

Министерство образования и науки Российской Федерации
АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Н.П. Семичевская

**ЗАДАЧИ И МЕТОДЫ
ОПТИМИЗАЦИИ**

Учебно-методическое пособие

Благовещенск

2017

ББК 22.18
С30

*Рекомендовано
учебно-методическим советом университета*

Рецензенты:

Д.А. Теличенко, руководитель группы АСУ ТП, СП «Благовещенская ТЭЦ», филиал «Амурская генерация», АО «Дальневосточная генерирующая компания»;

Е.Л. Еремин, д-р техн. наук, профессор АмГУ.

Семичевская Н.П.

С30 Задачи и методы оптимизации/ Н.П. Семичевская. – Благовещенск: Амурский гос. ун-т, 2017. – 62 с.

Учебно-методическое пособие содержит материалы практических и лабораторных занятий курса “Методы оптимизации”, читаемого на факультете математики и информатики Амурского государственного университета для магистрантов направления подготовки 09.04.01 – «Информатика и вычислительная техника». В пособии рассматриваются разделы: теоретический практикум по решению задач оптимизации, лабораторный практикум по методам оптимизации и раздел обеспечения самостоятельной работы. В каждом разделе приведен необходимый минимум теоретических сведений для решения стандартных задач оптимизации, разобраны примеры задач оптимизации и разработаны примеры алгоритмических структур для решения задач оптимизации на лабораторном практикуме. Целью данного пособия является первоначальное знакомство читателя с задачами и алгоритмами оптимизации, а также формирование у него понимания теоретических основ алгоритмов, предназначенных для решения задач оптимизации.

Учебно-методическое пособие «Задачи и методы оптимизации» предназначено для магистрантов направления подготовки 09.04.01 – «Информатика и вычислительная техника».

ББК 22.18

В авторской редакции

© Амурский государственный университет, 2017
© Семичевская Н.П., 2017

Оглавление

Раздел I	Теоретический практикум по решению задач оптимизации	4
	Практическое занятие №1 Классификация задач оптимизации	5
	Практическое занятие №2 Простейшие задачи теории оптимизации	11
	Практическое занятие №3 Задачи оптимального проектирования	15
	Практическое занятие №4 Задачи оптимального планирования	24
	Практическое занятие №5 Классификация методов оптимизации	28
Раздел II	Лабораторный практикум по методам оптимизации	31
	Лабораторная работа №1 Численные методы решения задач одномерной оптимизации. Метод дихотомии и золотого сечения для нахождения оптимума унимодальной функции	32
	Лабораторная работа №2 Численные методы оптимизации. Метод Ньютона для решения уравнений и задач оптимизации	37
	Лабораторная работа №3 Методы полиномиальной аппроксимации. Квадратичная аппроксимация	41
	Лабораторная работа №4 Методы линейного программирования. Графический метод. Табличный симплекс-метод	42
	Лабораторная работа №5 Методы нелинейного программирования. Минимизация без ограничений	49
	Лабораторная работа №6 Методы нелинейного программирования. Минимизация с ограничениями	50
	Лабораторная работа №7 Оптимальное управление. Задача оптимального управления по быстродействию	53
Раздел III	Обеспечение самостоятельной работы	57
	Задачи для самостоятельного решения по линейному программированию	58
	Задачи оптимального управления	59
	Список литературы	61

РАЗДЕЛ I

**ТЕОРЕТИЧЕСКИЙ ПРАКТИКУМ ПО РЕШЕ-
НИЮ ЗАДАЧ ОПТИМИЗАЦИИ**

Практическое занятие №1 Классификация задач оптимизации

В современных теориях часто встречаются прикладные задачи, которые надо уметь разрешать, используя аппарат теории оптимизации. Поэтому наш практикум по методам оптимизации начинается с классификации задач оптимизации и в дальнейшем с рассмотрения различных примеров постановок конкретных задач оптимизации и их решений. Лабораторный практикум по методам оптимизации содержит алгоритмические реализации разных методов оптимизации при решении классических оптимизационных задач.

В практическом занятии №1 рассматриваются классические постановки оптимизационных задач, взятые из разных учебников по теории и методам оптимизации.

1. ОБЩАЯ ЗАДАЧА МАТЕМАТИЧЕСКОГО ПРОГРАММИРОВАНИЯ

$$f_0(x) \rightarrow \inf, x \in \Omega.$$

1.1 Стандартная задача линейного программирования

$$f_0(x) = (c, x) = \sum_{j=1}^n c_j x_j$$

$$(c, x) \rightarrow \min, x \in \mathbf{R}^n$$

в частном случае, когда заданы линейные ограничения типа равенства $Bx = d$, $x \in \mathbf{R}^n$, $d \in \mathbf{R}^m$, параметры оптимизации принимают положительные значения $x_j \geq 0$, $j=1, n$.

(каноническая задача линейного программирования)

Если к задаче добавить m ограничений типа неравенства $\sum_{j=1}^n a_{ij} x_j \leq b_m$, где $a_{ij} \in \mathbf{R}$, $i=1, m$, то получим общую задачу линейного программирования.

1.2 Задача квадратичного программирования

$$(c, x) + S(Qx, x) \rightarrow \min, x \in \mathbf{R}^n,$$

с линейными ограничениями типа равенства $Bx = d$, $x \in \mathbf{R}^n$, $d \in \mathbf{R}^m$.

1.3 Задача дробно-линейного программирования

$$\frac{(q, x) + \alpha}{(r, x) + \beta} \rightarrow \min, q, r \in \mathbf{R}^n, \alpha, \beta \in \mathbf{R},$$

с ограничениями типа равенства $Bx = d$, $x \in \mathbf{R}^n$, $d \in \mathbf{R}^m$, $x_j \geq 0$, $j=1, n$ и ограничениями типа неравенства $\sum_{j=1}^n a_{ij} x_j \leq b_m$, где $a_{ij} \in \mathbf{R}$, $i=1, m$.

1.4 Задача сепарабельного программирования

Соотношения определяют задачу сепарабельного программирования

$$\begin{cases} f_0(x) = \sum_{j=1}^n h_j(x_j) \rightarrow \min \\ g_i(x) = \sum_{j=1}^n g_{ij}(x_j) \leq \gamma_i, i = 1, m; \\ x_j \geq 0, j = 1, n \end{cases}$$

$\gamma_i \in R$

I.5 Задача геометрического программирования

$$y(x) = \sum_{i=1}^m c_i p_i(x) \rightarrow \inf - \text{позином},$$

где $x = (x_1, \dots, x_n) \in \mathbf{R}_+^n$, $c_i \in \mathbf{R}_+$, $p_i(x) = \prod_{j=1}^n x_j^{a_{ij}}$, если целевая функция и левые

части ограничений типа равенства и / или неравенства в задаче минимизации являются позиномами.

I.6 Задача выпуклого программирования

Задачи оптимизации, в которых необходимо найти минимум выпуклой целевой функции, заданной на выпуклом множестве.

I.7 Задача дискретного программирования

Если множество Ω допустимых решений конечно, то это задача дискретного программирования, а если к тому же координаты этих точек – целые числа, то задачу называют – задачей целочисленного программирования.

II. ЗАДАЧИ НЕЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ

II.1 Задачи нелинейного программирования

II.2. Задача динамического программирования

II.3. Задачи оптимального управления

Классификация задач оптимального управления

Задачи оптимального управления можно разделить на ряд основных типов по следующим классификационным признакам [7]:

- 1) по виду ограничений;
- 2) по виду граничных условий;
- 3) по времени начала и окончания процесса управления;
- 4) по критерию оптимальности.

По виду ограничений могут быть выделены задачи *классического типа*, когда ограничения представляют собой равенства вида:

голомной (алгебраической) связи

$$\Phi(x, u, t) = 0;$$

неголономной (дифференциальной) связи

$$\Phi(dx/dt, x, u, t) = 0;$$

изопериметрической связи

$$\int_t^{t_h} \Phi(x, u, t) dt = \Phi_0 = const.$$

Отметим, что используя новую переменную, можно осуществить формальный переход от изопериметрической связи к неголономной связи и дополнительным граничным условиям, а именно:

$$\frac{dz}{dx} = F(x, u, t), z(t_0) = 0, z(t_h) = \Phi_0.$$

Ограничения *неклассического типа* могут иметь вид следующих неравенств:

$$\Phi(x, u, t) \leq 0;$$

$$\int_t^{t_h} \Phi(x, u, t) dt \leq \Phi_0 = const.$$

По виду граничных условий различают задачи:

с закрепленными (фиксированными) концами оптимальной траектории, когда множества $X(t_{нач})$ и $X(t_{кон})$ состоят из заданных точек фазового пространства;

с подвижными концами траектории, когда множества $X(t_{нач})$ и $X(t_{кон})$ состоят из более чем одной заданной точки фазового пространства, например, в начальный момент времени при любой скорости движения объекта задано только его положение;

со свободным правым концом, когда $X(t_{кон})$ - множество совпадающее со всем фазовым пространством, т.е. нет каких-либо ограничений (заданий) на конечное положение. При этом левый конец может быть как закрепленным, так и подвижным.

По времени начала и окончания процесса управления различают задачи:

с фиксированным временем, когда $t_{нач}$ и $t_{кон}$ заданные моменты времени;

с нефиксированным временем, когда моменты времени $t_{нач}$ и $t_{кон}$ не задаются заранее.

По критерию оптимальности различают следующие задачи:

задачу Больца, когда критерий представляет собой выражение

$$J = G_0[x(t_0), x(t_k), t_0, t_k] + \int_{t_0}^{t_k} F_0(x, u, t) dt;$$

задачу Лагранжа, когда функционал качества включает в себя лишь интегральную составляющую уравнения (2.34), а именно

$$J = \int_{t_0}^{t_k} F_0(x, u, t) dt;$$

задачу Майера, когда функционал качества не содержит интегральной составляющей

$$J = G_0[x(t_0), x(t_k), t_0, t_k].$$

Отметим, что в частном случае задача Майера может быть задачей *терми-*

нального управления (управления конечным положением), в этом случае функционал имеет вид

$$J = G_0[x(t_k), t_k].$$

Если же критерий качества имеет вид

$$J = t_k - t,$$

то задача Майера называется задачей *максимального быстрогодействия*.

Постановка задач оптимального управления

В системах автоматического управления (САУ) проблема оптимизации, как правило, сводится к решению вопросов определения (синтеза) оптимального управляющего воздействия, при этом, если закон управления задан (уравнение регулятора известно), то оптимизация заключается в определении наилучших в некотором смысле значений параметров регулятора [7]. В наиболее общем случае такая САУ, показанная на рис. 1, состоит из:

оптимального программного устройства (ОПУ), вырабатывающего оптимальное программное воздействие $g_{opt}(t)$, определяющее соответствующий оптимальный режим работы системы на основании полного объема входной и выходной информации (известно задающее воздействие $r(t)$ и доступны измерению не только координата $y(t)$, но и весь набор внешних возмущений $f(t)$);

оптимального управляющего устройства (ОУУ), вырабатывающего оптимальное управление $u_{opt}(t)$, в соответствии с $g_{opt}(t)$ и текущим поведением $y(t)$ - выхода объекта управления (ОУ).

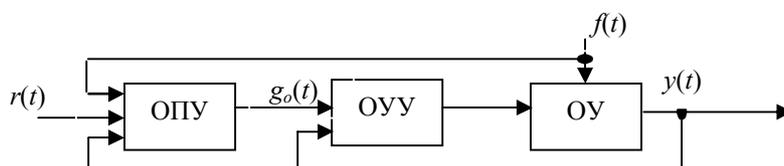


Рис. 1

Данная структура системы оптимального управления может иметь и ряд частных случаев построения, например, при отсутствии обратной связи она становится разомкнутой и получает вид, который представлен на рис.2, 3.

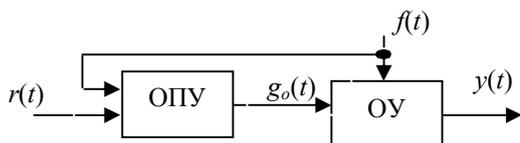


Рис. 2

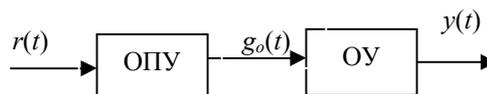


Рис. 3

Такие системы называют *системами с программным оптимальным управ-*

лением, они описываются в работе [7]. Для них характерно отсутствие ОУУ, поскольку обычно выполняется условие $u(t)=g(t)$, причем система управления, при этом оказывается в лучшем случае только *квазиоптимальной*, из-за невозможности измерения при реальной эксплуатации системы полного набора всех возмущений, действующих на объект управления. Этого недостатка лишены системы с обратной связью, структуры которых приведены на рис. 4, 5, где управление формируется с помощью ОУУ.

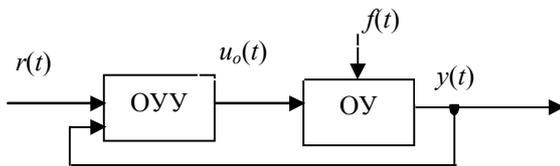


Рис. 4

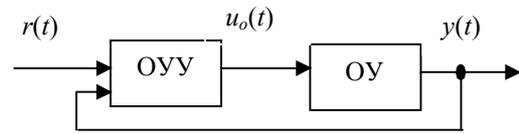


Рис. 5

Можно отметить, что системы, показанные на рис. 2 и 3, не имеют широкого практического применения, но важны с теоретической точки зрения [7].

Задача синтеза оптимальных САУ является типичной задачей на условный экстремум, поскольку оптимальность понимается в смысле минимума некоторого функционала качества, и зачастую формулируется как вариационная задача.

Рассматривая *математическую постановку задачи оптимального управления* требуется иметь:

- 1) уравнение объекта управления;
- 2) ограничения на управление и вектор фазовых координат;
- 3) краевые или граничные условия (значения вектора фазовых координат в начальный и конечный моменты времени);
- 4) критерий оптимальности.

Пусть ОУ описывается уравнением

$$\frac{dx(t)}{dt} = F(x, u, t), \quad (1)$$

где $F(\cdot)$ - некоторая нелинейная вектор-функция от элементов вектора состояния (фазовых координат) $x=x(t)=\text{col}(x_1, x_2, \dots, x_N)$ и составляющих вектора управления $u=u(t)=\text{col}(u_1, u_2, \dots, u_M)$.

На практике значения элементов векторов $u(t)$ и $x(t)$ имеют физические пределы, которые определяются в явном виде, например, с помощью следующих неравенств или равенств:

$$\begin{aligned} u_{j\min} \leq u_j \leq u_{j\max}, \quad u_{j\min}, u_{j\max} = \text{const}, \quad j=1, \dots, M, \\ x_{i\min} \leq x_i \leq x_{i\max}, \quad x_{i\min}, x_{i\max} = \text{const}, \quad i=1, \dots, N. \end{aligned} \quad (2)$$

Поскольку при оптимальном управлении обычно определяется оптимальная траектория, связывающая начальное и конечное положения вектора состояния, то должны быть заданы и граничные условия

$$x_i(t_{\text{нач}}) = x_i(t_n) = x_i(0), \quad x_i(t_{\text{кон}}) = x_i(t_k) = x_i(T_k), \quad i=1, \dots, N, \quad (3)$$

где $t_{нач} = 0$, $t_{кон} = T_k$, соответственно, начальный и конечный моменты времени.

Критерий оптимальности, определяющий числовую характеристику качества управления, как правило, представляет собой некоторый функционал вида

$$J = J(u(t), x(t)). \quad (4)$$

Теперь задачу оптимального управления можно сформулировать следующим образом:

для объекта (1), ограничений (2) и граничных условий (3), требуется найти оптимальное управление $U_{опт} = U_{опт}(x(t), t)$ и фазовую траекторию $X_{опт}(t)$, такую, чтобы функционал (4) принимал бы экстремальное значение.

В большинстве случаев синтеза систем оптимального управления рассматривается как задача минимизации функционала J , но при этом следует иметь в виду, что задачу максимизации функционала J можно всегда свести к задаче минимизации функционала J_0 , используя соотношение $J_0 = -J$. Таким образом, если $J_0\{U_{опт}(t), X_{опт}(t)\} = \min(J_0)$, то это означает, что $U_{опт}(t)$ и $X_{опт}(t)$ также являются оптимальными [7].

Практическое занятие №2 Простейшие задачи теории оптимизации

Задачи, рассматриваемые в практическом занятии №2, относятся к самым первым задачам оптимизации в истории.

