

И.К. Васильева, П.Е. Ельцов

ЧИСЛЕННЫЕ МЕТОДЫ ОПТИМИЗАЦИИ

2009

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ УКРАИНЫ
Национальный аэрокосмический университет им. Н.Е. Жуковского
"Харьковский авиационный институт"

И.К. Васильева, П.Е. Ельцов

ЧИСЛЕННЫЕ МЕТОДЫ ОПТИМИЗАЦИИ

Учебное пособие по лабораторному практикуму

Харьков «ХАИ» 2009

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ УКРАИНЫ
Национальный аэрокосмический университет им. Н.Е. Жуковского
"Харьковский авиационный институт"

И.К. Васильева, П.Е. Ельцов

ЧИСЛЕННЫЕ МЕТОДЫ ОПТИМИЗАЦИИ

Учебное пособие по лабораторному практикуму

Харьков «ХАИ» 2009

УДК 621.396.6

Васильева И.К. Численные методы оптимизации : учеб. пособие по лаб. практикуму / И.К. Васильева, П.Е. Ельцов. – Х. : Нац. аэрокосм. ун-т "Харьк. авиац. ин-т", 2009. – 65 с.

Описаны лабораторные работы по дисциплине "Методы оптимизации", которая входит в программу подготовки бакалавров по направлению "Геодезия, картография и землеустройство". Изложены теоретические сведения, приведены постановка задач, методика их решения и примеры выполнения работ.

Лабораторные работы ориентированы на использование программы математических расчетов и моделирования MathCAD.

Для студентов факультета радиотехнических систем летательных аппаратов.

Ил. 16. Табл. 17. Библиогр.: 9 назв.

Рецензенты: д-р техн. наук, проф. В.А. Батаев,
канд. физ.-мат. наук А.Н. Роечко

© Национальный аэрокосмический университет им. Н.Е. Жуковского
"Харьковский авиационный институт", 2009 г.

ВВЕДЕНИЕ

Задачи оптимизации (или экстремальные) встречаются там, где исследуемое явление (в физике, технике, экономике) допускает множество ситуаций в зависимости от условий наблюдения. Тогда задачей является поиск оптимальных ситуаций – наилучших в каком-либо смысле. Чтобы избежать субъективизма и неопределённости при оценке, необходимо, прежде всего, формализовать понятие оптимальности ситуации.

Пусть $X \in R^n$ – некоторое подмножество пространства, а $f(\vec{x})$ – числовая функция, определённая на нём. Среди элементов множества X нужно выбрать наиболее предпочтительное (с определённой точки зрения) решение. Сравнение решений по предпочтительности осуществляется с помощью $f(\vec{x})$, которая называется **целевой функцией**. Выделяют два варианта сравнения произвольной пары решений $\vec{x}^{(1)}$ и $\vec{x}^{(2)}$: 1) считают, что $\vec{x}^{(1)}$ предпочтительнее $\vec{x}^{(2)}$, если $f(\vec{x}^{(1)}) < f(\vec{x}^{(2)})$ – это задача минимизации; 2) $\vec{x}^{(1)}$ предпочтительнее $\vec{x}^{(2)}$, если $f(\vec{x}^{(1)}) > f(\vec{x}^{(2)})$ – задача максимизации. Так как путем преобразования $f(\vec{x}) = -f(\vec{x})$ всегда можно перейти от одной задачи к другой, рассматривают, как правило, задачи минимизации.

Таким образом, целевая функция $f(\vec{x})$ позволяет количественно сравнить два альтернативных решения. С математической точки зрения целевая функция описывает некоторую $(n+1)$ -мерную поверхность, значение которой определяется проектными параметрами. Если имеется только один проектный параметр, то целевую функцию можно представить кривой на плоскости. Если проектных параметров два, то целевая функция является поверхностью в пространстве трех измерений и т.д.

Проектными параметрами могут быть любые основные или производные величины, служащие для количественного описания оп-

тимизируемой системы: неизвестные значения длины, массы, времени, температуры, номиналы радиоэлементов и т.д. Число проектных параметров (x_1, x_2, \dots, x_n) характеризует степень сложности данной задачи проектирования.

Пространство проектирования – область, определяемая всеми n проектными параметрами. Пространство проектирования обычно ограничено условиями, связанными с физическим смыслом задачи. Ограничения разделяют на две группы: ограничения-равенства и ограничения-неравенства.

Ограничения-равенства $\varphi_i(\vec{x}) = 0$ ($i = 1, \dots, m$) – это зависимости между проектными параметрами, которые нужно учитывать при отыскании решения. Каждое из этих ограничений геометрически представляет собой некую гиперповерхность и уменьшает размерность пространства на единицу, поэтому независимыми являются только $(n - m)$ переменных, $m < n$. Если какое-либо из этих соотношений можно разрешить относительно одного из параметров, то данный параметр исключают из процесса оптимизации; таким образом, упрощается решение задачи.

Ограничения-неравенства имеют вид $\psi_j(\vec{x}) \leq 0$, $j = 1, \dots, p$. Эти условия выделяют в пространстве n -мерную область, ограниченную гиперповерхностями $\psi_j(\vec{x}) = 0$, поэтому число этих условий может быть любым. Очень часто в связи с ограничениями-неравенствами оптимальное решение соответствует не экстремуму целевой функции, а одной из границ допустимой области проектирования.

При решении многих технических задач целевую функцию принимают в качестве меры “нежелательных” свойств объекта или стоимости, которую нужно минимизировать, либо рассматривают как меру “качества”, которую нужно максимизировать. При этом минимизация или максимизация должна быть выполнена при условии соблюдения определённых ограничений. Это справедливо применительно к самым различным объектам: электромеханическим устройствам, электрическим схемам, системам управления. Вопрос о том, оптимальна

или неоптимальна какая-либо совокупность расчётных данных, в общем случае зависит от выбора критерия. Примеры целевых функций, часто встречающихся в инженерной практике: стоимость, масса, прочность, габаритные размеры, КПД, коэффициент усиления, полоса пропускания и т.п. В некоторых задачах оптимизации требуется введение более одной целевой функции. Иногда одна из них может быть несовместима с другой. Например, одновременно требуется обеспечить максимальную прочность, минимальную массу и минимальную стоимость. В этих случаях вводят систему приоритетов и ставят в соответствие каждой целевой функции некоторый безразмерный множитель. В результате появляется “функция компромисса”, позволяющая в процессе оптимизации пользоваться одной составной целевой функцией.

