом называется итеративный процесс вычислений, который порождает последовательность точек в соответствии с предписанным набором правил, включающим критерий окончания этого процесса.

Алгоритмическое отображение.

Пусть А: $\mathrm{X} \to \mathrm{X}; \mathrm{X} \subset \mathrm{E}^n$, $x^{(\kappa+1)} = A(x^{\kappa})$ для $\forall \kappa \in \mathrm{N}$,

Таким образом $x^{(k+1)} = \mathrm{A}\Big(\mathrm{A}\Big(\ldots\Big(\mathrm{A}\big(x^0\big)\Big)\ldots\Big)\Big)$, где $x^{(0)}$ начальная точка.

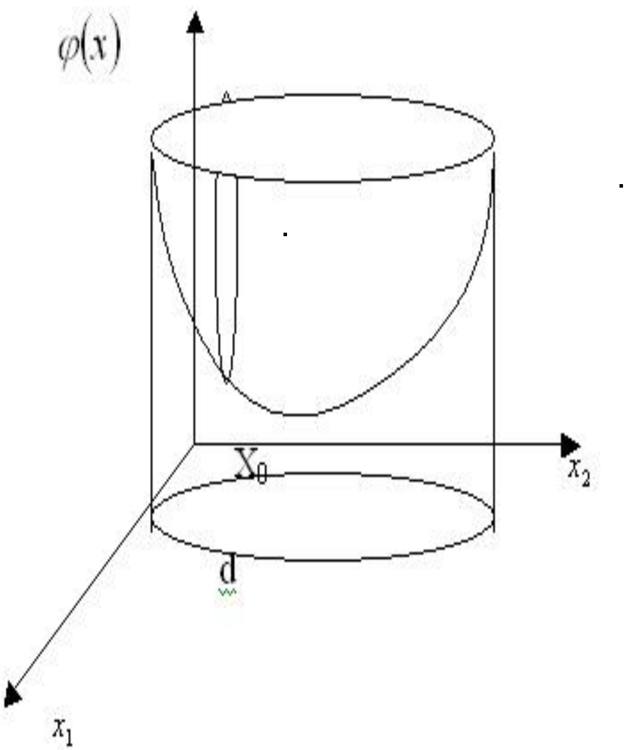
В качестве реализации алгоритмического отображения методов оптимизации может быть выбрана некоторая функция спуска, зависящая от переменных проектирования $\varphi\left(x^{(k)}\right)$.

При этом она должна удовлетворить условию $\varphi\left(x^{(k+1)}\right) \le \varphi\left(x^{(k)}\right)$

В качестве функции спуска может быть использована целевая функция $f(x)$ или градиент целевой функции $\nabla f(x)$.

4. Алгоритм и сходимость алгоритмов.

4.2. Композиция алгоритмических отображений.



1.Выбор направления уменьшения функции спуска d – в области возможного направления. Например $\nabla f(x)*d<0$

Этот алгоритм обозначим D ($d \in E^n$)

2. Выбор шага $\alpha \to \left(\varphi\left(x^{(k)}+\alpha^{(k)}*d^{(k)}\right) \le \varphi\left(x^{(k)}\right)\right)$

$$\alpha^{(k)e} = Arg\min \varphi\left(x^{(k)}+\alpha^{(k)}*d^{(k)}\right)$$

. Этот алгоритм обозначим U.

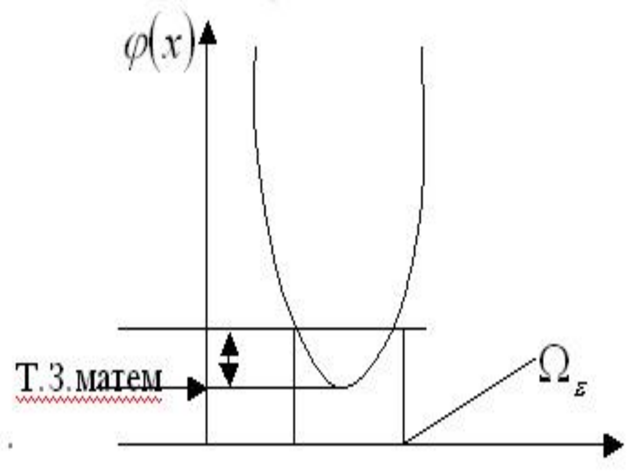
3. Если решается задача с ограничениями, то необходимо ввести еще один алгоритм L , учитывающий наличие ограничений.
Тогда общий алгоритм будет композицией алгоритмов: $A(\varphi(x)) = L*U*D(\varphi(x))$

4. Алгоритм и сходимость алгоритмов.

4.2. Критерий окончания процесса оптимизации.

Точка зрения математика состоит в нахождении точного решения x^{e}.

Точка зрения инженера состоит в определение области $\Omega_{\varepsilon} \subset X$ близкой к оптимальной. В связи с этим существует несколько критериев останова в зависимости от точности решаемой задачи.



Точка зрения инженера.

Критерии останова алгоритма.

1. $\left\| x^{(k+N)} - x^{(k)} \right\| \langle \delta$

2. $\dfrac{\left\| x^{(k+1)} - x^{(k)} \right\|}{\left\| x^{(k)} \right\|} \langle \delta$

3. $\varphi\left(x^{(k)}\right) - \varphi\left(x^{(k+N)}\right) \langle \varepsilon$

4. $\dfrac{\varphi\left(x^{(k)}\right) - \varphi\left(x^{(k+1)}\right)}{\varphi\left(x^{(k)}\right)} \langle \varepsilon$

4. Алгоритм и сходимость алгоритмов.

4.3. Скорость сходимости алгоритмов.

Пусть $\left\{x^{(k)}\right\}$ сходится к $\overline{x}$, тогда, если выполняются неравенства:

(1.) $\lim\limits_{k \to \infty} \sup \dfrac{\left\| x^{(k+1)} - \overline{x} \right\|}{\left\| x^{(k)} - \overline{x} \right\|} = \beta, \quad \beta < 1$, β - коэффициент сходимости,

– линейная сходимость,

(2.) $\lim\limits_{k \to \infty} \sup \dfrac{\left\| x^{(k+1)} - \overline{x} \right\|}{\left\| x^{(k)} - \overline{x} \right\|^{\gamma}} < \beta < +\infty, \gamma > 1, \gamma$ - порядок сходимости,

- суперлинейная сходимость.

При $\gamma = 2$ – квадратичная сходимость.