Задача 1.

Найдем стороны прямоугольника, вписанного в окружность радиуса R и имеющего наибольшую площадь S (рис. 6).

«Задача известна с глубокой древности, и ее нетрудно решить геометрическим путем. Диагональ вписанного в окружность прямоугольника является диаметром окружности и имеет фиксированную длину. Площадь S прямоугольника равна половине произведения его диагоналей на синус угла φ между диагоналями, т.е. $S = 2R^2 \sin \varphi$. Ясно, что эта площадь будет наибольшей при $\sin \varphi = 1$, или при $\varphi = \pi/2$ » [3].

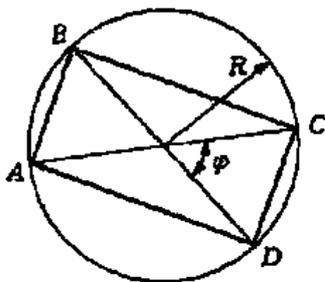


Рис. 6

В этом случае диагонали прямоугольника перпендикулярны, а сам прямо-

угольник представляет собой квадрат. Таким образом, прямоугольником наибольшей площади, вписанным в окружность, является квадрат со стороной $R\sqrt{2}$ и площадью $2R^2$.

Если в качестве *параметров оптимизации* выбрать длины $a > 0$ и $b > 0$ сторон прямоугольника, то получим *целевую функцию* $S(a, b)$ и *ограничение* $a^2 + b^2 = 4R^2$ типа равенства, нелинейные по отношению к этим параметрам. Поэтому рассматриваемую задачу оптимизации следует отнести к задачам *нелинейного программирования*. Ее математическую формулировку можно представить в виде

$$\begin{cases} a \cdot b \rightarrow \max \\ a^2 + b^2 = 4R^2, a \geq 0, b \geq 0. \end{cases}$$

Эту задачу можно решить одним из известных стандартных путей:

либо использовать формальную процедуру поиска условного экстремума функции $S = ab$ двух переменных с одним уравнением связи, построив функцию Лагранжа, либо выразить при помощи ограничения одно переменное через другое и перейти к поиску экстремума функции одного переменного. Отметим, что в данном случае возможен и нестандартный способ решения, связанный с построением вспомогательного соотношения

$$4R^2 - 2S = a^2 + b^2 - 2ab = (a - b)^2 \text{ или } S = 2R^2 - \frac{(a - b)^2}{2}$$

Следовательно, S^* достигает $\frac{1}{2}$ наибольшего значения $S^* = 2R^2$ при условии $a = b$, т.е. когда прямоугольник является квадратом, причем $a = b = R\sqrt{2}$.

Задача 2 (задача Евклида).

Эта задача относится к самым первым оптимизационным задачам древности. В заданный треугольник ABC с высотой H и основанием b вписать параллелограмм наибольшей площади, стороны которого параллельны двум сторонам треугольника (рис. 7).

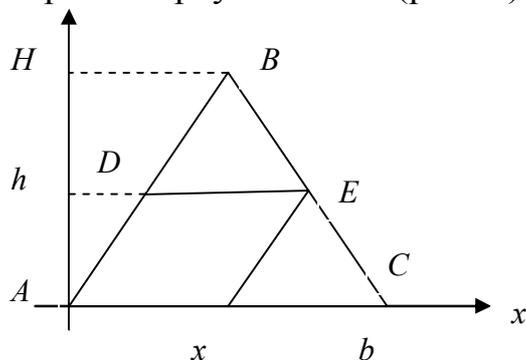


Рис.7

Критерием оптимальности в этой задаче является достижение площадью параллелограмма наибольшего значения, а ограничения свя-

заны с условиями параллельности сторон и размещения параллелограмма в пределах заданного треугольника.

Выберем прямоугольную систему координат, совместив ее начало с общей вершиной параллелограмма и треугольника (вершина A на рис. 1.2), а ось абсцисс — с одной из сторон треугольника. В качестве параметров оптимизации выберем основание x параллелограмма и его высоту h . Тогда площадь S параллелограмма можно записать в виде $S = hx$, а условие, что параллелограмм вписан в треугольник, — как ограничение типа равенства $(H - h)/H = x/b$, которое вытекает из подобия треугольников DBE и ABC .

Итак, имеем задачу оптимизации вида:

$$\begin{cases} S = h \cdot x \rightarrow \max \\ \frac{H - h}{H} = \frac{x}{b}, \quad h \geq 0, x \geq 0. \end{cases}$$

Эту задачу также можно решить стандартными способами (см. задачу 1). Ее решением является параллелограмм с основанием $x^* = b/2$ и высотой $h^* = (1 - x^*/b)H = H/2$. Отметим, что выбор вершины треугольника, с которой совпадает вершина параллелограмма, не является существенным, так как в любом из трех вариантов выбора вершины треугольника площадь параллелограмма максимальной площади равна половине площади треугольника ABC .

Задача 3.

В прямой круговой конус вписан прямой круговой цилиндр так, что основания конуса и цилиндра лежат в одной плоскости (рис. 8). Найдем наибольшую возможную часть объема конуса, занятую таким цилиндром.

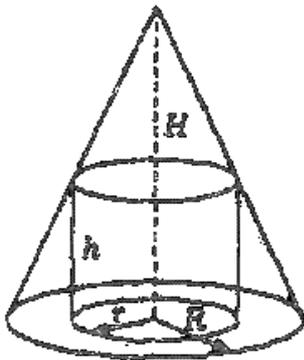


Рис. 8

Как и задача Евклида, эта задача известна тысячи лет. Ее можно решить, используя неравенство средних, известное еще Пифагору, согласно которому среднее геометрическое не превышает среднего арифметического, т.е.

$$\sqrt{ab} \leq \frac{a+b}{2}, \quad \sqrt[3]{abc} \leq \frac{a+b+c}{3} \quad \text{и т.п.,}$$

причем среднее геометрическое равно среднему арифметическому только в том случае, когда все числа равны между собой.

Обозначим высоту и радиус основания цилиндра через h и r , а высоту и радиус основания конуса через H и R . Запишем отношение их объемов:

$$\eta = \frac{\pi r^2 h}{\pi R^2 H / 3} = 3 \frac{r^2 h}{R^2 H}$$

Это равенство задает целевую функцию, в которой в качестве параметров оптимизации можно взять отношения r/R и h/H . Из условия задачи (цилиндр вписан в

конус) вытекает ограничение $r/R = (H - h)/H$. Подставляя его в целевую функцию, получаем

$$\eta\left(\frac{h}{H}\right) = 3\left(1 - \frac{h}{H}\right)^2 \frac{h}{H}, \quad \frac{h}{H} \in (0,1).$$

Ясно, что функция $\eta(h/H)$ достигает максимума при том же значении отношения h/H , что и любая положительная степень правой части этого равенства, в частности степень $1/3$. Используем неравенство для среднего геометрического таких трех величин, чтобы правая часть этого неравенства оказалась константой:

$$\sqrt[3]{3\left(1 - \frac{h}{H}\right)^2 \frac{h}{H}} = \sqrt[3]{12} \sqrt[3]{\left(\frac{1}{2} - \frac{h}{2H}\right)^2 \frac{h}{H}} \leq \frac{\sqrt[3]{12}}{3} \left(\frac{1}{2} - \frac{h}{2H} + \frac{1}{2} - \frac{h}{2H} + \frac{h}{H}\right) = \sqrt[3]{\frac{12}{27}}.$$

Максимум левой части в этом неравенстве будет достигнут при его переходе в равенство, т.е.

$$\frac{1}{2} - \frac{h}{2H} = \frac{h}{H}.$$

Отсюда находим $h/H = 1/3$ и, подставляя в (1), вычисляем $\eta(1/3) = 4/9$. При этом отношение радиусов оснований цилиндра и конуса будет равно $r/R = 2/3$.

Несложно проверить, что к тому же результату можно прийти и стандартным путем поиска экстремума функции $\eta(h/H)$ при $h/H \in [0,1]$.

Задача 4.

Найдем значение параметра a , при котором достигает максимума функция $f(a) = \exp(-1/2(x_1^2 + x_2^2)) = 0$, где x_1, x_2 - корни квадратного уравнения $x^2 + ax + 3a - 1 = 0$.

По теореме Виета¹ имеем $x_1 + x_2 = -a$ и $x_1 x_2 = 3a - 1$.

Отсюда находим $x_1^2 + x_2^2 = (x_1 + x_2)^2 - 2x_1 x_2 = a^2 - 6a + 2 = (a-3)^2 - 7$, и в итоге $f(a) = \exp(3,5 - 0,5(a-3)^2)$. Так как $(a-3)^2 > 0$, а $\exp(-0,5(a-3)^2)$ убывает с возрастанием $(a-3)^2$, то функция $f(a)$ достигает своего максимума при $a = 3$ и этот максимум равен

$$f(a^*) = \exp(3,5) \approx 33,115.$$

Применение теоремы Виета позволило в данном случае упростить задачу и даже избежать использования дифференциального исчисления для поиска экстремума функции $f(a)$, заданной как композиция функций.

Задача 5.

Луч света, переходя из одной однородной среды в другую, падает на границу раздела этих сред под углом α_1 к направлению нормали и после преломления на этой границе составляет с нормалью угол α_2 (рис. 9).

Требуется найти соотношение между этими углами.

¹ Ф. Виет (1540-1603) — французский математик и юрист.

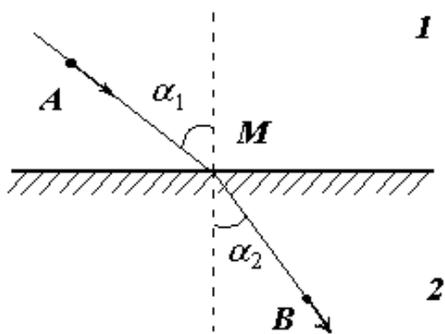


Рис.9

Из курса физики известно, что в соответствии с принципом Ферма² геометрической оптики свет при прохождении через среду с различными свойствами выбирает такую траекторию, время прохождения которой минимально.

Пусть t_1 - время прохождения световым лучом отрезка AM со скоростью c_1 в среде 1, где

x – абсцисса точки M падения светового луча на границу двух сред, t_2 - время прохождения световым лучом отрезка MB со скоростью c_2 в среде 2.

Тогда время прохождения света из точки A в точку B будет равно

$$t(x) = t_1 + t_2 = \frac{\sqrt{b_1^2 + x^2}}{c_1} + \frac{\sqrt{b_2^2 + (a - x)^2}}{c_2}.$$

Из необходимого условия экстремума функции $t(x)$, который по физическому смыслу задачи является минимумом, получаем

$$\frac{c_1}{c_2} = \frac{\frac{x}{\sqrt{b_1^2 + x^2}}}{\frac{a - x}{\sqrt{b_2^2 + (a - x)^2}}} = \frac{\sin \alpha_1}{\sin \alpha_2}.$$

Это равенство выражает закон преломления света, установленный В. Снеллиусом³.

Практическое занятие №3 Задачи оптимального проектирования

При создании технического устройства различного назначения обычно часть его параметров можно изменять в определенных пределах. Это приводит к тому, что при проектировании появляется некоторое множество вариантов создаваемого устройства. В результате возникает проблема выбора из этого множества *альтернатив* наилучшей альтернативы с точки зрения *критерия оптимальности*.

Соответствующие такому выбору *задачи оптимизации* часто называют задачами оптимального проектирования.

² П. Ферма (1601-1665) — французский математик.

³ В. Снеллиус (1580-1626) — нидерландский астроном и математик.

Задача 6.

Одной из наиболее простых задач оптимального проектирования является выбор размеров емкости определенной формы, имеющей наибольший объем при заданной площади поверхности или же наименьшую площадь при заданном объеме.

Требуется спроектировать бак горючего в виде прямого кругового цилиндра заданного объема V , на изготовление которого будет затрачено наименьшее количество листовой стали. В качестве *параметров оптимизации* выберем радиус R и высоту H цилиндра. Тогда затраты материала на изготовление бака будут определять суммарная площадь S его боковой поверхности и двух плоских днищ. Таким образом, необходимо минимизировать *целевую функцию* $S=2\pi R(H+R)$ при *ограничении типа равенства* $\pi R^2 H=V$, т.е. решить задачу *нелинейного программирования*.

Если из целевой функции при помощи ограничения исключить H и записать ее в виде

$$S(R) = \frac{V}{R} + \frac{V}{R} + 2\pi R^2,$$

то произведение слагаемых не будет зависеть от R , и для нахождения ее минимального значения удобно использовать то же неравенство между геометрическим и арифметическим средними, что и в задаче 3:

$$S(R) = \frac{V}{R} + \frac{V}{R} + 2\pi R^2 \geq 3\sqrt[3]{\frac{V}{R} \frac{V}{R} 2\pi R^2} = 3\sqrt[3]{2\pi V^2} = S_*.$$

Равенство имеет место только при равенстве слагаемых, т.е. при $\frac{V}{R} = 2\pi R^2$, тогда

$$R_* = \sqrt[3]{\frac{V}{2\pi}}.$$

Учитывая ограничение, получаем оптимального бака равна его диаметру.

$$H_* = \frac{V}{\pi R_*^2} = \sqrt[3]{\frac{4V}{\pi}} = 2R_*, \quad \text{т.е. высота}$$

При изготовлении одного бака нужно учитывать, что для заготовки круглого днища площадью πR^2 придется взять квадратный лист площадью πR^2 , причем после раскроя оставшуюся часть листа \tilde{S} использовать будет практически невозможно. Поэтому более \tilde{S} обоснованно в качестве целевой минимизировать функцию $= 2\pi RH + 8R^2$ при прежнем ограничении $\pi R^2 H=V$.

Тогда в результате процедуры, аналогичной рассмотренной, получим $\tilde{R}_* = \frac{1}{2}\sqrt[3]{V}$, $\tilde{H}_* = \frac{8}{\pi}\tilde{R}_*$.

Если предстоит изготовить крупную партию баков, то раскрой листовой стали при заготовке днищ можно провести более рационально, располагая соседние центры днищ в вершинах правильных треугольников со стороной $2R$. Тогда расход листа на

каждое днище будет соответствовать площади $2\sqrt{3}R^2$ правильного шестиугольника, описанного около окружности радиуса R . При этом следует минимизировать целевую функцию $\hat{S} = 2\pi RH + 4\sqrt{3}R^2$ при том же ограничении.

В итоге получим
$$\hat{R}_* = \sqrt[3]{\frac{V}{4\sqrt{3}}}, \quad \hat{S}_* = 3\sqrt[3]{V^2 4\sqrt{3}}, \quad \hat{H}_* = \frac{4\sqrt{3}}{\pi} \tilde{R}_*.$$

Отметим, что при постановке задач оптимального проектирования важно, чтобы *математическая модель объекта оптимизации* достаточно полно отражала именно те свойства объекта, улучшение которых является целью оптимизации. Разработка такой модели обычно требует использования сведений из соответствующих инженерных дисциплин или областей техники.

Задача 7.

Из курса сопротивления материалов известно, что допустимая нагрузка, воспринимаемая прямолинейным стержнем, работающим на растяжение или сжатие (без потери устойчивости прямолинейной формы равновесия), пропорциональна площади F его поперечного сечения. При сжатии стержень может потерять устойчивость, изгибаясь в некоторой плоскости. Сжимающая сила, вызывающая потерю устойчивости стержня, пропорциональна *геометрическому моменту инерции I* его сечения относительно *центральной оси тела*, перпендикулярной плоскости изгиба.

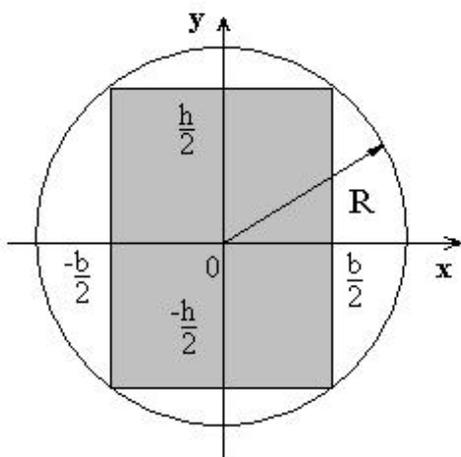


Рис. 10

От значения I также зависит прогиб стержня, изгибаемого в этой плоскости, но зависимость эта обратно пропорциональная. Допустимая нагрузка при изгибе стержня пропорциональна моменту сопротивления $W = I/y_m$ его сечения, где y_m — расстояние до наиболее удаленной от центральной оси точки сечения. Эти сведения важны для выбора критерия оптимальности при проектировании отдельных элементов конструкций, воспринимающих нагрузку.