Таким образом, задача оптимизации формулируется как оптимизация заданной целевой функции при условии выполнения ряда заданных ограничений.

Оптимизация – процесс или последовательность операций, позволяющих получить уточненные значения проектных параметров. Любое решение, удовлетворяющее ограничениям задачи, называется *допустимым*. Лучшее из допустимых решений в смысле заданного критерия называется *оптимальным*. Интерес представляют не только оптимальные, но и допустимые решения: есть ряд задач, для которых оптимальное решение неизвестно или практически неприемлемо из-за своей сложности. Изучая допустимые решения, близкие к оптимальным, можно оценить, насколько чувствительны оптимальные решения к небольшим изменениям в формулировках задач (например, изменение количества ограничений, учет практических соображений, не входящих в целевую функцию).

Локальный оптимум – точка пространства проектирования, в которой целевая функция имеет наибольшее значение по сравнению с ее значениями во всех других точках ее ближайшей окрестности. *Глобальный оптимум* – оптимальное решение для всего пространства проектирования.

ОБЩАЯ ХАРАКТЕРИСТИКА МЕТОДОВ МИНИМИЗАЦИИ

Аналитическое решение задачи отыскания экстремума $f(\bar{x})$ не всегда можно получить. В большинстве случаев эту задачу решают численно с помощью ЭВМ. При этом различают два подхода: первый – заключается в замене задачи на экстремум задачей численного решения системы уравнений $f'(\bar{x}) = \mathbf{0}$ в случае безусловного экстремума или аналогичной системы для функций Лагранжа

$$F(\bar{x}, \bar{\lambda}) = f(\bar{x}) + \sum_{i=1}^m \lambda_i \varphi_i(\bar{x}) \text{ или } F(\bar{x}, \bar{\lambda}) = f(\bar{x}) + \sum_{j=1}^p \lambda_j \psi_j(\bar{x})$$

в случае условного экстремума; второй – не использует непосредственно необходимые условия экстремума. Он заключается в построении каким-либо способом последовательности точек $\bar{x}^{(1)}, \bar{x}^{(2)}, \dots, \bar{x}^{(k)}$, в которых целевая функция убывает; предел этой последовательности принимают за искомую точку минимума. Эти способы, как правило, более эффективны, поскольку позволяют полнее учитывать специфику задач минимизации.

Численные методы оптимизации классифицируют в соответствии с порядком производных: в *методах нулевого порядка* используется информация только о значениях целевой функции, в *методах первого порядка* требуется вычисление её первой производной, а в *методах второго порядка* дополнительно к этому необходимо знание вторых производных. Алгоритмы более высоких порядков в задачах оптимизации практически не применяются.

Работа алгоритма состоит из двух этапов. На первом – вычисляются предусмотренные алгоритмом характеристики задачи (значения целевой функции, функций ограничений, задающих допустимое множество, их производные). На втором этапе по полученной информации строится приближение к решению – искомой точке минимума \bar{x}^* или, если такая точка не единственная, к множеству точек минимума.

Для задания алгоритма достаточно указать способ выбора точек вычисления (конечно, при условии, что уже решен вопрос о том, ка-

кие именно характеристики задачи следует вычислять; так, для минимизации недифференцируемой функции нельзя воспользоваться алгоритмом, предусматривающим вычисление в произвольных точках градиента функции). Если все точки выбираются одновременно до начала вычислений, то алгоритм называется *пассивным*. Пассивные алгоритмы необходимо применять, например, в связи с условиями постановки и проведения физических экспериментов, результатами которых являются значения целевой функции. Однако для решения большинства задач точки вычисления выбираются по очереди, т.е. $\bar{x}^{(i+1)}$ определяется, когда уже известны точки предыдущих вычислений $\bar{x}^{(1)}, \dots, \bar{x}^{(i)}$, и в каждой из них выполнены предусмотренные алгоритмом вычисления, результаты которых обозначим $y^{(1)}, \dots, y^{(i)}$. Такие алгоритмы называют *последовательными*. На практике используют наиболее простые виды зависимости, например, $\bar{x}^{(i+1)}(\bar{x}^i, y^i)$ (т.е. выбор точки очередного вычисления зависит лишь от точки предыдущего вычисления и полученного результата) или $\bar{x}^{(i+1)}(\bar{x}^i, a_1 y^1 + \dots + a_i y^i)$ (от точки предыдущего вычисления и линейной комбинации всех результатов).

Для записи методов минимизации применяют соотношения вида $\bar{x}^{(k+1)} = \bar{x}^{(k)} + \alpha_k \vec{h}^{(k)}$, где $\alpha_k \in \mathbf{R}$, $k = 1, 2, \dots$ – номер шага (итерации) метода. Конкретный алгоритм определяется заданием начальной точки $\bar{x}^{(0)}$, правилами выбора векторов $\vec{h}^{(k)}$ и чисел α_k на основании полученной в результате вычислений информации, а также условием останова счёта.

Вектор $\vec{h}^{(k)}$ указывает направление $(k + 1)$ -го шага, а коэффициент α_k равен длине этого шага. Следует иметь в виду, что при $\|\vec{h}^{(k)}\| \neq 1$ длина отрезка, соединяющего точки $\bar{x}^{(k)}$ и $\bar{x}^{(k+1)}$, не равна $|\alpha_k|$, длина шага – величина $\|\alpha_k \vec{h}^{(k)}\|$.

Для задач без ограничений любое направление возможно, но не все они приемлемы. Очевидно, что не следует использовать направление \vec{h} , для которого не выполняется условие $f(\vec{x} + \alpha\vec{h}) < f(\vec{x})$ при достаточно малом α , т.е. если значение функции не уменьшается в направлении \vec{h} хотя бы вблизи точки \vec{x} .

Направление \vec{h} в точке \vec{x} называется *приемлемым*, если скалярное произведение вектора \vec{h} и вектора-градиента целевой функции в этой точке отрицательно:

$$\langle \nabla f(\vec{x}), \vec{h} \rangle < 0.$$

Для задач с ограничениями приемлемые направления также должны удовлетворять этому условию, кроме того, при достаточно малых α точка $\vec{x} + \alpha\vec{h}$ не должна нарушать ни одного из ограничений.