Предположим, что из круглого бревна радиуса R необходимо выпилить балку с прямоугольным поперечным сечением (рис. 10) так, чтобы ее можно было наиболее эффективно использовать в строительной конструкции. Если балка будет использована в качестве стойки, работающей на сжатие без опасности потери устойчивости, то целесообразно максимизировать площадь F ее сечения, выбрав его в виде квадрата со стороной $R\sqrt{2}$, вписанного в окружность радиуса R .

Максимальное значение площади сечения будет $F_* = 2R^2$ (см. задачу 1).

Таким же должно быть сечение стойки, если возможна потеря ее устойчивости в любой плоскости, проходящей через ее ось. Действительно, нетрудно показать, что в этом случае момент инерции сечения относительно любой оси, лежащей в

плоскости квадрата и проходящей через его центр, постоянен и равен $I=R^4/3$. Если же плоскость изгиба стойки при возможной потере устойчивости predetermined условиями ее крепления, то целесообразно выбрать такое отношение сторон прямоугольника, чтобы момент инерции сечения относительно его центральной оси, перпендикулярной этой плоскости, был максимальным. Обозначим через b ширину сечения вдоль этой оси, а через h его высоту (см. рис. 10).

$$\text{Тогда получим } I = b \int_{-h/2}^{h/2} y^2 dy = \frac{bh^3}{12}.$$

Но стороны прямоугольника должны удовлетворять ограничению $b^2+h^2 \leq 4R^2$. Таким образом, приходим к задаче нелинейного программирования

$$\begin{cases} I = \frac{bh^3}{12} \rightarrow \max; \\ b^2 + h^2 \leq 4R^2, b, h, R \geq 0. \end{cases}$$

Несложно установить, что максимальное значение $I_* = \frac{1}{4}\sqrt{3}R^4$ будет достигнуто при оптимальных высоте $h_* = R\sqrt{3}$ и ширине $b_* = R$ сечения балки.

Такое же поперечное сечение балки следует выбрать, если она нагружена в рассматриваемой плоскости изгибающей нагрузкой. При таком выборе жесткость балки будет максимальной, а ее прогиб — минимальным. Поскольку для рассматриваемого прямоугольного сечения балки $y_m = h/2$, то для момента сопротивления получим $W = I/y_m = bh^2/6$. В данном случае задача нелинейного программирования принимает вид:

$$\begin{cases} W = \frac{bh^2}{6} \rightarrow \max; \\ b^2 + h^2 \leq 4R^2, b, h, R \geq 0. \end{cases}$$

$$\text{Ее решением будет } W_* = \frac{8}{9\sqrt{3}}R^3, h_* = \frac{2\sqrt{2}}{\sqrt{3}}R = \sqrt{2}b_*.$$

Задача 8.

Из курса физики и термодинамики известно, что работа, затрачиваемая в компрессоре на сжатие 1 кг воздуха от начального давления p_0 до давления $p > p_0$, равна $A = \frac{\gamma RT_0}{\gamma - 1} \left(\left(\frac{p}{p_0} \right)^{\frac{\gamma-1}{\gamma}} - 1 \right)$

где γ - коэффициент, характеризующий процесс сжатия (при адиабатическом процессе для воздуха $\gamma = 1,4$); R - газовая постоянная воздуха; T_0 — его температура до сжатия. Чтобы уменьшить затрачиваемую работу в случае высокой степени сжатия $\mu = p/p_0$, проектируют многоступенчатые компрессорные установки, состоящие из нескольких последовательно соединенных компрессоров (ступеней) с промежуточными холодильными устройствами между ступенями, в которых охлаждают воздух, нагревшийся при сжатии.

Пусть требуется спроектировать компрессорную установку из m ступеней в предположении, что воздух, поступающий в i -ю ступень, $i = 1, m$, охлажден до одинаковой температуры T_0 и имеет давление p_{i-1} , равное давлению на выходе из предыдущей ступени. Необходимо при заданной общей степени сжатия μ так выбрать степени сжатия отдельных ступеней $\mu_i = p_i/p_{i-1}$, $i = 1, m$, $p_m = p$, чтобы работа, затрачиваемая на весь процесс сжатия воздуха, была минимальной.

Работа, затрачиваемая на сжатие в i -й ступени, равна

$$A_i = \frac{\gamma RT_0}{\gamma - 1} \left(\left(\frac{p_i}{p_{i-1}} \right)^{\frac{\gamma-1}{\gamma}} - 1 \right) = \frac{\gamma RT_0}{\gamma - 1} (\mu_i^{\frac{\gamma-1}{\gamma}} - 1)$$

а работа, затрачиваемая на весь процесс сжатия, -
$$A = \frac{\gamma RT_0}{\gamma - 1} \sum_{i=1}^m (\mu_i^{\frac{\gamma-1}{\gamma}} - 1)$$

Так как число m ступеней задано, то минимизируемую целевую функцию можно представить в виде

$$- \frac{A}{A_0 m} + 1 = \sum_{i=1}^m \frac{\mu_i^{\frac{\gamma-1}{\gamma}}}{m}, \quad A_0 = \frac{\gamma RT_0}{\gamma - 1}$$

Правая часть этого равенства является средним арифметическим неизвестных чисел $\mu_i^{\frac{\gamma-1}{\gamma}}$, $i = 1, m$. Геометрическое среднее этих чисел равно

$$\sqrt[m]{a_1 a_2 \dots a_m} = \prod_{i=1}^m \mu_i^{\frac{\gamma-1}{m\gamma}} = \prod_{i=1}^m \left(\frac{p_i}{p_{i-1}} \right)^{\frac{\gamma-1}{m\gamma}} = \left(\frac{p}{p_0} \right)^{\frac{\gamma-1}{m\gamma}} = \mu^{\frac{\gamma-1}{m\gamma}}$$

Используя неравенство между геометрическим и арифметическим средними, получаем

$$\frac{A}{A_0 m} + 1 = \sum_{i=1}^m \frac{\mu_i^{\frac{\gamma-1}{\gamma}}}{m} \geq \mu^{\frac{\gamma-1}{m\gamma}}$$

Отсюда следует, что наименьшее значение работы при заданной степени μ сжатия 1 кг воздуха в m -ступенчатой компрессорной установке равно

$$A_* = A_0 m (\mu^{\frac{\gamma-1}{m\gamma}} - 1) \frac{m \gamma RT_0}{\gamma - 1} \left(\left(\frac{p}{p_0} \right)^{\frac{\gamma-1}{m\gamma}} - 1 \right)$$

и может быть достигнуто лишь при равенстве всех слагаемых в правой части целевой функции, т.е. при $a_i = \text{const}$, $i=1, m$. Это возможно только при выборе одинаковой для всех ступеней степени сжатия, равной

$$\tilde{\mu} = \mu^{\frac{1}{m}} = \left(\frac{p}{p_0} \right)^{\frac{1}{m}}$$

Таким образом, затрачиваемая работа минимальна, если значения давлений на выходе из ступеней образуют геометрическую прогрессию со знаменателем $\tilde{\mu}$: $p_1 = \tilde{\mu} \cdot p_0$, $p_2 = \tilde{\mu} \cdot p_1$ и т.д.

Некоторые параметры оптимизации могут принимать лишь дискретные значения (например, целочисленные). В этом случае дифференцирование по таким параметрам при поиске экстремума целевой функции носит условный характер. После формального нахождения точки экстремума в предположении непрерывного изменения этих параметров для них следует принять ближайшие к найденным дискретные значения. Ясно, что это повлияет на экстремальное значение целевой функции.

Задача 9.

Пусть требуется соединить N одинаковых источников постоянного электрического тока в батарею таким образом, чтобы батарея имела наибольшую электрическую мощность, выделяемую на внешней нагрузке с сопротивлением R . Предполагается, что каждый источник имеет электродвижущую силу E и внутреннее сопротивление r .

Если эти источники соединить последовательно, то получим батарею с электродвижущей силой NE и внутренним сопротивлением Nr . Тогда при подключении внешней нагрузки сила тока

$$J_1 = \frac{NE}{Nr + R}$$

, а передаваемая мощность

$$W_1 = J_1^2 R = \frac{(NE)^2 R}{(Nr + R)^2}$$

При параллельном соединении получим батарею с электродвижущей силой E и внутренним сопротивлением r/N .

В этом случае при подключении внешней нагрузки

$$W_2 = \frac{(NE)^2 R}{(r + NR)^2}$$

имеем и

$$J_2 = \frac{E}{r/N + R} = \frac{NE}{r + NR}$$

В частном случае $r = R$ обе батареи дадут одинаковый результат, но при $r < R$ получим $W_2 < W_1$, и наоборот (при $N > 1$).

Можно использовать комбинированную схему соединения источников тока.

Предположим, что N допустимо изменять так, чтобы для некоторого натурального числа n число N/n также было натуральным и больше единицы. Тогда можно параллельно соединить N/n ветвей, каждая из которых будет состоять из n последовательно соединенных источников тока и иметь электродвижущую силу nE и внутреннее сопротивление nr . Внутреннее сопротивление такой батареи равно

, а электродвижущая сила равна n .

$$\frac{nr}{N/n} = \frac{n^2 r}{N}$$

При подключении внешней нагрузки сила тока составит $J = \frac{nE}{n^2 r/N + R}$,

а передаваемая мощность будет равна

$$W = J^2 R = \frac{(nNE)^2 R}{(n^2 r + NR)^2}.$$

Фиксируя N и предполагая, что n изменяется непрерывно, попытаемся найти

максимум передаваемой мощности. Если точка n^* максимума существует, то в ней достигается максимума и сила тока, которую представим функцией J от n

$$J(n) = \frac{nNE}{n^2r + NR}$$

Из необходимого условия $J'(n) = 0$ максимума функции получаем при $n > 0$ ее единственную стационарную точку $n_* = \sqrt{NR/r}$. Проверка достаточного условия показывает, что n_* - точка локального максимума функции $J(n)$. Этой точке соответствуют максимальные значения $J_* = \frac{E}{2\sqrt{\frac{rR}{N}}}$ и $W_* = \frac{NE^2}{4r}$.

Сравним силу тока батарей в зависимости от способа соединения источников тока. Сначала рассмотрим случай $r = R$.

$$\text{При этом } n_* = \sqrt{N}, J_* = \frac{E\sqrt{N}}{2R} \text{ и } J_1 = J_2 = \frac{E}{R(1 + 1/N)} = J(N) < \frac{E}{R}.$$

Наименьшее значение N , при котором имеет смысл комбинированная схема соединения, равно 4. Тогда $n_* = 2$ (две параллельные ветви по два последовательно соединенных источника тока) и $J_* = \frac{E}{R} > J(4)$. Следующее значение N , позволяющее осуществить комбинированное соединение, равно 6 (три ветви по два источника или две ветви по три источника), но формальное вычисление дает для n_* иррациональное число между натуральными числами 2 и 3. Если принять $n = 2$ или $n = 3$, то получим $J = \frac{6E}{5R} > J(6)$. При $N=8$ формальное вычисление n_* также приведет к иррациональному числу, но в этом случае имеют смысл значения $n = 2$ или $n = 4$, для которых найдем $J = \frac{4E}{3R} > J(8)$. Сравнение ограничим выбором значения $N=9$, для которого $n_* = 3$ и $J_* = \frac{3E}{2R}$ т.е. сила тока при комбинированной схеме более чем в 1,5 раза превышает значение $J(9)$.

Так как при $r < R$ выполняется неравенство $J_2 < J_1$, то комбинированную схему соединения источников тока достаточно сравнить с их последовательным соединением. Представим отношение сил токов в виде: $\frac{J_*}{J_1} = \frac{Nr + R}{\sqrt{NrR}} = 2 \frac{Nr + R}{2\sqrt{NrR}}$. Уд-

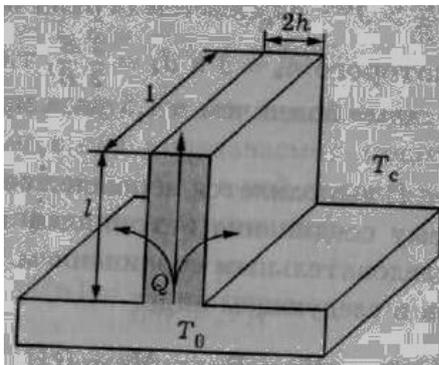
военное отношение среднего арифметического к среднему геометрическому чисел Nr и R не меньше единицы, причем отношение равно 1 только при $Nr = R$. Следовательно, $J_* > 2 J_1$ для всех приемлемых значений N , при которых возможна комбинированная схема соединения. Аналогично показывается, что при $r > R$ для всех приемлемых значений N справедливо $J_* \geq 2J_2 > 2J_1$.

Математическая модель объекта оптимизации может оказаться столь сложной, что решение задачи оптимизации с использованием такой модели

потребуется слишком много времени. В таком случае следует попытаться упростить модель, но так, чтобы в упрощенном виде она отражала свойства объекта, существенные с точки зрения цели оптимизации, выражаемой критерием оптимальности. После выбора значений параметров оптимизации целесообразно провести поверочный расчет с использованием более полной модели и оценить погрешность, связанную с ее упрощением.

Задача 10.

Для интенсификации теплообмена между нагретой поверхностью и охлаждающей средой увеличивают площадь поверхности путем ее оребрения. Рассмотрим плоскую стенку, к которой присоединено достаточно длинное ребро прямоугольного профиля высотой l и толщиной $2h$ (рис. 11).



Помимо передачи теплоты охлаждающей среде с известной температурой T_c непосредственно с поверхности стенки, имеющей температуру $T_0 > T_c$, некоторая часть теплоты распространяется путем теплопроводности от основания ребра к его вершине и переходит в охлаждающую среду через боковые поверхности. Целесообразно так подобрать материал ребра и размеры l и h его профиля,

Рис.11

чтобы тепловой поток Q (количество теплоты в единицу времени), передаваемый охлаждающей среде через участок ребра единичной длины, был максимальным.

Для изготовления ребер желательно использовать материал с достаточно

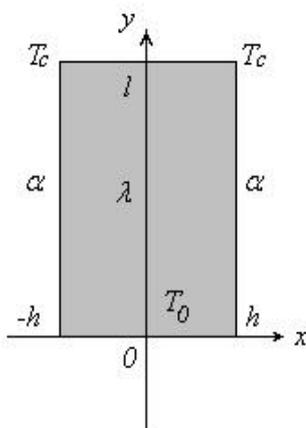


Рис.12

большим коэффициентом теплопроводности λ . Тогда температура боковой поверхности ребра будет в меньшей степени отличаться от температуры его основания, что повысит эффективность использования этой поверхности. Масса ребра, приходящаяся на единицу его длины, пропорциональна площади $F = 2lh$ его сечения.

Следовательно, один из вариантов постановки задачи оптимального проектирования ребра, изготавливаемого из выбранного материала, может быть таким:

критерием оптимальности является максимум передаваемого через ребро теплового потока Q , а ограничением типа неравенства будет $2lh \leq F_0$, где F_0 – заданное значение площади. Параметрами оптимизации в этом случае являются l и h . Поэтому необходимо установить связь этих параметров с ве-

личиной Q , т.е. построить целевую функцию. Это потребует использования математической модели процесса передачи теплоты от стенки через ребро охлаждающей среде.

Если условия теплообмена с охлаждающей средой постоянны вдоль достаточно длинного ребра, то режим теплообмена будет установившимся во времени, а температурное поле в ребре – плоским, определяемым функцией $T(x,y)$, $x \in [0,l]$, $y \in [-h, h]$ (рис.12).

Известно, что эта функция при $\lambda = \text{const}$ удовлетворяет двумерному уравнению Лапласа $\frac{\partial^2 T(x,y)}{\partial x^2} + \frac{\partial^2 T(x,y)}{\partial y^2} = 0$.