Обычно название метода оптимизации определяется способом выбора $\vec{h}^{(k)}$, а его различные варианты – способами выбора α_k .

Среди методов минимизации можно условно выделить конечношаговые и бесконечношаговые методы. *Конечношаговыми* (конечными) называются методы, обеспечивающие нахождение точного решения за конечное число шагов. Такие методы можно построить только для некоторых специальных задач оптимизации, например, задач линейного программирования. Для бесконечношаговых методов точное решение может быть получено в пределе. Важная характеристика бесконечношаговых методов – сходимость.

Метод сходится, если $\vec{x}^{(k)} \rightarrow \vec{x}^*$ при $k \rightarrow \infty$, где \vec{x}^* – решение задачи. Если $f(\vec{x}^{(k)}) \rightarrow f(\vec{x}^*)$, то метод сходится по функции, при этом последовательность $\vec{x}^{(k)}$ называется минимизирующей.

Эффективность сходящегося метода можно охарактеризовать с помощью понятия *скорость сходимости*.

Алгоритм сходится линейно (или со скоростью геометрической прогрессии), если $\|\vec{x}^{(k+1)} - \vec{x}^*\| \leq q \|\vec{x}^{(k)} - \vec{x}^*\|$, где константа $q \in (0, 1)$.

Скорость сходимости сверхлинейна, если $\|\bar{x}^{(k+1)} - \bar{x}^*\| \leq q_k \|\bar{x}^{(k)} - \bar{x}^*\|$, где $q \rightarrow 0$ при $k \rightarrow \infty$, и квадратична, если $\|\bar{x}^{(k+1)} - \bar{x}^*\| \leq c \|\bar{x}^{(k)} - \bar{x}^*\|^2$, $c \geq 0$. Наибольшей скоростью сходимости обладают методы высокого (второго) порядка. Это означает, что с их помощью можно достигнуть минимума с заданной точностью за наименьшее число шагов (итераций). Однако вычисление старших производных требует большего числа арифметических действий, чем вычисление значения целевой функции, что приводит к увеличению затрат машинного времени. Кроме того, иногда $f(\bar{x})$ настолько сложная функция, что её первая и вторая производные не могут быть получены аналитически, а их численные приближения оказываются очень грубыми. Таким образом, метод, эффективный для одних задач, может быть неприемлемым для других.

Рассмотрим вопрос о критериях остановки счёта. Чаще всего используют такие условия прекращения вычислений:

$$\begin{aligned} \|\bar{x}^{(k+1)} - \bar{x}^{(k)}\| &\leq \varepsilon_1; \\ \|f(\bar{x}^{(k+1)}) - f(\bar{x}^{(k)})\| &\leq \varepsilon_2; \\ \|f'(\bar{x}^{(k+1)})\| &\leq \varepsilon_3, \varepsilon_i \ll 1 \end{aligned}$$

(последний критерий относится только к задачам безусловной оптимизации).

Эти критерии характеризуют абсолютную погрешность вычислений. В некоторых задачах удобнее применять аналогичные критерии, основанные на понятии относительной погрешности:

$$\begin{aligned} \|\bar{x}^{(k+1)} - \bar{x}^{(k)}\| &\leq \delta_1 (1 + \|\bar{x}^{(k+1)}\|); \\ \|f(\bar{x}^{(k+1)}) - f(\bar{x}^{(k)})\| &\leq \delta_2 (1 + \|f(\bar{x}^{(k+1)})\|); \\ \|f'(\bar{x}^{(k+1)})\| &\leq \delta_3 (1 + \|f'(\bar{x}^{(k+1)})\|). \end{aligned}$$

СРАВНЕНИЕ ЭФФЕКТИВНОСТИ ЧИСЛЕННЫХ МЕТОДОВ ОДНОМЕРНОЙ БЕЗУСЛОВНОЙ МИНИМИЗАЦИИ

Цель работы:

- изучить методику построения алгоритмов, применяемых для решения задач минимизации функций одной переменной при отсутствии ограничений на область допустимых значений;
- ознакомиться с критериями эффективности численных методов оптимизации и научиться выбирать алгоритм, наиболее подходящий для решения конкретной задачи;
- получить навыки реализации алгоритмов одномерной безусловной минимизации в среде MathCAD.

Теоретические сведения

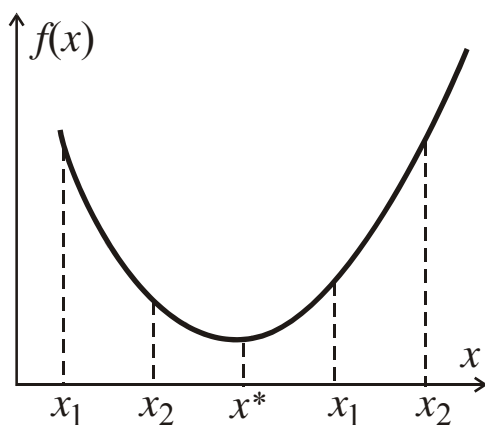


Рис. 1.1. Поиск отрезка неопределенности

Большинство численных методов оптимизации базируются на использовании следующей леммы: пусть $f(x)$ унимодальна на X (непрерывна и имеет единственный минимум x^*); $x_1, x_2 \in X$ и $x_1 < x_2$ (рис. 1.1). Тогда, если $f(x_1) \leq f(x_2)$, то $x^* \leq x_2$, если $f(x_1) \geq f(x_2)$, то $x^* \geq x_1$. На основа-

нии этой леммы можно указать отрезок, в котором содержится x^* , поэтому будем искать минимум $f(x)$ при $x \in [a, b]$. Кроме того, считаем, что число вычислений целевой функции задано и равно n .

В **методе пассивного поиска** интервал неопределенности $[a, b]$ для одномерной унимодальной функции делится на несколько равных частей, и затем вычисляются значения целевой функции в

узлах полученной сетки. В результате интервал неопределенности сужается до двух шагов сетки.