Вследствие симметричности сечения ребра относительно оси Ox можно рассматривать лишь половину этого сечения и задавать граничные условия в виде

$$T(0,y) = T_0, \lambda \frac{\partial T(x,y)}{\partial x} \Big|_{x=l} + \alpha T(l,y) = \tilde{\alpha} T_c, y \in [0,l];$$

$$\frac{\partial T(x,y)}{\partial y} \Big|_{y=0} = 0, \lambda \frac{\partial T(x,y)}{\partial x} \Big|_{y=h} + \alpha T(x,h) = \tilde{\alpha} T_c, x \in [0,h],$$

где α_1, α_2 – коэффициенты теплообмена с охлаждающей средой на боковой поверхности ребра и на поверхности его торца соответственно. Решение краевой задачи (8) — (10), записанное в виде двойного ряда по ортогональной системе функций, может быть найдено одним из известных методов, например методом Фурье или с помощью интегральных преобразований. После ее решения для теплового потока, передаваемого через ребро охлаждающей среде и равного тепловому потоку, проходящему через его основание, получим уравнение

$$Q = 2 \int_0^h \left(-\lambda \frac{\partial T(x,y)}{\partial x} \Big|_{x=0} \right) dy = -2\lambda \int_0^h \frac{\partial T(x,y)}{\partial x} \Big|_{x=0} dy.$$

Рассмотренная математическая модель является достаточно сложной и не позволяет представить зависимость Q от параметров оптимизации l и h в удобном виде. Эту модель можно упростить, сохранив точность, достаточную для инженерных приложений. Дело в том, что из соображений качественного характера при ограниченной площади $F = 2lh$ сечения ребра целесообразно увеличить его высоту l , чтобы увеличить площадь боковой поверхности. При этом ребро станет достаточно тонким и по его толщине $2h = F/l \leq F_0/l$, в направлении оси Oy температуру можно принять практически неизменной.

Интегрируя двумерное уравнение Лапласа по y в пределах от 0 до h при граничном условии (10) находим

$$\begin{aligned}
& \int_0^h \frac{\partial^2 T(x, y)}{\partial x^2} + \frac{\partial^2 T(x, y)}{\partial y^2} dy = \\
& = \int_0^h \frac{\partial^2 T(x, y)}{\partial x^2} dy + \frac{\partial T(x, y)}{\partial y} \Big|_{x=h} - \frac{\partial T(x, y)}{\partial y} \Big|_{x=0} = \\
& = \int_0^h \frac{\partial^2 T(x, y)}{\partial x^2} dy + \alpha \frac{T_c - T(x, h)}{\lambda} = 0
\end{aligned}$$

Отсюда, полагая температурное поле в ребре одномерным, определенным функцией $T(x)$, приходим к обыкновенному дифференциальному уравнению вида $\frac{d^2 T(x)}{dx^2} + \alpha \frac{T_c - T(x, h)}{\lambda h} = 0$. При этом граничные условия примут вид $T(0) = T_0$, $\lambda \frac{\partial T(x)}{\partial x} \Big|_{x=l} + \tilde{\alpha} T(l) = \tilde{\alpha} T_c$. Но передачей теплоты с поверхности торца ребра

чаще всего можно пренебречь по сравнению с передачей через боковую поверхность, т.е. положить в уравнении $\tilde{\alpha} = 0$. Это будет соответствовать идеально теплоизолированному торцу и приведет к несколько заниженной оценке для передаваемого через ребро теплового потока, для которого вместо теперь получим $Q = -2\lambda h \frac{\partial T(x)}{\partial x} \Big|_{x=0}$ (14).

Краевая задача (12), (13) при $\tilde{\alpha} = 0$ имеет решение $T(x) = T_c + (T_0 - T_c) \frac{chm(l-x)}{chml}$, $m = \sqrt{\frac{\alpha}{\lambda h}}$. Отсюда, используя (14), находим

$$Q = 2\lambda h(T_0 - T_c)m \frac{shm(l-x)}{chml} \Big|_{x=0} = 2\sqrt{\alpha\lambda h}(T_0 - T_c)thml$$

Таким образом, мы имеем целевую функцию, зависящую от двух параметров оптимизации l и h , и ограничение $2lh \leq F_0$. Подставляя верхнюю оценку $l = F_0/(2h)$ для высоты ребра в целевую функцию, получаем функцию

$Q(h) = 2\sqrt{\alpha\lambda h}(T_0 - T_c)th \frac{F_0}{2} \sqrt{\frac{\alpha}{\lambda h^3}}$ одного переменного h . Необходимое условие $Q'(h) = 0$ максимума этой функции дает трансцендентное уравнение – $sh\zeta = 3\zeta$, $\zeta = \sqrt{\frac{\alpha}{\lambda h^3}}F_0$, которое при $\zeta > 0$ имеет единственное решение $\zeta^* \approx 2,8384$.

Несложно проверить, что соответствующее этому решению значение $h_* = \zeta^* \sqrt[3]{\frac{\alpha F_0^2}{\lambda \zeta^{*2}}} \approx 0,4994 \sqrt[3]{\frac{\alpha F_0^2}{\lambda}}$ удовлетворяет достаточному условию максимума функции $Q(h)$.

Из ограничения получаем верхнюю оценку для оптимальной высоты ребра $l_* = \frac{1}{2} \sqrt[3]{\frac{F_0 \lambda \zeta^{*2}}{\alpha}} \approx 1,0024 \sqrt[3]{\frac{F_0 \lambda}{\alpha}}$. При этом отношение оптимальных значений высоты

ребра к толщине примет вид $\frac{l_*}{2h_*} = \frac{1}{4} \sqrt[3]{\frac{\lambda^2 \zeta_*^4}{\alpha^2 F_0}} \approx 1,0047 \sqrt[3]{\frac{\lambda^2}{\alpha^2 F_0}}$, а максимальное значение теплового потока –

$$Q_* = Q(h_*) = 2 \sqrt[3]{\frac{\alpha^2 F_0}{\zeta_*}} (T_0 - T_C) \frac{\zeta_*}{2} \approx 1,2563 \sqrt[3]{\alpha^2 \lambda F_0} (T_0 - T_C).$$

Практическое занятие №4

Задачи оптимального планирования

Задачи математического программирования часто возникают в экономике, при планировании производственных процессов и количественной оценке *альтернатив*, связанных с принятием управленческих решений. Постановка этих задач обычно основана на анализе и сопоставлении расходуемых ресурсов и полученного результата. Такой подход принято называть методом «затраты – эффективность». Применение этого подхода приводит, как правило, к двум связанным между собой типам задач: либо максимизировать эффективность при ограниченных затратах, либо обеспечивать эффективность не ниже заданной при минимальных затратах. Таким образом, *критерием оптимальности* может быть количественное выражение затрат или эффективности. Рассмотрим несколько примеров.

ОБЩАЯ И ОСНОВНАЯ ЗАДАЧИ ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ

Рассмотренные выше примеры задач линейного программирования позволяют сформулировать общую задачу линейного программирования.

Определение. Общей ЗЛП называется задача, которая состоит в определении оптимального (максимального или минимального) значения линейной функции

$$L = \sum_{j=1}^n c_j x_j, \quad (1)$$

при условиях

$$\begin{cases} \sum_{j=1}^n a_{ij} x_j \leq b_i, & (i = 1, 2, \dots, k), \\ \sum_{j=1}^n a_{ij} x_j = b_i, & (i = k + 1, \dots, m), \\ x_j \geq 0. \end{cases} \quad (2)$$

Функция (1) называется целевой функцией ЗЛП, а условия (2) – ограничениями ЗЛП.

Определение1. Совокупность чисел $\bar{X} = (x_1; x_2; \dots; x_n)$, удовлетворяющих ограничениям ЗЛП, называют допустимым решением (или планом).

Определение2. Оптимальным решением ЗЛП называют допустимое решение $\bar{X}^* = (x_1^*; x_2^*; \dots; x_n^*)$, при котором целевая функция принимает максимальное или мини-

мальное значение.

Определение 3. Основной (или канонической) ЗЛП называется задача, которая состоит в определении оптимального значения целевой функции, при условии, что система ограничений представлена в виде системы уравнений

$$L = \sum_{j=1}^n c_j x_j,$$

при ограничениях

$$\begin{cases} \sum_{j=1}^n a_{ij} x_j = b_i, & (i = 1, 2, \dots, m), \\ x_j \geq 0. \end{cases}$$

Стандартной (или симметричной) ЗЛП называется задача, которая состоит в определении оптимального значения целевой функции, при условии, что система ограничений представлена в виде системы неравенств

$$L = \sum_{j=1}^n c_j x_j,$$

при ограничениях

$$\begin{cases} \sum_{j=1}^n a_{ij} x_j \leq b_i, & (i = 1, 2, \dots, m), \\ x_j \geq 0, & (j = 1, 2, \dots, n). \end{cases}$$

Задача 11.

Предположим, что предприятие может выпускать продукцию n наименований, для производства которой требуется m видов ресурсов (сырья, энергии, оборудования и т.п.). Обозначим через a_{ij} затраты i -го вида ресурсов, $i = 1, m$, на производство единицы продукции j -го наименования, $j = 1, n$, а через b_i и x_j полные объемы располагаемых ресурсов и планируемые объемы выпуска продукции соответственно. Если к тому же по каждому наименованию продукции заданы нижняя a_j и верхняя A_j границы объема выпуска продукции, то можно записать *ограничения типа неравенства*

$$\sum_{j=1}^n a_{ij} x_j \leq b_i, \quad i = \overline{1, m}, \quad a_j \leq x_j \leq A_j, \quad i = \overline{1, n}.$$

Если эффективность производства продукции характеризовать суммарной выручкой от продажи продукции, то оптимальный план $x = (x_1, \dots, x_n)$ выпуска продукции должен удовлетворять этим ограничениям и обеспечивать максимум *целевой функции*

$$S = \sum_{j=1}^n d_j x_j,$$

где d_j - цена единицы продукции j -го наименования.

В данном случае и целевая функция, и ограничения линейны относительно *параметров оптимизации* $x_j, j=1, n$. Поэтому рассмотренная задача оптимального планирования выпуска продукции является *задачей линейного программирования*.

Задача 12 (транспортная задача). Пусть необходимо составить план перевозок некоторого товара с m складов в n магазинов так, чтобы затраты на эти перевозки были минимальными. Предположим, что на i -м складе, $i = 1, m$, имеется a_i единиц товара, а j - магазин, $j=1, n$, сделал заказ на b_j единиц этого товара, причем стоимость его перевозки с i -го склада в j -й магазин равна c_{ij} . Обозначим через x_{ij} планируемое количество товара, перевозимое с i -го склада в j - магазин, тогда стоимость его перевозки составит $c_{ij}x_{ij}$. Общие затраты на перевозки – это сумма затрат на перевозки со всех складов во все магазины. Поэтому оптимальный план перевозок соответствует минимуму целевой функции

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

что должно быть достигнуто выбором mn значений $x_{ij} \geq 0$, которые в данном случае являются параметрами оптимизации. Но при этом необходимо обеспечить потребности магазинов, т.е. должны быть выполнены ограничения типа равенства

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

Однако с любого склада нельзя вывезти товара больше, чем там его находится. Следовательно, должны быть выполнены ограничения типа неравенства

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

Отметим, что сформулированная задача оптимизации, относящаяся к классу задач линейного программирования, имеет решение, если сумма заказов магазинов не превышает суммарного запаса товара на всех складах, т.е.

$$\sum_{j=1}^n b_j \leq \sum_{i=1}^m a_i.$$

Задача 13 (задача о диете). Рассмотрим задачу построения оптимального рациона питания. Обозначим: n — число видов пищевых продуктов; m — число видов питательных веществ; a_{ij} — число единиц i -го питательного вещества в единице j -го продукта; b_i — ежегодная потребность в i -м питательном веществе; c_j — стоимость единицы j -го продукта. Сколько единиц каждого пищевого продукта нужно употребить за рассматриваемый период (в данном случае за год) таким образом, чтобы, обеспечив потребности в каждом питательном веществе, затратив минимальное количество денег.

Назовем рационом вектор $x = (x_1 \ x_2 \ x_3 \ \dots \ x_n)$, где x_j - ежегодное потребление j -го пищевого продукта. Таким образом, необходимо построить рацион минимальной стоимости. Математически эта задача может быть сформулирована следующим образом: минимизировать целевую функцию $\sum_{j=1}^n c_j x_j \rightarrow \min$

при ограничениях
$$\sum_{j=1}^n a_{ij} x_j \geq b_i, \quad i=1, m;$$

$$x_j \geq 0, \quad j=1, n.$$

Задача 14 Предприятие может производить n изделий, причем затраты на производство x_i единиц i -го изделия составляют
$$S(x_i) = a_i x_i^{k_i}$$

где a_i — затраты на производство одного i -го изделия (при мелкосерийном или индивидуальном производстве обычно $k_i \geq 1$, а при крупносерийном — $k_i < 1$). Предположим также, что должно быть выполнено так называемое условие на ассортимент: предприятие должно выпустить не менее b_i единиц i -го изделия, т.е. имеем n ограничений типа неравенства
$$x_i \geq b_i, \quad i=1, n.$$

Если эффективность производства изделий определить как суммарную выручку от их продажи, то получим еще одно ограничение типа неравенства

$$\sum_{j=1}^n d_j x_j \geq b,$$

где $d_i, i=1, n$, — цена единицы i -го изделия, а b — заданный нижний уровень эффективности. При этих ограничениях необходимо минимизировать нелинейную целевую функцию

$$S(x) = \sum_{i=1}^n a_i x_i^{k_i},$$

характеризующую суммарные затраты на производство изделий.

Следовательно, сформулированная задача является *задачей нелинейного программирования*.

Задача 15 Пусть сеть газопроводов связывает между собой m месторождений $A_i, i=1, m$, газа и n пунктов $B_j, j=1, n$, его потребления с известными значениями $p_j \geq 0$ расхода газа в единицу времени. Производительность g_i i -го место рождения, $i=1, m$, ограничена заданным значением G_i , т.е. заданы ограничения типа неравенства $0 \leq g_i \leq G_i$. Затраты на добычу газа на i -м месторождении, $i=1, m$, являются функцией $\varphi_i(g_i)$ производительности g_i . Сеть состоит из K участков, причем стоимость подачи газа по k -му участку, $k=1, K$, является функцией $f_k(q_k)$ расхода q_k через этот участок. В пунктах потребления газа имеем ограничения типа равенства

$$\sum_{k \in B_j^+} q_k = p_j + \sum_{k \in B_j^-} q_k, \quad j=1, n,$$

где B_j^+ и B_j^- — множества участков сети с входящими в j -й пункт и выходящими из него потоками газа соответственно.

Аналогично для месторождений газа получаем ограничения типа равенства

Оптимальным планом $g_i = \sum_{k \in A_i^-} q_k, \quad i=1, m$. добычи газа на место-

рождениях и распределения потоков газа по участкам сети газопроводов будет план, который удовлетворяет указанным ограничениям и обеспечивает минимум общих затрат на добычу и подачу газа составит:

$$s = \sum_{i=1}^m \varphi_i(g_i) + \sum_{k=1}^K f_k(q_k)$$

Все ограничения в сформулированной задаче линейные. Поэтому в частном случае линейных функций $\varphi_i(g_i)$ и $f_k(q_k)$ она будет *задачей линейного программирования*, но в общем случае — *задачей нелинейного программирования*.

Практическое занятие №5 **Классификация численных методов оптимизации**

В практическом занятии №5 обсуждаются различные методы решения задач оптимизации, причем используется классификация методов оптимизации, предложенная в работах [2,3].

Существует классификация методов оптимизации по наличию информации о производных функции, методы оптимизации подразделяются на методы нулевого, первого и второго порядка, соответствующая классификация методов представлена на рис. 13.

Все *методы нулевого порядка* используют в процессе оптимизации только значения функции в области значений и значения аргумента в области определения; они не используют значения производных первого, второго или n -го порядков. Все эти методы отлично изложены в источниках [1-4]. Они требуют от функции удовлетворения определенным условиям, например, унимодальность непрерывной выпуклой функции для метода дихотомии или метода деления отрезка области определения в отношении «золотого сечения».

Методы первого порядка (градиентные) требуют существования первой производной оптимизируемой функции для аналитического вида оптимизируемой функции или численном приближенном виде для конечных разностей. Данная группа методов использует информацию о направлении спуска к минимуму по антиградиенту, не настраивая при этом величину шага. В методе Флетчера-Ривса поиск ведется вдоль взаимно сопряженных направлений, в методе Давидсона-Флетчера-Пауэлла необходима положительная определенность и симметричность матрицы вторых производных, эти методы наиболее быстро сходятся, если оптимизируемая функция квадратичная.

Методы второго порядка (ньютоновские методы) требуют существования первой и второй производной оптимизируемой функции (например, для вычисления матрицы Гессиана) в аналитическом или численном приближенном виде. Данная группа методов использует информацию о направлении спуска к минимуму по антиградиенту и информацию о выпуклости функций, настраивая при этом величину шага.

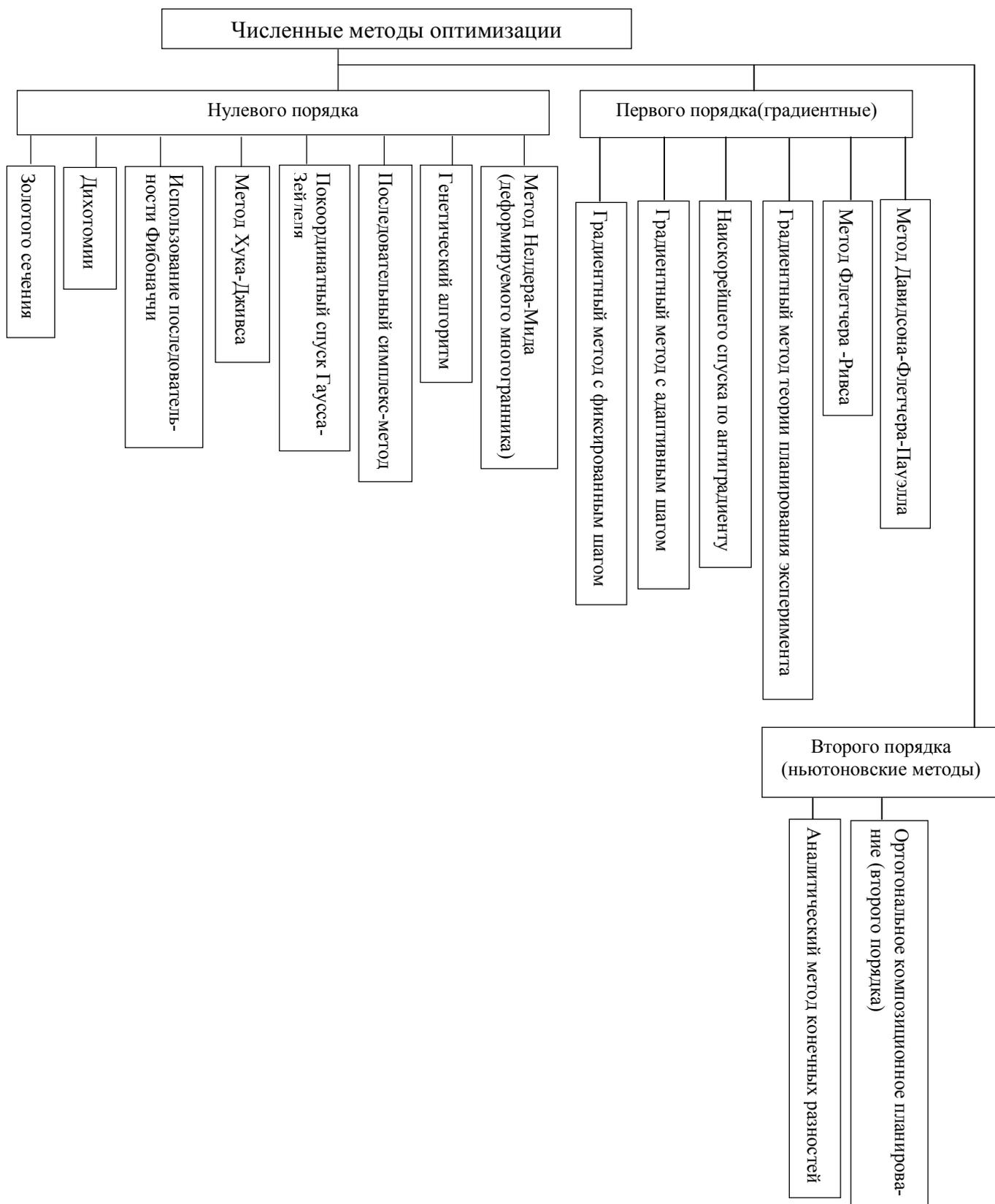


Рис. 13. Классификация численных методов оптимизации по порядку производной

Замечание: Если функция не дифференцируема, но является унимодальной, выпуклой и непрерывной, то возможно применение методов нулевого порядка. Методы оптимизации первого и второго порядков очень медленно сходятся вблизи

зи окрестности оптимума, так как или градиент близок к нулю, или матрица вторых производных плохо обусловлена. Кроме того, данные методы сходятся не к глобальному оптимуму, а к ближайшему локальному экстремуму (т. е. очень зависят от выбора начальной точки), что недопустимо при решении многоэкстремальных задач ввиду существенной погрешности вычислений. Также градиентные методы медленно сходятся на овражных плохо масштабированных функциях.

Существует и другая классификация методов оптимизации, она представлена на рис.14.

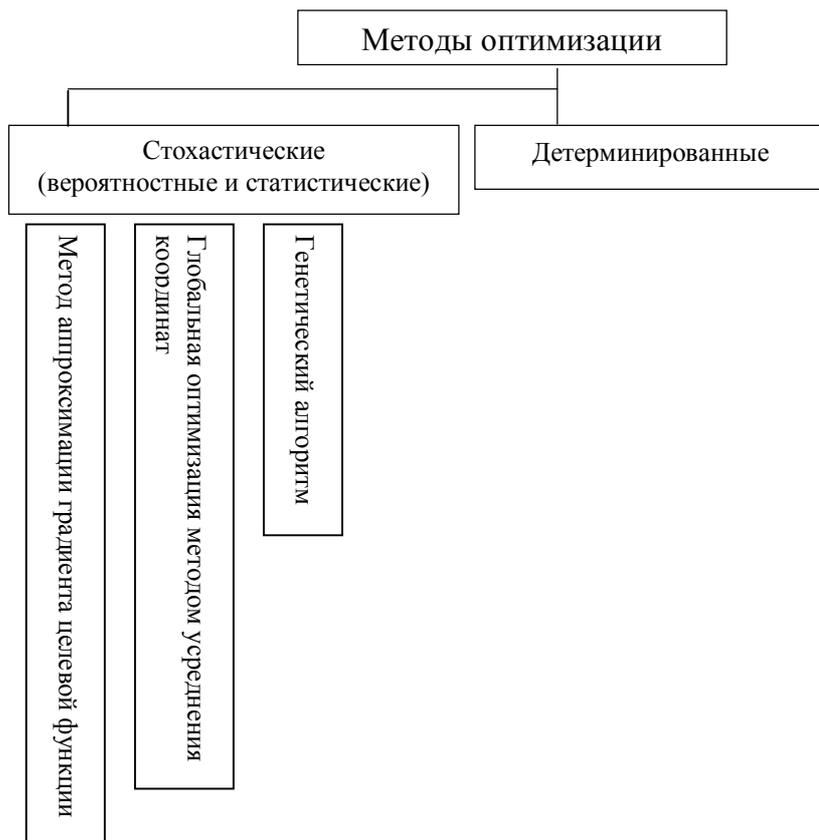


Рис.14. Классификация методов оптимизации

РАЗДЕЛ II

ЛАБОРАТОРНЫЙ ПРАКТИКУМ ПО МЕТОДАМ ОПТИМИЗАЦИИ

Лабораторная работа №1

Численные методы решения задач одномерной оптимизации. Метод дихотомии и золотого сечения для нахождения оптимума унимодальной функции.

Методы решения задачи одномерной оптимизации

Метод дихотомии (метод половинного деления).

Координаты точек вычисляются по формулам: $x_k = \frac{a_k + b_k}{2}$, $y_k = \frac{a_k + x_k}{2}$,
 $z_k = \frac{x_k + b_k}{2}$.

Далее вычисляются значения функции в этих точках. Следующий промежуток неопределенности определяется по правилу:

- 1) если $f(y_k) < f(x_k)$, то $a_{k+1} = a_k$, $b_{k+1} = x_k$;
- 2) если $f(z_k) < f(x_k)$, то $a_{k+1} = x_k$, $b_{k+1} = b_k$;
- 3) если $f(y_k) \geq f(x_k)$ и $f(z_k) \geq f(x_k)$, то $a_{k+1} = y_k$, $b_{k+1} = z_k$;

Работа алгоритма заканчивается, когда длина текущего промежутка неопределенности оказывается не более некоторой величины $\varepsilon > 0$, которую называют требуемой точностью метода.

Создадим файл-функцию в MATLAB `halfdiv.m`, в которой реализован алгоритм вычисления минимума функции методом дихотомии:

```
function halfdiv
clear all
eps=0,01;
disp('---ak---yk---xk---zk---bk---Lk---f(yk)---f(xk)---f(zk)');
a=0;
b=1;
n=-1;
while((b-a)>eps)
x=(a+b)/2;
y=(a+x)/2;
z=(x+b)/2;
dispLine=[a y x z b abs(a-b) fun(y) fun(x) fun(z)];
disp(dispLine);
if fun(y)<fun(x)
a=a;
b=x;
end
if fun(z)<fun(x)
a=x;
b=b;
end
```

```

if (fun(y) >= fun(x) && f(z) >= fun(x))
a=y;
b=z;
end
n=n+1;
end
xmin=x;
fmin=fun(x);
disp('Число итераций')
disp(n)
xmin
fmin
end

```

Замечание. Перед запуском `halfdiv.m` необходимо написать функцию `fun`, реализующую саму функцию, для которой определяется минимум.

Метод золотого сечения

Метод золотого сечения относится к последовательным методам нулевого порядка. В методе золотого сечения две внутренние точки, которые используются для сокращения промежутка неопределенности, выбираются таким образом, чтобы одна из них использовалась с той же целью и на следующем уже сокращенном промежутке. Такое правило выбора точек приводит к тому, что число вычислений функции сокращается вдвое и одна итерация требует расчета только одного нового значения функции. Такими свойствами обладают точки, называемые точками золотого сечения.

Алгоритм метода аналогичен алгоритму метода половинного деления. Первоначально определяется промежуток неопределенности, на котором функция является унимодальной. Вычисляются координаты точек по формулам:

$$y_k = b_k - \frac{(b_k - a_k)}{\varphi}, \quad z_k = a_k + \frac{(b_k - a_k)}{\varphi}, \quad \varphi = \frac{1 + \sqrt{5}}{2} \text{ - золотое сечение.}$$

Следующий промежуток неопределенности определяйся по правилу: если $f(y_k) \geq f(z_k)$ и $f(z_k) \geq f(x_k)$, то $a_{k+1} = y_k$, $b_{k+1} = z_k$.

Работа алгоритма заканчивается, как только длина текущего интервала не станет меньше заданной наперед точности $\varepsilon > 0$.

При этом полагают $x_{\min} = (a_k + b_k)/2$, $f_{\min} = f(x_{\min})$, где k -номер последней итерации.

Следует отметить, что на каждой итерации метода вычисление значения функции только в одной новой точке. При этом к k -итерации длина интервала неопределенности становится равной величине:

$$b_k - a_k = \left(\frac{\sqrt{5} - 1}{2} \right)^k (b_0 - a_0) .$$

Тогда, при заданной точности $\varepsilon > 0$, можно рассчитать количество итераций метода, необходимых для достижения требуемой точности, как наимень-

шее целое число удовлетворяющее неравенству:
$$N \geq \frac{\ln \frac{b_0 - a_0}{\varepsilon}}{\ln \frac{2}{\sqrt{5} - 1}} .$$

Создадим файл-функцию в MATLAB `golddiv.m`, в которой реализован алгоритм вычисления минимума функции методом золотого сечения:

```
function golddiv
clear all
eps=0,01;
disp('---ak---yk---xk---zk---bk---Lk---f(yk)---f(xk)---f(zk)');
a=0;
b=1;
n=0;
fi=(1+sqrt(5))/2;
while((b-a)>eps)
y=b-(b-a)/fi;
z=a+(b-a)/fi;
dispLine=[a y x z b abs(a-b) fun(y) fun(z)];
disp(dispLine);
if fun(y)>=fun(x)
a=y; else
b=z;
end
n=n+1;
end
xmin=(a+b)/2;
fmin=fun(xmin);
xmin=(a+b)/2;
fmin=fun(xmin);
disp('Число итераций:');
disp(n);
disp('-----');
xmin
fmin
end
```

Метод Фибоначчи

В данном методе также реализована стратегия, обеспечивающая макси-

мальное гарантированное сокращение интервала неопределенности при заданном количестве вычислений функции. Стратегия метода опирается на числа Фибоначчи.

Определение. Числа Фибоначчи определяются по формуле: $F_0=F_1=1$, $F_n=F_{n-1}+F_{n-2}$ ($n=2,3,\dots$).

Последовательность Фибоначчи: $F_n = 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, \dots$

Метод относится к последовательным стратегиям. Перед началом работы алгоритма следует вычислить число N вычислений функции. Пусть известен начальный промежуток неопределенности $L_0 = [a_0, b_0]$ и задана точность ε - длина конечного промежутка неопределенности. Тогда N вычисляется как наименьшее целое число, при котором верно неравенство: $F_N \geq \frac{|L_0|}{\varepsilon} = \frac{b_0 - a_0}{\varepsilon}$.

Итерации алгоритма осуществляются по следующему алгоритму. На k -й итерации метода вычисляются координаты точек:

$$y_k = a_k + \frac{F_{N-k-2} + x_k}{F_{N-k}}(b_k - a_k), \quad z_k = a_k + \frac{F_{N-k-1} + x_k}{F_{N-k}}(b_k - a_k).$$

Очевидно, справедливо $z_k = b_k - \frac{F_{N-k-2} + x_k}{F_{N-k}}(b_k - a_k)$, то есть точки y_k и z_k выбираются симметрично внутри отрезка $[a_k, b_k]$, причем $y_k < z_k$.

Следующий промежуток неопределенности определяется по правилу:

- 1) если $f(y_k) < f(z_k)$, то $a_{k+1}=a_k$, $b_{k+1}=z_k$, а $z_{k+1}=y_k$;
- 2) если $f(z_k) < f(y_k)$, то $a_{k+1}=y_k$, $b_{k+1}=b_k$, $y_{k+1}=z_k$;
- 3) если $f(y_k) = f(z_k)$, то $a_{k+1}=y_k$, $b_{k+1}=z_k$;

Работа алгоритма заканчивается, как только $k = N-2$, при этом выполняется условие, что длина текущего интервала не станет меньше заданной наперед точности $\varepsilon > 0$, которую называют требуемой точностью метода. При этом полагают $x_{\min} = (a_k + b_k)/2$, $f_{\min} = f(x_{\min})$, где k -номер последней итерации.

Создадим файл-функцию в MATLAB `fibonacci.m`, вычисляющую числа Фибоначчи:

```
function t=fibonacci(x)
```

```

n(1)=1;
n(2)=1;
for i=3:x
    n(i)=n(i-2)+n(i-1);
end
t=n(x);
end

```

Создадим файл-функцию в MATLAB `fibonaccimethod.m`, в которой реализован алгоритм вычисления минимума функции методом Фибоначчи:

```

function fibonaccimethod
clear all
eps=0,01;
a=0;
b=1;
Fn=max(b-a)/eps;
N=1;
while (fibonacci(N)<Fn)
N=N+1;
end
disp('---ak---yk---zk---bk---|Lk|---f(yk)---f(xk)---f(zk)');
for k=1:N-3
y(k)=a+(b-a)*(fibonacci(N-k-2)/fibonacci(N-k));
z(k)=b-(b-a)*(fibonacci(N-k-2)/fibonacci(N-k));
dispLine=[a y(k) z(k) b abs(a-b) fun(y(k)) fun(z(k))];
disp(dispLine);
if fun(y(k))<fun(z(k))
a=a;
b=z(k);
end
if fun(z(k))<fun(y(k))
a=y(k);
b=b;
end
if fun(y(k))==fun(z(k))
a=y(k);
b=z(k);
end
end
end
xmin=(a+b)/2;
fmin=fun(xmin);
disp('Число итераций')
disp(N)
xmin
fmin
end

```

Представление результатов расчетов тремя методами.