Алгоритм пассивного поиска. Разделим $[a, b]$ на $n + 1$ часть:

$x_i = a + \frac{b-a}{n+1}i$, $i = 1, \dots, n$. Вычислим все значения $f(x_i)$, сравним

их между собой и найдём $f(x_j) = \min\{f(x_i)\}$. Тогда на основании

леммы заключаем, что $x^* \in [x_{j-1}, x_{j+1}]$ (интервал неопределённости

или отрезок локализации минимума) и за $\min f(x)$ примем

$f(x_j)$. При этом длина отрезка локализации минимума

$L_n = x_{j+1} - x_{j-1} = 2\frac{b-a}{n+1}$, а погрешность в определении x^* $\varepsilon \leq L_n/2$.

Следующая группа методов относится к классу алгоритмов активного поиска (последовательных), поскольку в них очередное приближение x^{k+1} зависит от результатов предыдущих вычислений.

Метод дихотомии (деления пополам). Обычно под этим методом понимают способ численного решения уравнения $y(x) = 0$. Он

заключается в том, что вычисляются три значения функции в точках a , b и $(a+b)/2$, т.е. на концах и в середине отрезка. После этого про-

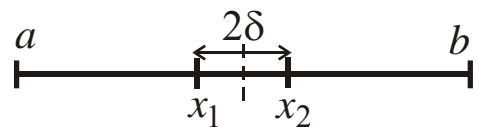
водится сравнение знаков функций в этих точках и отбрасывается либо первая, либо вторая половина отрезка. При поиске минимума

недостаточно вычислить функцию только в одной точке внутри исследуемого отрезка, так как будет неизвестно, справа или слева от

неё находится точка минимума. Чтобы сделать подобный вывод, нужно вычислить значения функции в двух внутренних точках – x_1 и

x_2 . Так как на каждом шаге вычисляются два значения функции, то $n = 2k$ (k – номер шага).

Пусть $x_1 = \frac{a+b}{2} - \delta$, $x_2 = \frac{a+b}{2} + \delta$,



где δ – малое число (рис. 1.2).

Рис. 1.2. Выбор точек по методу дихотомии

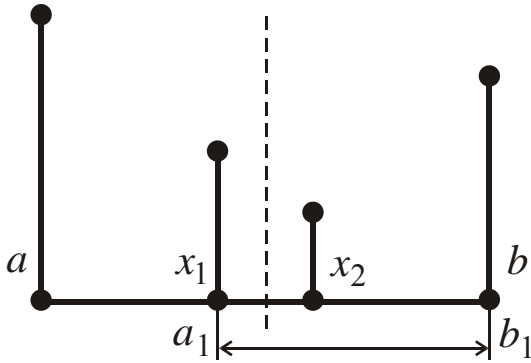


Рис. 1.3. Метод дихотомии

Вычислим $f(x_1)$, $f(x_2)$ и сравним их между собой (рис. 1.3).

Если $f(x_1) \leq f(x_2)$, то $x^* \in [a, x_2]$, отрезок $[x_2, b]$ можно отбросить, т.е. $a_1 = a$, $b_1 = x_2$.

Если $f(x_1) \geq f(x_2)$, то отбрасывают $[a, x_1]$: $a_1 = x_1$, $b_1 = b$.

На следующем шаге разделим интервал $[a_1, b_1]$ пополам, вновь возьмём две точки (x'_1, x'_2) на расстоянии δ от его середины и повторим процедуру.

После первого шага длина отрезка локализации $L_1 = \frac{b-a}{2} + \delta$, после второго — $L_2 = \frac{1}{2} \left(\frac{b-a}{2} + \delta \right) + \delta = \frac{b-a}{4} + \frac{3}{2} \delta$, после k -го ($k = n/2$) — $L_k = \frac{b-a}{2^k} + \frac{2^k - 1}{2^{k-1}} \delta = \frac{b-a}{2^{(n/2)}} + \frac{2^{(n/2)} - 1}{2^{(n/2)-1}} \delta \approx \frac{b-a}{2^{(n/2)}} + 2\delta$.

Выбором δ длину L_k можно сделать сколь угодно близкой к $\frac{b-a}{2^{(n/2)}}$.

Сравнивая эту величину с длиной отрезка локализации в методе пассивного поиска, видно, что метод дихотомии существенно эффективнее. Так, для уменьшения L_k с 1 до 0,01 требуется 199 шагов по методу пассивного поиска (если считать, что $f(a)$ и $f(b)$ уже известны) и всего 14 шагов по методу дихотомии (если пренебречь величиной δ). Из этого не следует, что метод пассивного поиска неприемлем; он может оказаться достаточно оптимальным при параллельных вычислениях. Кроме того, метод пассивного поиска может входить в качестве составной части в алгоритмы минимизации многоэкстремальных функций.

Таким образом, в методе дихотомии интервал неопределенности разделяют каждый раз надвое. Этот прием повышает эффективность

поиска. На первом этапе необходимо вычислить и сравнить между собой три значения целевой функции – на концах и в середине интервала, на следующем этапе – только два значения (вместо трех) (так как значение целевой функции в середине нового интервала уже известно).

Метод Фибоначчи более оптимальный, чем метод дихотомии. В методе дихотомии на каждом шаге теряется одна точка, например x_1 , поскольку на следующем шаге вычисляются две новые точки (x'_1, x'_2) вблизи центра $[a, x_2]$, а информация о значении $f(x_1)$ не используется. Поэтому метод дихотомии может быть модифицирован следующим образом: пусть на одном шаге каким-либо образом выбрали две точки x_1, x_2 внутри $[a, b]$, вычислили $f(x_1)$ и $f(x_2)$, сравнили их между собой и отбросили часть отрезка (для определённости $[x_2, b]$) (рис. 1.4).



Рис. 1.4. К выбору точек по методу Фибоначчи

На втором шаге воспользуемся тем, что в одной точке (x_1) нового интервала неопределенности $[a, x_2]$ значение функции $f(x_1)$ уже известно, поэтому для продолжения итерационного процесса достаточно вычислить функцию только в одной новой точке x_3 (а не в двух, как в методе дихотомии). Поскольку на втором шаге будет отброшен один из двух отрезков $[a, x_3]$ или $[x_1, x_2]$ (какой именно – заранее известно), то следует выбрать положение точки x_3 таким, чтобы длины этих отрезков были равны. Таким образом, на каждом шаге (начиная с первого) положение новой точки надо выбирать так, чтобы она была симметричной уже имеющейся точке относительно середины отрезка.

Если обозначить длину отрезка локализации на k -м шаге через L_k , то на трёх последовательных шагах эти величины связаны соотношением $L_{k-2} = L_{k-1} + L_k$ (рис. 1.5). Чтобы в наибольшей степени уменьшить последний отрезок неопределенности L_n , расстояние

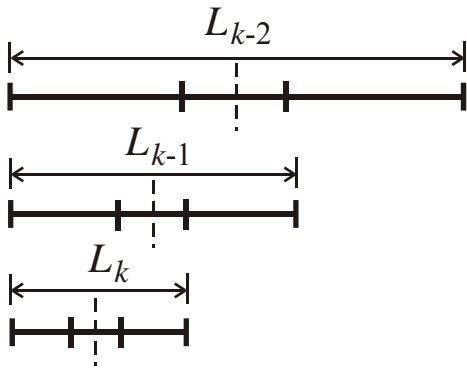


Рис. 1.5. Правило симметрии точек

между точками x_n и x_{n-1} должно быть минимально возможным: $L_{n-1} = 2L_n - \varepsilon$.

Тогда $L_{n-k} = F_{k+1}L_n - F_{k-1}\varepsilon$, $k = 1, \dots, n-1$, где F_k – числа Фибоначчи:

$$\begin{cases} F_0 = F_1 = 1; \\ F_k = F_{k-1} + F_{k-2}, \quad k = 2, \dots, n; \end{cases}$$

$$L_n = \frac{L_1}{F_n} + \frac{F_{n-2}}{F_n} \varepsilon.$$

Выбирая $\varepsilon \ll L_1/F_{n-2}$, после n вычислений уменьшаем первоначальный интервал примерно в $F_n \approx \tau^{n+1}/\sqrt{5}$ раз, $\tau = (1 + \sqrt{5})/2 \approx 1,618$. (Для стократного уменьшения достаточно сделать $n = 11$ шагов против 14 для метода дихотомии).

Таким образом, для расчёта методом Фибоначчи необходимо:

1) оценить желаемую точность L_n вычисления точки минимума и значение параметра ε и с помощью соотношения $L_n/L_1 \approx 1/F_n \approx \sqrt{5}/\tau^{n+1}$ определить требуемое число шагов n ;

2) вычислить числа F_n, F_{n-1} и рассчитать величину L_2 как

$$L_2 = \frac{F_{n-1}}{F_n} L_1 + \frac{(-1)^n}{F_n} \varepsilon;$$

3) рассчитать координаты точек $x_1 = b - L_2, x_2 = a + L_2$.

Последующие точки должны располагаться симметрично предшествующим относительно центра отрезков, т.е. $x_{k+1} = a_k + b_k - x_k$. Однако такой способ расчёта приводит к быстрому нарастанию погрешности. Поэтому в качестве очередного приближения x_{k+1} лучше брать одну из двух точек, рассчитываемых по формулам $x'_{k+1} = a_k + L_{k+1}, x''_{k+1} = b_k - L_{k+1}$, а именно ту из них, которая наиболее удалена от x_k .

Метод золотого сечения. Недостаток метода Фибоначчи заключается в том, что необходимо заранее выбрать количество шагов n . Однако не всегда заранее известно, какое число шагов потребуется для поиска минимума. В методе золотого сечения стратегия поиска такая же, как и в методе Фибоначчи, также используется соотношение $L_{k-1} = L_k + L_{k+1}$, выражающее закон симметрии точек, но расположение точек выбирается по правилу золотого сечения: отрезок делится на две неравные части так, чтобы отношение длины всего отрезка к длине его большей части равнялось отношению длины большей части к длине меньшей: $\frac{L_{k-1}}{L_k} = \frac{L_k}{L_{k+1}} = \tau \approx 1,618$. Таким об-

разом, первые два вычисления значения функции позволяют уменьшить интервал неопределённости в 1,618 раза, а длина интервала после n шагов $L_n = L_1 / \tau^{n-1}$, т.е. окончательный интервал лишь на 17% больше, чем в методе Фибоначчи, зато заранее выбирать число шагов не нужно.

Применение методов исключения интервалов накладывает единственное ограничение на исследуемую целевую функцию – унимодальность. Следовательно, рассмотренные методы можно использовать для анализа как непрерывных, так и разрывных и дискретных функций. Логическая структура поиска основана на простом сравнении значений функции в двух пробных точках.

Методы парабол. Рассмотренные выше методы (все нулевого порядка) – надёжные, но относительно медленные. Если целевая функция $f(x)$ дифференцируема, то можно построить более быстрые (но и более “капризные”) методы, основанные на параболической аппроксимации $f(x)$.

Пусть каким-либо способом (например графическим) получено нулевое приближение $x^{(k)}$ ($k = 0$).

Разложим $f(x)$ в точке $x^{(k)}$ в ряд Тейлора, ограничившись тремя членами, т.е. аппроксимировав $f(x)$ параболой:

$$f(x) \approx f(x^{(k)}) + (x - x^{(k)})f'(x^{(k)}) + \frac{1}{2}(x - x^{(k)})^2 f''(x^{(k)}).$$

Найдём минимум этой квадратичной функции, используя необходимое условие экстремума $f'(x) = 0$. Полученную точку примем за очередное приближение $x^{(k+1)}$ к истинной точке минимума x^* :

$$f'(x) \approx f'(x^{(k)}) + (x^{(k+1)} - x^{(k)})f''(x^{(k)}) = 0,$$

$$x^{(k+1)} = x^{(k)} - \frac{f'(x^{(k)})}{f''(x^{(k)})}.$$

Вблизи экстремума с ненулевой f'' этот метод сходится квадратично. Если $f''(x^*) = 0$, то сходимость в достаточно малой окрестности x^* тоже имеется, но более медленная (линейная).