Таблица 1 - Результаты расчетов

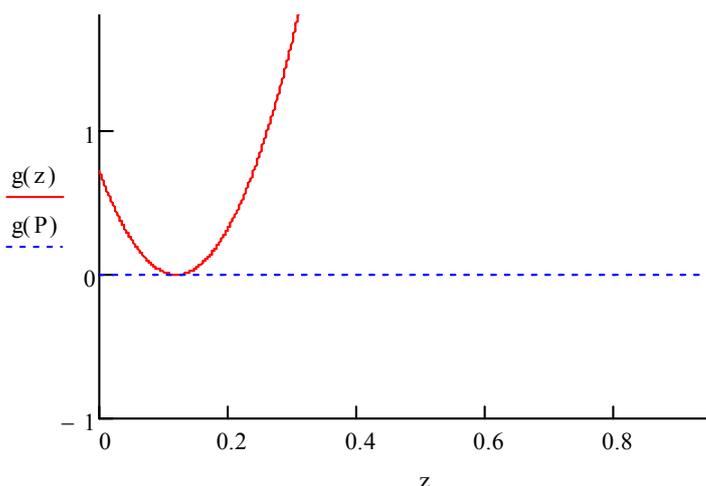
Метод	x^*	$f(x^*)$	Число итераций
Метод половинного деления	0,125	0,0375	3
Метод золотого сечения	0,1008	$3,97 \cdot 10^{-5}$	5
Метод Фибоначчи	0,125	0,0375	7

Следовательно, наиболее эффективным методом является метод половинного деления, в котором достигается нужная точность за меньшее число итераций.

Задача1.

Для функции одной переменной $f(x) = 50(x - 0,12)^2$ определить экстремум графически, классифицировать точки экстремума. Решение задачи оптимизации в MathCad.

Построим график функции и найдем минимум в MathCad



Простое решение в MathCad

```
g(x) := 50(x - 0.12)^2 *  
  
x := 0.01      Given  
x ≤ 1  
  
P := Minimize(g, x)      P = 0.12      g(P) = 0
```

Задание к лабораторной работе

1. Решить задачу оптимизации (минимизация функции), используя разные математические пакеты:

№	Функции	№	Функции
1	$f(x) = 53(x - 0,13)^2$	2	$f(x) = 100(x - 0,25)^2$
3	$f(x) = 110(x - 0,22)^2$	4	$f(x) = 55(x - 0,15)^2$
5	$f(x) = 60(x - 0,10)^2$	6	$f(x) = 10,5(x - 1,25)^2$
7	$f(x) = 2,5x^2 - 50x + 0,13$	8	$f(x) = 55x^2 - 0,3x + 7$
9	$f(x) = -17x^2 - 55x$	10	$f(x) = -10x^2 - 5,8(x + 7)$

Сравнить решение задачи оптимизации в MathCad и MATLAB. Решить задачу оптимизации аналитическим методом, через условие оптимальности первого порядка.

- Сравнить методы последовательного поиска (дихотомии, золотого сечения и Фибоначчи) для функции, выбранной из таблицы вариантов, при заданном значении точности $\varepsilon^* = 0,1$ наибольшей допустимой длины интервала неопределенности при отыскании минимума унимодальных функций заданных на отрезке $[0, 1]$. Задание выполнить в MATLAB.

МЕТОДИКА ВЫПОЛНЕНИЯ РАБОТЫ

- Открыть рабочее окно программы MATLAB 6.
- Создать m-файлы для необходимых функций и алгоритмов.
- Выполнить практическое задание и сравнить алгоритмы по вычислительной сложности.

Лабораторная работа №2

Численные методы оптимизации. Метод Ньютона для решения уравнений и задач оптимизации.

Особенности методов второго порядка

Метод Ньютона является фундаментальным инструментом в численном анализе, исследовании операций, оптимизации и управлении. У него есть множество приложений к инженерным, финансовым и статистическим задачам. Его роль в оптимизации невозможно переоценить: большинство наиболее эффективных методов в линейном и нелинейном программировании строятся на его основе.

Методы безусловной оптимизации второго порядка используют вторые частные производные минимизируемой функции $f(x)$. Суть этих методов состоит в следующем: необходимым условием экстремума функции $f(x)$ в точке x^* является равенство нулю ее первой производной.

Итерационный процесс для построения последовательных приближений к решению задачи минимизации функции $f(x)$ описывается выражением $x_{k+1} = x_k - \frac{f'(x_k)}{f''(x_k)}$.

Полученный метод минимизации называют *методом Ньютона*.

При минимизации овражных функций скорость сходимости метода Ньютона более высока по сравнению с градиентными методами. Существенным недостатком метода Ньютона является зависимость сходимости для невыпуклых функций от начального приближения $x(0)$. Если $x(0)$ находится достаточно далеко от точки минимума, то метод может расходиться, т. е. при проведении итерации каждая следующая точка будет более удаленной от точки минимума, чем предыдущая. Сходимость метода, независимо от начального приближения, обеспечивается выбором не только направления спуска, но и величины шага h_k вдоль этого направления. Соответствующий алгоритм называют *методом Ньютона с регулировкой шага*. Итерационный процесс в таком случае определяется выражением $x_{k+1} = x_k - h_k \frac{f'(x_k)}{f''(x_k)}$. Величина шага h_k выбирается из условия минимума функции $f(x)$ по h в направлении движения, т. е. в результате решения задачи одномерной минимизации.

Создадим файл-функции в MATLAB `dfun.m` и `d2fun.m`, возвращающие значения первой и второй производных функции $f(x)=(x-1)^4$, минимум которой необходимо определить.

```
function y=dfun(x)
    y=4*(x-1).^3;
end
```

```
function y=d2fun(x)
    y=12*(x-1).^2;
end
```

Создадим файл-функцию в MATLAB `newton.m`, реализующей алгоритм вычисления минимума функции $f(x)=(x-1)^4$.

```
function newton
clear all
x0=0,1;
```

```

eps=0,001;
x(1)=x0;
disp(x(1));
k=1;
n=0;
while abs(dfun(x(k))) >=eps
    x(k+1)=x(k)-dfun(x(k))/d2fun(x(k));
    disp(x(k+1));
    k=k+1;
    n=n+1;
end
xmin=x(k);
fmin=fun(xmin);
disp('Число итераций')
disp(n)
xmin
fmin
end

```

Задача 1.

Методом Ньютона минимизируем функцию $h(x)$, т.е. найдем точку x^* наименьшего значения функции $h(x) = 2(x-1)^2 \sin(x)$ на отрезке унимодальности функции $[a, b]$.

Это решение задачи минимизации функции одной переменной, как в лабораторной работе №1

$$h(x) := 2(x-1)^2 \cdot \sin(x)$$

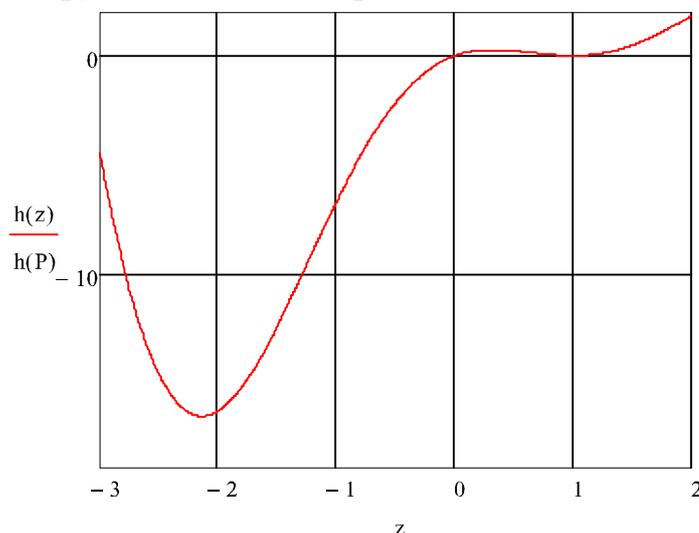
$$x := 0.01$$

Given

$$x \leq 1$$

$$P := \text{Minimize}(h, x)$$

$$P = 138.1661$$



По графику функции отрезок унимодальности примем $[-2.5, -1.5]$

Вычислим производные функции $h(x)$: $h'(x) = 2(x-1)\sin(x) + (x-1)^2 \cos(x)$, $h''(x) = 2\sin(x) + 4(x-1)\cos(x) - (x-1)^2 \sin(x)$

Задание к лабораторной работе

1. Ознакомиться с примерами использования программного пакета Optimization Toolbox в оптимизационных задачах.
2. Минимизировать функции методом Ньютона.

№	Функции	№	Функции
1	$f(x) = (x - 2)^2(x - 1,5)$	6	$f(x) = (x - 2)^2 \sin x$
2	$f(x) = (x - 1)^2(x - 2)^2$	7	$f(x) = (x - 1,5)^2 \sin x$
3	$f(x) = (x - 1)^4$	8	$f(x) = (x - 1)^2 \cos 2x$
4	$f(x) = (x - 2,2)(x - 1,6)^3$	9	$f(x) = (x - 1)^4 \cos x$
5	$f(x) = (x - 1,9)^2 \sin 2x$	10	$f(x) = (x - 1)^2(x - 1,5)$

Самостоятельно определить отрезок унимодальности функции. Также определить нули функции.

- Какой из методов – золотого сечения, Ньютона или кубической интерполяции окажется наиболее эффективным, если производные вычисляются приближенно через разность значений функции в близких точках?

МЕТОДИКА ВЫПОЛНЕНИЯ РАБОТЫ

- Открыть рабочее окно программы MATLAB 6.
- Изучая теоретические сведения о пакете Optimization Toolbox, вводить в рабочем окне команды, приводимые в описании пакета. Найти функцию, реализующую поиск оптимального решения методом Ньютона.
- Закончив изучение теоретических сведений о пакете Optimization Toolbox, выполнить практическое задание.

Лабораторная работа №3

Методы полиномиальной аппроксимации. Квадратичная аппроксимация

Пусть в ограниченном интервале функцию $f(x)$ можно аппроксимировать квадратичным полиномом. Пусть задана последовательность точек x_1, x_2, x_3 и известны значения функции в этих точках: $f(x_1) = f_1, f(x_2) = f_2, f(x_3) = f_3$. Тогда можно определить постоянные коэффициенты a_0, a_1 и a_2 таким образом, что значения квадратичного полинома $\varphi(x) = a_0 + a_1(x - x_1) + a_2(x - x_1)(x - x_2)$ совпадут со значениями функции $f(x)$ в трех указанных точках. Для этого вычислим значение $j(x)$ в трех заданных точках:

$$a_0 = f(x_1), \quad a_1 = \frac{(f_2 - f_1)}{(x_2 - x_1)}, \quad a_2 = \frac{1}{x_3 - x_2} \left(\frac{f_3 - f_1}{x_3 - x_1} - \frac{f_2 - f_1}{x_2 - x_1} \right).$$

Если точность аппроксимации функции $f(x)$ на отрезке $[x_1, x_2]$ квадратичным полиномом оказывается достаточно высокой, то построенный полином далее используется для оценивания координаты точки минимума функции $f(x)$.

Стационарные точки полинома определяются посредством приравнивания к нулю его первой производной и последующим нахождением корней полученного таким образом уравнения. Для оценки координаты точки минимума функции $f(x)$

имеем формулу $x^* = \frac{1}{2}(x_2 + x_1) - \frac{a_1}{2a_2}$.

Приведем описание алгоритма метода последовательного оценивания с использованием квадратичной аппроксимации, разработанного Пауэллом.

1. Пусть $f(x)$ – оптимизируемая функция, x_1 – начальная точка, Dx – величина шага по оси абсцисс. Вычислить $x_2 = x_1 + Dx$ и значения функции $f(x_1)$ и $f(x_2)$.
2. Если $f(x_1) > f(x_2)$, то положить $x_3 = x_1 + 2 Dx$. Если $f(x_1) < f(x_2)$, то положить $x_3 = x_1 - Dx$.
3. Вычислить значение функции $f(x_3)$. Найти $f_{\min} = \min\{f(x_1), f(x_2), f(x_3)\}$, x_{\min} – точка, которой соответствует f_{\min} .
4. По трем точкам x_1, x_2, x_3 вычислить x^* по формуле $x^* = \frac{1}{2}(x_2 + x_1) - \frac{a_1}{2a_2}$.
5. Произвести проверку на окончание поиска минимума. Если разности $f_{\min} - f(x^*)$ и $x_{\min} - x^*$ являются достаточно малыми величинами, то закончить поиск; иначе перейти к п. 6.
6. Выбрать «наилучшую» точку (x_{\min} или x^*) и две точки по обе стороны от нее. Обозначить эти точки в естественном порядке и перейти к п. 3.

Задание к лабораторной работе

1. Ознакомиться с примерами использования программного пакета Optimization Toolbox в оптимизационных задачах.
2. Запрограммировать формулу квадратичной оптимизации для функции $f(x)$ в окрестности точки локального минимума.
3. Построить график функции $f(x)$
4. Построить таблицу вычислений метода квадратичной аппроксимации для нахождения минимума функций:

№	Функции	№	Функции
1	$f(x) = (x - 2)^2(x - 1,5)$	6	$f(x) = (x - 2)^2 \sin x$
2	$f(x) = (x - 1)^2(x - 2)^2$	7	$f(x) = (x - 1,5)^2 \sin x$
3	$f(x) = (x - 1)^4$	8	$f(x) = (x - 1)^2 \cos 2x$
4	$f(x) = (x - 2,2)(x - 1,6)^3$		
5	$f(x) = (x - 1,9)^2 \sin 2x$		

МЕТОДИКА ВЫПОЛНЕНИЯ РАБОТЫ

1. Открыть рабочее окно программы MATLAB 6.
2. Создать m-файлы для необходимых функций и алгоритмов.
3. Изучая теоретические сведения о пакете Optimization Toolbox, вводить в рабочем окне команды, приводимые в описании пакета.
4. Найти функцию, реализующую поиск оптимального решения методом Ньютона, если найти функцию, то составить функцию самостоятельно.
5. Закончив изучение теоретических сведений о пакете Optimization Toolbox, выполнить практическое задание.

Лабораторная работа №4

Методы линейного программирования. Графический метод. Табличный симплекс-метод

Пример.

Решение задачи линейного программирования с использованием электронных таблиц Microsoft Excel

Задача оптимального планирования производства ЗАО «БМК» производит два вида продукции: молоко и сметану. Молоко продается по цене 20 руб., сметана по цене 30 руб. Для производства обоих продуктов используется два вида сырья: цельное молоко и сахар. Затраты сырья на производство продукции приведены в таблице, запасы сырья молока цельного составляют 300 тонн, а сахара 430 тонн.

Сырье	Продукт		Запасы сырья, тонн
	Молоко	Сметана	
Молоко (цельное)	0,3	0,6	300
Сахар	0,45	0,4	430
Цена, руб.	20	30	Max

Необходимо определить оптимальный объем производства, при котором доход будет максимальным.

Решение:

Составим математическую модель по исходным данным задачи оптимального планирования производства. Пусть x_1 – количество произведенного молока, а x_2 – количество произведенной сметаны.

Итак, математическую модель задачи можно записать следующим образом:

$W = 20 x_1 + 30 x_2 \rightarrow \max$ (целевая функция, выражающая доход предприятия) при ограничениях:

$$\begin{aligned} 0,3 x_1 + 0,6 x_2 &\leq 300, \\ 0,45 x_1 + 0,4 x_2 &\leq 430, \\ x_1 &\geq 0, x_2 \geq 0. \end{aligned}$$

	A	B	C	D	E	F	G
1	Переменные						
2	Имя	Молоко	Сметана				
3	Значение						
4	Кэф. ЦФ	20	30	max	0		
5							
6	Ограничения				знак		
7	Молоко цельное	0,3	0,6	0	<=	300	
8	Сахар	0,45	0,4	0	<=	430	
9							

Рисунок – Исходные данные задачи

	A	B	C	D	E	F	G
1	Переменные						
2	Имя	Молоко	Сметана				
3	Значение						
4	Кэф. ЦФ	20	30	max	=СУММПРОИЗВ(B4:C4;B3:C3)		
5							
6	Ограничения				знак		
7	Молоко цельное	0,3	0,6	=СУММПРОИЗВ(B3:C3;B7:C7)	<=	300	
8	Сахар	0,45	0,4	=СУММПРОИЗВ(B3:C3;B8:C8)	<=	430	
9							

Рисунок – Форма для ввода данных в режиме формул

Рисунок – Форма для ввода ограничений

	A	B	C	D	E	F
1	Переменные					
2	Имя	Молоко	Сметана			
3	Значение	920	40			
4	Коэф. ЦФ	20	30	max	19600	
5	Ограничения				знак	
6	Молоко цельное	0,3	0,6	300	<=	300
7	Сахар	0,45	0,4	430	<=	430

Рисунок – Решение задачи

	A	B	C	D	E	F	G
1	Microsoft Excel 10.0 Отчет по результатам						
2	Рабочий лист: [Поиск Решения.xls]Лист1						
3	Отчет создан: 22.09.2013 8:56:08						
4							
5							
6	Целевая ячейка (Максимум)						
7	Ячейка	Имя	Исходное значение	Результат			
8	\$E\$4	max	0	19600			
9							
10							
11	Изменяемые ячейки						
12	Ячейка	Имя	Исходное значение	Результат			
13	\$B\$3	Значение Молоко	0	920			
14	\$C\$3	Значение Сметана	0	40			
15							
16							
17	Ограничения						
18	Ячейка	Имя	Значение	Формула	Статус	Разница	
19	\$D\$7	Молоко цельное max	300	\$D\$7 <= \$F\$7	связанное	0	
20	\$D\$8	Сахар max	430	\$D\$8 <= \$F\$8	связанное	0	
21	\$B\$3	Значение Молоко	920	\$B\$3 >= 0	не связан.	920	
22	\$C\$3	Значение Сметана	40	\$C\$3 >= 0	не связан.	40	
23							

Рисунок – Отчет по результатам

Из рисунка 5 видно, что расчетное максимальное значение целевой функции равно 19 600 руб. Искомые объемы выпуска продукции равны соответственно: молока – 920 шт.(упаковок), сметаны – 40 шт.(упаковок). Исходное сырье использовано полностью.