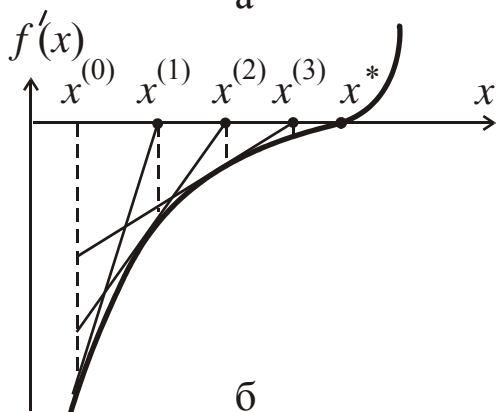
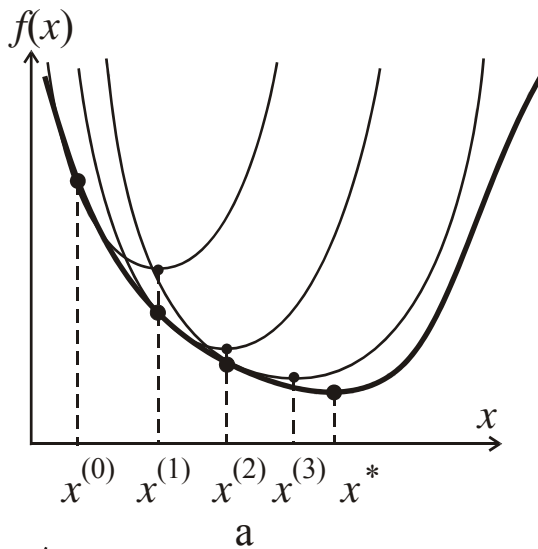


Рис. 1.6. Аппроксимации целевой функции $f(x)$ (а) и ее производных в точках $x^{(k)}$ (б)

сти x^* тоже имеется, но более медленная (линейная).

Таким образом, исходная кривая $f(x)$ заменяется интерполяционной параболой (рис. 1.6), построенной по трём точкам: $x^{(k)} - h$, $x^{(k)}$, $x^{(k)} + h$. На каждой итерации вычисляют три значения функции. Можно уменьшить число арифметических операций, строя параболу по уже вычисленным ранее точкам $x^{(k-2)}$, $x^{(k-1)}$, $x^{(k)}$.

Если для $f'(x)$ и $f''(x)$ аналитически получаются громоздкие выражения, то производные целевой функции можно заменить конечно-разностными аппроксимациями; обычно используют симметричные разности с постоянным шагом h :

$$y'(x) = \frac{dy}{dx} \approx \frac{\Delta y}{\Delta x} = \frac{\Delta y}{h};$$

$$y'(x_i) \approx \left\{ \frac{y_{i+1} - y_i}{h}; \frac{y_i - y_{i-1}}{h}; \frac{y_{i+1} - y_{i-1}}{2h} \right\};$$

$$y''(x_i) \approx \frac{y_{i+1} - 2y_i + y_{i-1}}{h^2}.$$

Тогда

$$x^{(k+1)} = x^{(k)} - \frac{h}{2} \frac{f(x^{(k)} + h) - f(x^{(k)} - h)}{f(x^{(k)} + h) - 2f(x^{(k)}) + f(x^{(k)} - h)}, \quad h^2 \leq \varepsilon,$$

где ε – желаемая точность.

Для того, чтобы алгоритм работал надежно, необходимо предусмотреть ряд дополнительных проверок:

1. Следует контролировать направление поиска, так как условие $f'(x) = 0$ может привести не к минимуму, а к максимуму или седловой точке. Для этого нужно проверять знак второй производной (аналитической или разностной). Если $f''(x) < 0$, делаем шаг в противоположном направлении.

2. Двигаясь к минимуму, надо его не пропустить, т.е. проверять условие $f(x^{(k+1)}) \leq f(x^{(k)})$. Если оно не выполняется, то последнюю итерацию пересчитывают с уменьшенным (например вдвое) шагом и вновь проверяют условие.

Таким образом, теоретически методы парабол обладают более высокой скоростью сходимости, чем методы нулевого порядка. Фактическая же скорость работы зависит от степени продуманности алгоритма и качества написания программы.

Постановка задачи

Требуется найти с точностью $\varepsilon \leq 0,001$ минимум функции $f(x)$ на отрезке $[a, b]$ (согласно заданному варианту).

Необходимо разработать и реализовать в системе MathCAD следующие алгоритмы одномерной минимизации:

- пассивный поиск;

- дихотомия;
- метод золотого сечения;
- метод парабол.

Сравнить эффективность указанных алгоритмов по критериям:

- число итераций для достижения заданной точности ε ;
- количество арифметических операций;
- универсальность по отношению к виду целевой функции.

Примеры реализации в MathCAD алгоритмов одномерной минимизации даны в прил. 1.

Контрольные вопросы

1. Что такое целевая функция? Привести примеры.
2. Как выбрать интервал неопределенности?
3. Дайте общую характеристику изученных методов.
4. Назовите критерии эффективности методов оптимизации.
5. Что такое «скорость сходимости» метода?

Варианты заданий

Номер варианта	$f(x)$	$[a, b]$
1	$(1 - e^x \sin x)^2$	$[0, 1]$
2	$\frac{x}{x^2 + 1}$	$[-2, 0]$
3	$2x^2 + 3e^{-x}$	$[0, 1]$
4	$-e^{-x} \ln x$	$[1, 3]$
5	$e^x(x^2 - 1)$	$[0, 2]$
6	$-x \sin(\ln(x))$	$[10, 12]$
7	$x \exp\left(\frac{1}{x}\right)$	$[0,5, 1,5]$
8	$\sqrt{1 + x^2} \sin(x)$	$[4, 6]$

МНОГОМЕРНАЯ БЕЗУСЛОВНАЯ МИНИМИЗАЦИЯ

Цель работы:

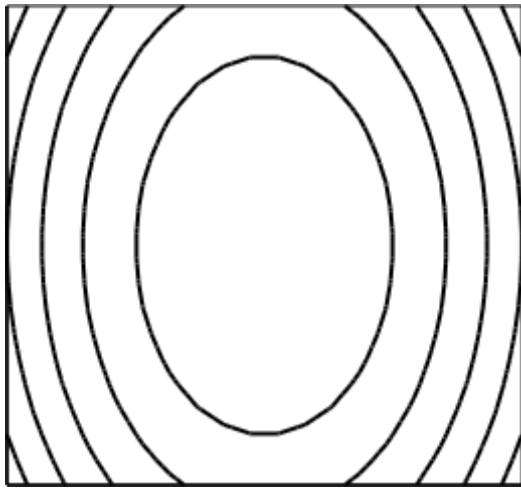
- изучить наиболее распространенные алгоритмы, применяемые для решения многомерных задач минимизации;
- получить навыки реализации алгоритмов минимизации функций двух переменных в среде MathCAD.

Теоретические сведения

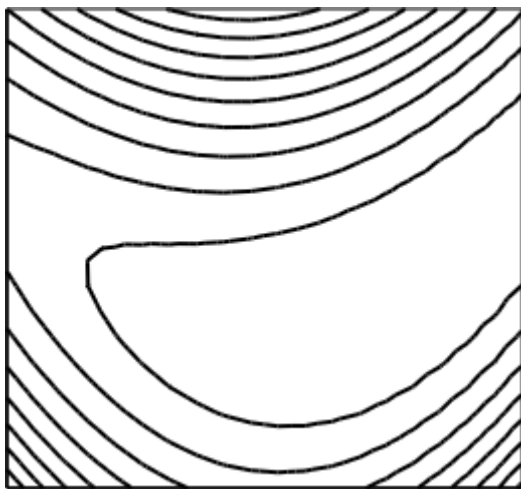
Методы одномерного поиска минимума, рассмотренные ранее, могут являться составными частями алгоритмов многомерной минимизации. Однако следует учитывать, что многомерное пространство качественно отличается от одномерного. Прежде всего, с увеличением числа измерений уменьшается вероятность унимодальности целевой функции $f(\vec{x})$, где $\vec{x} = (x_1, x_2, \dots, x_n)$, или $\vec{x} \in R^n$. Кроме того, объем вычислений, необходимых для сужения интервала неопределенности L_k в многомерном пространстве R^n , – степенная функция, показатель которой равен размерности пространства. Так, если в одномерном пространстве для уменьшения L_k от 1 до 0,1 требуется вычислить 19 значений целевой функции (по методу пассивного поиска), то в двумерном – это число составляет 361, трехмерном – 6859, четырехмерном – 130321 и т.д. При этом число проектных переменных при решении практических задач нередко достигает пяти и более.

Методы многомерной оптимизации подразделяют на три группы:

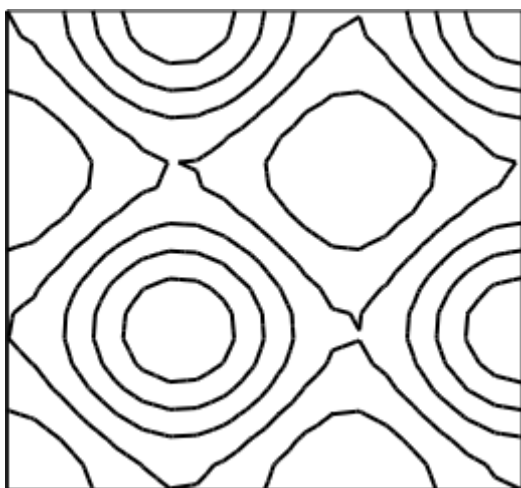
- 1) прямые (нулевого порядка) – используется информация только о значениях функции;
- 2) градиентные (первого порядка) – наряду со значениями функции вычисляется и ее градиент;
- 3) методы второго порядка – дополнительно требуется знание вторых производных.



а



б



в

Рис. 2.1. Типы рельефа:
 а – котловинный;
 б – овражный;
 в – неупорядоченный

Каждой сложной задаче проектирования присущи специфические особенности и нет ни одного универсального метода, позволяющего успешно решать все задачи. Выбирая алгоритм, следует проанализировать особенности поверхности, описываемой целевой функцией. Для простоты рассмотрим случай двух переменных. Функции $f(x, y)$ описывают некоторые поверхности в трехмерном пространстве. Основные типы рельефа функций (по виду линий уровня) показаны на рис. 2.1.

При котловинном рельефе (см. рис. 2.1, а) линии уровня похожи на эллипсы. В малой окрестности любого невырожденного минимума рельеф функции всегда котловинный. Этот вид рельефа наиболее благоприятен.

Овражный рельеф (см. рис. 2.1, б) более сложен для минимизации. Если линии уровня кусочно-гладкие, то геометрическое место точек излома образует дно оврага. Чаще всего линии уровня всюду гладкие, но на них имеются участки с большой кривизной. Сконструировано несколько функций с овражным типом рельефа, которые служат тестами для проверки качества методов минимизации. Например, функция Розенброка

$$f(x, y) = 100(y - x^2)^2 + (1 - x)^2, \\ \bar{x}^* = (1, 1).$$

Неупорядоченный тип рельефа (см. рис. 2.1, в) характеризуется наличием многих максимумов, минимумов и седловых точек.

Если известны топологические свойства исследуемой поверхности, то можно выбрать наиболее подходящий для нее алгоритм. Так, если поверхность имеет гладкие складки, не рекомендуется применять методы покоординатного спуска или градиентные. Для поверхностей с глубокими впадинами более эффективен прямой метод симплексов (см. ниже). Если поверхность мультимодальная, то нужно выбрать в пространстве проектирования несколько начальных точек и убедиться, что во всех случаях получается одно и то же решение. Анализ структуры алгоритма часто позволяет решить вопрос о его пригодности для решения конкретной задачи. Не рекомендуется использовать методы, требующие нахождения частных производных в аналитическом виде, так как для многих целевых функций это невозможно выполнить (например, для зависимостей, полученных экспериментально).

Метод покоординатного спуска – простейший из методов нулевого порядка. Идея метода состоит в следующем. Выбирается нулевое приближение \bar{x}^0 . Затем фиксируются значения всех координат x_i^0 , $i = 2, \dots, n$, кроме первой (x_1). Функция становится одномерной: $f(\bar{x}) = f_1(x_1)$; её минимум находят с помощью одного из методов одномерной минимизации (если для определения $\min f_1(x_1)$ используют метод парабол, то рассматриваемый метод, строго говоря, уже не нулевого порядка). В полученной точке \bar{x}^1 фиксируют значения всех координат, кроме x_2 , и находят минимум по этому направлению. После перебора всех координат один цикл заканчивается. Завершение одного цикла совсем не гарантирует попадания в точку минимума (рис. 2.2, а), поэтому циклы повторяют до сходимости. Этот метод сходится линейно, но не всегда к истинному минимуму. Если при овражном рельефе поиск приводит на дно оврага, то движение по любому координатному направлению не сможет вывести оттуда, так как будет соответствовать подъёму.