	A	Имя	C	D	E	F	G	H
1	Microsoft Excel 10.0 Отчет по устойчивости							
2	Рабочий лист: [Поиск Решения.xls]Лист1							
3	Отчет создан: 22.09.2013 8:56:08							
4								
5								
6	Изменяемые ячейки							
7			Результ.	Нормир.	Целевой	Допустимое	Допустимое	
8	Ячейка	Имя	значение	стоимость	Козффициент	Увеличение	Уменьшение	
9	\$B\$3	Значение Молоко	920	0	20	13,75	5	
10	\$C\$3	Значение Сметана	40	0	30	10	12,22222222	
11								
12	Ограничения							
13			Результ.	Теневая	Ограничение	Допустимое	Допустимое	
14	Ячейка	Имя	значение	Цена	Правая часть	Увеличение	Уменьшение	
15	\$D\$7	Молоко цельное max	300	36,66666667	300	345	13,33333333	
16	\$D\$8	Сахар max	430	20	430	20	230	
17								

Рисунок – Отчет по устойчивости

Нормированный градиент – это двойственные оценки, которые показывают, насколько изменится целевая функция при принудительном включении единицы этой продукции в оптимальное решение. При увеличении объема производства молока на единицу, целевая функция увеличится на 13,75 руб.

В нижней таблице представлен множитель Лагранжа, т.е. теневые оценки (двойственные оценки), которые показывают, как изменится целевая функция при изменении соответствующего ресурса на единицу. Для ограничения по цельному молоку 300, множитель Лагранжа равен 36,67 руб.,

	A	B	C	D	E	F	G	H	I	J
1	Microsoft Excel 10.0 Отчет по пределам									
2	Рабочий лист: [Поиск Решения.xls]Отчет по пределам 1									
3	Отчет создан: 22.09.2013 8:56:08									
4										
5										
6	Целевое									
7	Ячейка	Имя	Значение							
8	\$E\$4	max	19600							
9										
10										
11	Изменяемое			Нижний Целевой		Верхний Целевой				
12	Ячейка	Имя	Значение	предел	результат	предел	результат			
13	\$B\$3	Значение Молоко	920	0	1200	920	19600			
14	\$C\$3	Значение Сметана	40	0	18400	40	19600			
15										

Рисунок – Отчет по пределам

Данный вид отчета показывает, в каких пределах может меняться выпуск продукции, вошедшей в оптимальное решение при сохранении структуры опти-

мального решения, т.е. от нижнего предела, до расчетного значения.

Задание к лабораторной работе

Найти минимум/максимум в задаче линейного программирования графическим и симплекс методом, используя <Поиск решений> в MicrosoftExcel (аналогично примеру).

Задание 1.

Решить задачи линейного программирования графическим методом.

1	$L = 3x_1 + x_2 \rightarrow \min;$ $\begin{cases} x_1 \leq 2, \\ x_2 \geq 2, \\ 2x_1 + 5x_2 \geq 12, \\ x_1 \geq 0, x_2 \geq 0. \end{cases}$	2	$L = 3x_1 + 0,5x_2 \rightarrow \max;$ $\begin{cases} 4x_1 + x_2 \geq 5, \\ -3x_1 + 10x_2 \leq 50, \\ x_1 \geq 0, x_2 \geq 0. \end{cases}$
3	$L = 2x_1 - 5x_2 \rightarrow \min;$ $\begin{cases} 3x_1 + 5x_2 \geq 8, \\ -3x_1 + 10x_2 \leq 16, \\ x_1 \geq 0, x_2 \geq 0. \end{cases}$	4	$L = x_1 - 4x_2 \rightarrow \min;$ $\begin{cases} 3x_1 + 4x_2 \leq 6, \\ 2x_1 + 3x_2 \leq 4, \\ x_1 \geq 0, x_2 \geq 0. \end{cases}$
5	$L = x_1 + x_2 \rightarrow \max;$ $\begin{cases} x_1 + x_2 \geq 1, \\ -x_1 + x_2 \leq 1, \\ x_1 \geq 0, x_2 \geq 0. \end{cases}$	6	$L = 2x_1 + 2x_2 \rightarrow \min;$ $\begin{cases} x_1 + 3x_2 \leq 30, \\ 2x_1 + x_2 \leq 20, \\ x_1 \geq 0, x_2 \geq 0. \end{cases}$
7	$L = 2x_1 + 3x_2 \rightarrow \max;$ $\begin{cases} x_1 + x_2 \leq 3, \\ -x_1 + 2x_2 \leq 5, \\ x_1 \geq 0, x_2 \geq 0. \end{cases}$	8	$L = x_1 - x_2 \rightarrow \min;$ $\begin{cases} x_1 \geq 4, \\ x_2 \geq 3, \\ x_1 + x_2 \leq 8, \\ x_1 \geq 0, x_2 \geq 0. \end{cases}$
9	$L = x_1 + 4x_2 \rightarrow \max;$ $\begin{cases} 2x_1 + 4x_2 \leq 9, \\ 3x_1 \leq 1, \\ x_2 \leq 1, \\ x_1 \geq 0, x_2 \geq 0. \end{cases}$	10	$L = x_1 + 3x_2 \rightarrow \max;$ $\begin{cases} x_1 \geq 4, \\ x_2 \geq 5, \\ 3x_1 + x_2 \leq 15, \\ x_1 \geq 0, x_2 \geq 0. \end{cases}$

Задание 2.

Решить задачи линейного программирования табличным симплекс-методом и провести анализ на чувствительность

1	$L = x_1 - 2x_2 + 2x_3 + 3x_4 \rightarrow \max;$ $\begin{cases} x_1 + x_2 + 2x_3 + 2x_4 = 8, \\ 2x_1 + 2x_2 + x_3 + x_4 = 10, \\ x_1 - 2x_2 + x_3 + 2x_4 = 1, \\ x_j \geq 0, j = 1, \dots, 4. \end{cases}$	2	$L = 3x_1 + 2x_2 + 3x_3 + x_4 \rightarrow \min;$ $\begin{cases} 2x_1 + 3x_2 + 3x_4 = 10, \\ x_2 + 2x_3 + x_4 = 4, \\ x_1 + 2x_2 + 2x_3 + 2x_4 = 8, \\ x_j \geq 0, j = 1, \dots, 4. \end{cases}$
3	$L = x_1 + 2x_2 + 3x_3 + x_4 \rightarrow \max;$	4	$L = 3x_1 + 2x_2 + x_3 + x_4 \rightarrow \min;$

	$\begin{cases} 2x_1 + x_2 + 3x_3 + x_4 = 12, \\ x_1 + 2x_2 + x_3 + 2x_4 = 8, \\ 3x_1 + 3x_2 + x_3 + 3x_4 = 15, \\ x_j \geq 0, j = 1, \dots, 4. \end{cases}$		$\begin{cases} 2x_1 + 2x_2 + 3x_3 = 9, \\ x_2 + 2x_3 + x_4 = 4, \\ x_1 + 2x_2 + 2x_3 + 2x_4 = 8, \\ x_j \geq 0, j = 1, \dots, 4. \end{cases}$
5	$\begin{aligned} L = 2x_1 + x_2 + x_3 + 2x_4 \rightarrow \max; \\ \begin{cases} x_1 + 2x_2 + x_3 + 2x_4 = 16, \\ 2x_1 + x_2 + 2x_3 + x_4 = 14, \\ 2x_1 + 2x_2 - 2x_3 + x_4 = 4, \\ x_j \geq 0, j = 1, \dots, 4. \end{cases} \end{aligned}$	6	$\begin{aligned} L = 2x_1 + x_2 + 2x_3 + 2x_4 \rightarrow \min; \\ \begin{cases} 2x_1 + x_2 + 2x_3 + x_4 = 8, \\ x_1 + 2x_2 + x_3 + 2x_4 = 10, \\ 2x_1 + x_2 + 2x_3 + 2x_4 = 10, \\ x_j \geq 0, j = 1, \dots, 4. \end{cases} \end{aligned}$
7	$\begin{aligned} L = 2x_1 - x_2 + 3x_3 - 2x_4 \rightarrow \max; \\ \begin{cases} x_1 + x_2 + 2x_3 - x_4 = 3, \\ 2x_1 + x_2 - x_3 + 2x_4 = 4, \\ x_1 + 2x_2 + x_3 + x_4 = 5, \\ x_j \geq 0, j = 1, \dots, 4. \end{cases} \end{aligned}$	8	$\begin{aligned} L = x_1 + 2x_2 + x_3 + x_4 \rightarrow \min; \\ \begin{cases} x_1 + x_2 - x_3 + x_4 = 4, \\ 2x_1 + x_2 + 2x_3 - x_4 = 4, \\ x_1 - x_2 + x_3 + x_4 = 2, \\ x_j \geq 0, j = 1, \dots, 4. \end{cases} \end{aligned}$
9	$\begin{aligned} L = 3x_1 + x_2 + x_4 \rightarrow \max; \\ \begin{cases} x_1 + 2x_2 + 2x_4 = 6, \\ 2x_2 + 2x_3 + x_4 = 7, \\ x_1 + x_2 + x_3 + 2x_4 = 7, \\ x_j \geq 0, j = 1, \dots, 4. \end{cases} \end{aligned}$	10	$\begin{aligned} L = 4x_1 + 2x_2 + 2x_3 + x_4 \rightarrow \min; \\ \begin{cases} x_1 + x_2 + x_3 + 2x_4 = 8, \\ 2x_1 + x_2 + x_3 + 2x_4 = 10, \\ x_1 + x_2 + 2x_3 - 2x_4 = 6, \\ x_j \geq 0, j = 1, \dots, 4. \end{cases} \end{aligned}$

Лабораторная работа №5

Методы нелинейного программирования. Минимизация без ограничений

Примеры решения оптимизационных задач

Минимизация без ограничений

Рассмотрим задачу нахождения значений переменных x_1 и x_2 , обеспечивающих решение задачи минимизации

$$\min_{\mathbf{x}} f(\mathbf{x}) = e^{x_1} (4x_1^2 + 2x_2^2 + 4x_1x_2 + 1)$$

Нахождение решения производится в соответствии со следующими этапами.

Этап 1. Составление m-файла (с именем objfun), реализующего вычисление значения целевой функции:

```
function f = objfun(x)
f = exp(x(1)) * (4*x(1)^2 + 2*x(2)^2 + 4*x(1)*x(2) + 1);
```

Этап 2. Составление программы с использованием подходящей функции пакета (в данном случае – функции fminunc):

```
>> % Стартовое значение
>> x0 = [-1, 1];
>> % Задание опции использования алгоритма средней размерности
>> options = optimset('LargeScale', 'off');
>> % Поиск решения
```

```
>> [x,fval,exitflag,output] = fminunc('objfun',x0,options)
```

Выполнение программы приведет к следующему результату:

```
Optimization terminated successfully:  
Current search direction is a descent direction, and magni-  
tude of directional derivative in search direction less than  
2*options.TolFun  
x =  
    0.5000   -1.0000  
fval =  
    1.3031e-010  
exitflag =  
         1  
output =  
    iterations: 7  
    funcCount: 40  
    stepsize: 1  
    firstorderopt: 8.1995e-004  
    algorithm: 'medium-scale: Quasi-Newton line search'
```

Значения $x = [0.5000 \ -1.0000]$ и $fval = 1.3031e-010$ – искомое решение задачи. Значение $exitflag = 1$ дает информацию, что найдена точка минимума (возможно, локального).

Выходная структура (информация о результатах оптимизации) определяется идентификатором `output`:

число выполненных итераций (`iterations`): 7;
число вычислений функции (`funcCount`): 40;
шаг поиска (`stepsize`): 1;
степень оптимальности найденного решения (`firstorderopt`) – норма вектора-градиента в точке найденного решения: $8.1995e-004$;
использованный алгоритм (`algorithm`): квазиньютоновский с одномерной оптимизацией ('medium-scale: Quasi-Newton line search'), относящийся к числу алгоритмов средней размерности.

Задание к лабораторной работе

1. Ознакомиться с примерами использования программного пакета Optimization Toolbox в оптимизационных задачах.
2. Для предложенных задач реализовать алгоритм решения в соответствии с рассмотренными примерами.
- 3.

МЕТОДИКА ВЫПОЛНЕНИЯ РАБОТЫ

1. Открыть рабочее окно программы MATLAB 6.
2. Изучая теоретические сведения о пакете Optimization Toolbox, вводить в рабочем окне команды, приводимые в описании пакета и **выделенные желтым цветом**.

3. Закончив изучение теоретических сведений о пакете Optimization Toolbox, выполнить практическое задание.

Лабораторная работа №6

Методы нелинейного программирования. Минимизация с ограничениями.

Теперь рассмотрим задачу минимизации целевой функции $\min_x f(\mathbf{x}) = e^{x_1} (4x_1^2 + 2x_2^2 + 4x_1x_2 + 1)$, но при наличии ограничений в форме нелинейных неравенств:

$$\begin{aligned}x_1x_2 - x_1 - x_2 &\leq -1,5, \\x_1x_2 &\geq -10.\end{aligned}$$

Данная задача (см. таблицу 1) может быть решена с применением функции `fmincon` в соответствии с теми же этапами, что и предыдущая. Поскольку при использовании данной функции нелинейные ограничения должны быть представлены в виде $\mathbf{c}(\mathbf{x}) \leq 0$, перепишем их соответствующим образом:

$$\begin{aligned}x_1x_2 - x_1 - x_2 + 1,5 &\leq 0, \\-x_1x_2 - 10 &\geq 0.\end{aligned}$$

Этап 1. Составление m-файла (с именем `confun`), возвращающего значения левых частей ограничивающих неравенств:

```
function [c, ceq] = confun(x)
% Нелинейные ограничения в форме неравенств
c = [1.5 + x(1)*x(2) - x(1) - x(2); -x(1)*x(2) - 10];
% Нелинейные ограничения в форме равенств
ceq = [];
```

Этап 2. Составление программы минимизации:

```
>> x0 = [-1, 1]; % Задание начальных значений
>> options = optimset('LargeScale', 'off'); % Задание опций
>> % Поиск решения
>> [x, fval] = fmincon('objfun', x0, [], [], [], [], [], 'confun', options)
```

Результаты вычислений сообщают о найденном решении, количестве ограничениях и т. п.:

```
Optimization terminated successfully:
Search direction less than 2*options.TolX and maximum constraint violation is less than options.TolCon
Active Constraints:
    1
    2
x =
   -9.5474    1.0474
fval =
    0.0236
```

Минимизация с ограничениями в форме нелинейных неравенств

Теперь рассмотрим задачу минимизации той же целевой функции (в предыдущей лабораторной работе: Рассмотрим задачу нахождения значения переменных x_1 и x_2 , обеспечивающих решение задачи минимизации

$$\min_{\mathbf{x}} f(\mathbf{x}) = e^{x_1} (4x_1^2 + 2x_2^2 + 4x_1x_2 + 1), \text{ но при наличии ограничений в форме}$$

нелинейных неравенств:

$$\begin{aligned} x_1x_2 - x_1 - x_2 &\leq -1,5, \\ x_1x_2 &\geq -10. \end{aligned}$$

Данная задача может быть решена с применением функции `fmincon` в соответствии с теми же этапами, что и предыдущая. Поскольку при использовании данной функции нелинейные ограничения должны быть представлены в виде $\mathbf{c}(\mathbf{x}) \leq 0$, перепишем их соответствующим образом:

$$\begin{aligned} x_1x_2 - x_1 - x_2 + 1,5 &\leq 0, \\ -x_1x_2 - 10 &\geq 0. \end{aligned}$$

Этап 1. Составление m-файла (с именем `confun`), возвращающего значения левых частей ограничивающих неравенств:

```
function [c,seq] = confun(x)
% Нелинейные ограничения в форме неравенств
c = [1.5 + x(1)*x(2) - x(1) - x(2); -x(1)*x(2) - 10];
% Нелинейные ограничения в форме равенств
seq = [];
```

Этап 2. Составление программы минимизации:

```
>> x0 = [-1,1]; % Задание начальных значений
>> options = optimset('LargeScale','off'); % Задание опций
>> % Поиск решения
>> [x,fval] = fmincon('objfun',x0,[],[],[],[], [], 'confun', options)
```

Результаты вычислений сообщают о найденном решении, количестве ограничениях и т. п.:

```
Optimization terminated successfully:
Search direction less than 2*options.TolX and maximum constraint violation is less than options.TolCon
Active Constraints:
     1
     2
x =
    -9.5474     1.0474
fval =
     0.0236
```

Минимизация с дополнительными ограничениями на диапазоны изменения переменных

Функция `fmincon` может быть применена и для поиска решения в задаче минимизации, в которой допустимые значения переменных ограничены некоторыми диапазонами: $\mathbf{lb} \leq \mathbf{x} \leq \mathbf{ub}$.

Продолжим рассмотрение примера, введя дополнительные ограничения $x_1 \geq 0, x_2 \geq 0$, что эквивалентно заданию `lb = [0 0], ub = []`.

Программа оптимизации и результаты вычислений для данного случая приведены ниже:

```
>> x0 = [-1,1]; % Задание начальных значений
>> lb = [0,0]; % Задание нижних границ переменных
>> ub = []; % Сверху переменные не ограничены
>> options = optimset('LargeScale','off'); % Задание опций
>> % Поиск решения
>> [x,fval] = fmincon('objfun',x0,[],[],[],[],lb,
ub,'confun', options)
```

```
Optimization terminated successfully:
Search direction less than 2*options.TolX and maximum con-
straint violation is less than options.TolCon
Active Constraints:
     1
     3
x =
     0     1.5000
fval =
     8.5000
```

Найденное решение удовлетворяет ограничениям, наложенным на диапазоны изменения переменных. Проверим теперь выполнение ограничений в форме неравенств:

```
>> % Проверка выполнения ограничений в форме неравенств
>> [c,ceq] = confun(x)
c =
     0
    -10
ceq =
     []
```

Как следует из приведенного результата, ограничения в форме неравенств выполнены (ограничения в форме равенств отсутствуют).

Лабораторная работа №7

Оптимальное управление. Задача оптимального управления по быстродействию.

Задача достижения цели

Рассмотрим следующий пример.

Пусть некоторая замкнутая линейная динамическая система управления 3-го порядка описывается уравнениями

$$x = (\mathbf{A} + \mathbf{BK})x + \mathbf{B}u, y = \mathbf{C}x,$$

где матрицы

$$\mathbf{A} = \begin{bmatrix} -0,5 & 0 & 0 \\ 0 & -2 & 10 \\ 0 & 1 & -2 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 1 & 0 \\ -2 & 2 \\ 0 & 1 \end{bmatrix}, \quad \mathbf{C} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

отражающие динамические свойства объекта управления, являются заданными, а матрица \mathbf{K} –регулятора изменяемой.

Как известно, качество работы подобных систем (качество переходных процессов) определяется расположением на комплексной плоскости собственных чисел матрицы $\mathbf{A} + \mathbf{BK}$. Поставим задачу оптимизации таким образом: при задании диапазона возможных изменений элементов матрицы \mathbf{K} от -4 до +4 подобрать эти элементы таким образом, чтобы указанные собственные числа равнялись величинам $[-5, -3, -1]$. В подобной формулировке задача является многокритериальной (3 критерия) и соответствует задаче достижения цели, приведенной в [7].

Решение задачи, как и раньше, проведем поэтапно.

Этап 1. Создадим m-файл (с именем eigfun) для вычисления и упорядочивания по величине собственных чисел матрицы $\mathbf{A} + \mathbf{BK}$:

```
function F = eigfun(K,A,B,C) % Нахождение собственных чисел
F = sort(eig(A+B*K*C)); % Упорядочивание собственных чисел
```

Этап 2. Составление оптимизирующей программы:

```
>> A = [-0.5 0 0; 0 -2 10; 0 1 -2];
>> B = [1 0; -2 2; 0 1];
>> C = [1 0 0; 0 0 1];
>> K0 = [-1 -1; -1 -1]; % Задание начальных условий
>> % Задание вектора целей
>> goal = [-5 -3 -1];
>> weight = abs(goal) % Задание вектора весовых коэффициен-
```

тов

```
>> lb = -4*ones(size(K0)); % Нижние границы элементов матрицы K
```

```
>> ub = 4*ones(size(K0)); % Верхние границы элементов матрицы K
```

```
>> % Установка опции вывода информации
>> options = optimset('Display','iter');
>> [K,fval,attainfactor] = fgoalattain(@eigfun,K0,...
    goal,weight,[],[],[],[],lb,ub,[],options,A,B,C)
```

% Результаты вычислений:

	Attainment			Directional	
Iter	F-count	factor	Step-size	derivative	Procedure
1	6	1.885	1	1.03	
2	13	1.061	1	-0.679	
3	20	0.4211	1	-0.523	Hessian modified
4	27	-0.06352	1	-0.053	Hessian

						modified twice
5	34	-0.1571	1	-0.133		
6	41	-0.3489	1	-0.00768	Hessian modified	
7	48	-0.3643	1	-4.25e-005	Hessian modified	
8	55	-0.3645	1	-0.00303	Hessian modified twice	
9	62	-0.3674	1	-0.0213	Hessian modified	
10	69	-0.3806	1	0.00266		
11	76	-0.3862	1	-2.73e-005	Hessian modified twice	
12	83	-0.3863	1	-1.2e-013	Hessian modified twice	

Optimization terminated successfully:

Search direction less than 2*options.TolX and maximum constraint violation is less than options.TolCon

Active Constraints:

1
2
4
9
10

K =

-4.0000 -0.2564
-4.0000 -4.0000

fval =

-6.9313
-4.1588
-1.4099

attainfactor =

-0.3863

Выведенная информация имеет следующий характер:

- Iter – номер итерации;
- F-count – количество вычислений функции;
- Attainment factor – коэффициент (уровень) достижения целей;
- Step-size – шаг поиска;
- Directional derivative – норма градиента;

- Procedure – выполненная процедура (Hessian modified – гессиан модифицирован, Hessian modified twice – гессиан модифицирован дважды);
- Optimization terminated successfully – оптимизация проведена успешно;
- конечное значение матрицы

```
K =
    -4.0000    -0.2564
    -4.0000   -4.0000
```

- итоговые значения собственных чисел

```
fval =
    -6.9313
    -4.1588
    -1.4099
```

- итоговое значение коэффициента достижения целей attainfactor = -0.3863, что говорит о «перевыполнении» заданных целей в среднем на 38% (сравнение итоговых значений собственных чисел с заданными это подтверждает).

Полученные результаты не слишком близки к целевым. Поэтому для получения более приемлемого результата внесем коррекцию в программу оптимизации – зададим опцию точного достижения всех трех целей:

```
>> options = optimset('GoalsExactAchieve',3);
```

после чего повторим поиск оптимального решения:

```
>> [K,fval,attainfactor] = fgoalattain('eigfun',K0,goal,weight,[],[],[],lb,ub,[],options,A,B,C)
Optimization terminated successfully:
Magnitude of directional derivative in search direction
less than 2*options.TolFun and maximum constraint violation is
less than options.TolCon
Active Constraints:
     9
    10
    12
    14
K =
    -1.5954    1.2040
    -0.4201   -2.9046
fval =
    -5.0000
    -3.0000
    -1.0000
attainfactor =
     0
```

Теперь результат является вполне удовлетворительным. Подробную демонстрацию приведенного примера можно получить, используя функцию `goaldemo`.

РАЗДЕЛ III

МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ САМОСТОЯ- ТЕЛЬНОЙ РАБОТЫ

ЗАДАЧИ ДЛЯ САМОСТОЯТЕЛЬНОГО РЕШЕНИЯ

В этом разделе представлены задачи по линейному программированию и оптимальному управлению для самостоятельного решения. Задания выбирать по варианту, задание 1 по линейному программированию выполнять с помощью программных средств и оформлять в виде отчета в электронном виде, задание 2 предназначено для закрепления материала по оптимальному управлению, выполнять на практических занятиях и оформлять в виде отчета.

Задания по линейному программированию

Задачи линейного программирования взять из учебника [2] и выполнить задания 1,2.

Задание 1. Составить экономико-математические модели задач и решить их используя табличный симплекс-метод. Полученные результаты проанализировать.

Задание 2. По составленной экономико-математической модели написать программный код или использовать электронные таблицы или математические пакеты для реализации программы табличного симплекс-метода.

Задачи оптимального управления

Задание 1. Решить предложенные задачи, используя классические методы вариационного исчисления [2].

$$1. \int_{-\pi}^{\pi} x \cdot \sin t dt \rightarrow extr; \\ |\dot{x}| \leq 1, x(\pm\pi) = 0.$$

$$2. \int_0^{7\pi/4} x \cdot \sin t dt \rightarrow extr; \\ |\dot{x}| \leq 1, x(0) = 0.$$

$$3. \int_0^4 (\dot{x}^2 + x) dt \rightarrow extr; \\ |\dot{x}| \leq 1, x(4) = 0.$$

$$4. \int_0^{T_0} (\dot{x}^2 + x) dt \rightarrow extr; \\ |\dot{x}| \leq 1, x(0) = 0.$$

$$5. \int_0^{T_0} (\dot{x}^2 + x) dt \rightarrow extr; \\ |\dot{x}| \leq 1, x(0) = \xi.$$

$$6. \int_0^T (\dot{x}^2 + x) dt \rightarrow extr; \\ |\dot{x}| \leq 1, x(0) = \xi.$$

$$7. \int_0^{T_0} (\dot{x}^2 + x) dt \rightarrow extr; \\ |\dot{x}| \leq 1, x(T_0) = \xi.$$

$$8. \int_0^{T_0} (\dot{x}^2 + x) dt \rightarrow extr; \\ |\dot{x}| \leq 1, x(0) = 0, x(T_0) = 0.$$

$$9. \int_0^T (\dot{x}^2 + x) dt \rightarrow extr; \\ |\dot{x}| \leq 1, x(0) = 0, x(T) = \xi.$$

$$10. \int_0^{T_0} (\dot{x}^2 + x^2) dt \rightarrow extr; \\ |\dot{x}| \leq 1, x(0) = \xi.$$

$$11. \int_0^{T_0} (\dot{x}^2 + x^2) dt \rightarrow extr; \\ |\dot{x}| \leq 1, x(T_0) = \xi.$$

$$12. \int_0^1 x dt \rightarrow extr; \\ |\ddot{x}| \leq 2, x(0) = \dot{x}(0) = 0.$$

$$13. \int_0^1 x dt \rightarrow extr; \\ |\ddot{x}| \leq 2, x(1) = \dot{x}(1) = 0.$$

$$14. \int_0^1 x dt \rightarrow extr; \\ |\ddot{x}| \leq 2, x(0) = \dot{x}(1) = 0.$$

$$15. \int_0^2 x dt \rightarrow extr; \\ |\ddot{x}| \leq 2, x(0) + x(2) = \dot{x}(0) = 0.$$

$$16. \int_0^2 x dt \rightarrow extr; \\ |\ddot{x}| \leq 2, x(0) + x(2) = \dot{x}(2) = 0.$$

$$17. \int_0^1 x dt \rightarrow extr; \\ |\ddot{x}| \leq 2, x(0) + x(1) = \dot{x}(0) + \dot{x}(1) = 0.$$

$$18. \int_0^2 x dt \rightarrow extr; \\ |\ddot{x}| \leq 2, x(0) = x(2) = \dot{x}(0) = 0.$$

$$19. \int_0^2 x dt \rightarrow extr; \\ |\ddot{x}| \leq 2, \dot{x}(0) = \dot{x}(2) = x(2) = 0.$$

$$20. \int_0^4 x dt \rightarrow extr; \\ |\ddot{x}| \leq 2, x(0) + x(4) = 0, \dot{x}(0) = \dot{x}(4) = 0.$$

Задание 2. Решить предложенные задачи, используя принцип максимума Понтрягина [2].

1. $T \rightarrow \inf$;
 $|\ddot{x}| \leq 2, x(-1) = 1, x(T) = -1, \dot{x}(-1) = \dot{x}(T) = 0.$
2. $T \rightarrow \inf$;
 $|\ddot{x}| \leq 2, x(-1) = -1, x(T) = 1, \dot{x}(-1) = \dot{x}(T) = 0.$
3. $T \rightarrow \inf$;
 $|\ddot{x}| \leq 2, x(0) = 1, x(T) = 3, \dot{x}(0) = \dot{x}(T) = 0.$
4. $T \rightarrow \inf$;
 $|\ddot{x}| \leq 3, x(0) = 1, x(T) = -1, \dot{x}(0) = \dot{x}(T) = 0.$
5. $T \rightarrow \inf$;
 $-3 \leq \ddot{x} \leq 1, x(0) = 3, x(T) = -5, \dot{x}(0) = \dot{x}(T) = 0.$
6. $T \rightarrow \inf$;
 $0 \leq \ddot{x} \leq 1, x(0) = \xi_1, \dot{x}(0) = \xi_2, x(T) = \dot{x}(T) = 0.$
7. $T \rightarrow \inf$;
 $|\ddot{x}| \leq 1, x(0) = \xi_1, \dot{x}(0) = \xi_2, \dot{x}(T) = 0.$
8. $\int_0^2 |\ddot{x}| dt \rightarrow \inf$;
 $\ddot{x} \geq -2, x(0) = \dot{x}(0) = 0, x(2) = -3.$
9. $\int_0^2 |\ddot{x}| dt \rightarrow \inf$;
 $\ddot{x} \geq -2, x(0) = \dot{x}(2) = 0, x(2) = 3.$
10. $\int_0^2 \frac{t}{1 + \dot{x}^2} dt \rightarrow \inf$; (аэродинамическая
 $\dot{x} \geq 0, x(0) = 0, x(T_0) = \xi$
 задача Ньютона).

СПИСОК ЛИТЕРАТУРЫ

1. Аттеков А.В., Галкин С.В., Зарубин В.С. Методы оптимизации: Учеб. для вузов/ Под ред. В.С. Зарубина, А.П. Крищенко. – 2-е изд. стереотип.– М.: Изд-во МГТУ им. Н.Э. Баумана, 2003.
2. Алексеев В.М., Галеев Э.М., Тихомиров В.М. Сборник задач по оптимизации. Теория. Примеры. Задачи: Учеб. пособие. – 2-е изд. – М.: Физматлит, 2005.
3. Галеев Э.М. Оптимизация: теория, примеры, задачи. Учебное пособие. – М.: Элиториал УРСС, 2000.
4. Измайлов А.Ф., Солодов М.В. Численные методы оптимизации. – М.: Физматлит, 2003.
5. Карманов В.Г. Математическое программирование. - М.: Физматлит, 2000.
6. Дьяконов В., Круглов В. Математические пакеты расширения MATLAB. Специальный справочник. – СПб. : Питер, 2001. – 480 с. (Пакет Optimization Toolbox – с. 376-436.)
7. Ерёмин Е.Л. Теоретические основы автоматизированного управления: Серия "Курс лекций АмГУ", Выпуск №1. Благовещенск: Изд-во Амурского государственного университета, 1998.
8. Трифонов А.Г. "Постановка задачи оптимизации и численные методы ее решения". Сайт MATLAB. Exponenta

Наталья Петровна Семичевская,
доцент кафедры ИиУС АмГУ, канд. техн. наук

Задачи и методы оптимизации. Учебно-методическое пособие.

Заказ 786.