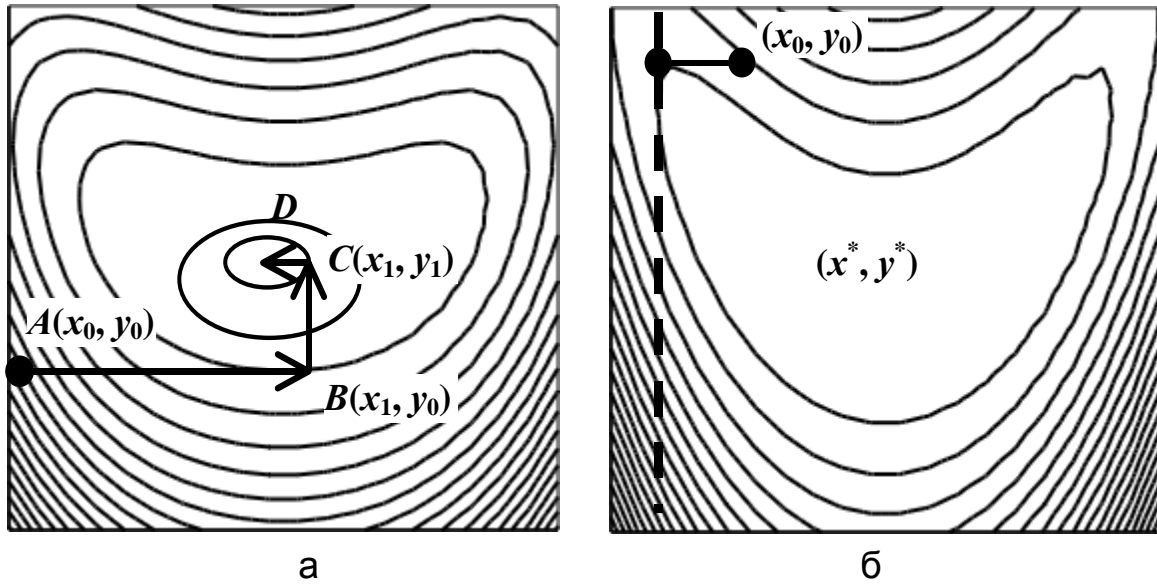


Рис. 2.2. Иллюстрация метода покоординатного спуска: нахождение минимума за два цикла (а); «зависание» метода (б)

Имеются разновидности этого метода, в которых поиск по каждому координатному направлению проводят не до точки минимума, а путём сдвижки на некоторый шаг h_j .

Алгоритм:

0. Выбираем начальное приближение \bar{x}^0 и шаги h_j вдоль каждого направления \bar{e}_j , где \bar{e}_j – единичные орты в направлении оси x_j . Вычисляем значение $f(\bar{x}^0)$.

1. Каждую переменную изменяем по очереди на величину шага. Вычисляем $f(\bar{x}^0 + h_1\bar{e}_1)$. Если это приводит к уменьшению функции, то переходим в эту точку. В противном случае вычисляем $f(\bar{x}^0 - h_1\bar{e}_1)$. Если это значение меньше $f(\bar{x}^0)$, то \bar{x}^0 заменяем на $\bar{x}^0 - h_1\bar{e}_1$. Иначе точка \bar{x}^0 остаётся неизменной. После этого рассматриваем изменения в направлении осей x_2, x_3, \dots, x_n . В результате получаем новую точку \bar{x}^1 .

2. Если $\bar{x}^1 \neq \bar{x}^0$, процедуру 1 повторяем. Если $\bar{x}^1 = \bar{x}^0$ (т.е. никуда не сдвинулись), то необходимо уменьшить длины шагов h_j и вновь перейти к этапу 1.

Это упрощённое описание алгоритма. Чтобы он работал более быстро и надёжно, необходимы различные дополнения к нему, как, например, в методе Хука - Дживса.

Метод Хука - Дживса (метод конфигураций) состоит из последовательности шагов исследующего поиска вокруг базисной точки, за которым в случае успеха следует поиск по образцу. Сначала выбирают исходную базовую точку пространства проектирования и величины шагов, которые будут использованы при исследовании функции. Затем выполняют исследование с заданными приращениями в направлениях, соответствующих всем независимым переменным. Туда, где получено уточненное значение функции (т.е. наименьшее из всех найденных значений), помещают новую временную базовую точку. После этого следует «поиск по образцу» – экстраполяция вдоль линии, соединяющей новую и прежнюю базовую точки, и исследование в окрестности найденной точки – точки образца. Если точка образца или одна из соседних с ней точек имеет преимущество перед другими, то процедуру повторяют с использованием лучшей точки в качестве базовой. Метод эффективен при отыскании экстремумов на гиперповерхностях, содержащих глубокие узкие впадины, т.е. когда градиентные методы «зависают».

Алгоритм метода Хука - Дживса:

0. Выбираем исходную базовую точку \bar{x}^0 и величины шагов h_i (они могут быть неодинаковыми для различных компонент \bar{x}).

1. Дискретно увеличиваем (уменьшаем) координаты x_i .

2. Новую временную базовую точку помещаем туда, где получено уточнённое значение функции.

3. После исследования по всем координатам выполняем "сдвиг схемы" (экстраполяцию вдоль линии, соединяющей новую и прежнюю базовые точки). В этом направлении вычисляем значения функции при постепенно увеличивающемся шаге (тем самым устанавливается конфигурация или тренд поиска):

$$x_{i,0}^{k+1} = x_i^{k+1} + \alpha(x_i^{k+1} - x_i^k),$$

где $\alpha \geq 1$ – коэффициент усиления, $x_{i,0}^{k+1}$ – новая временная базовая

точка или “точка роста”. Этап 3 продолжаем до тех пор, пока поиск в данном направлении приводит к точкам \vec{x} с меньшим значением функции $f(\vec{x})$. Если в направлении установленной конфигурации не удаётся найти точку с меньшим значением функции, то размер шага уменьшаем. После нескольких сокращений шага от принятой конфигурации отказываемся и предпринимаем новое исследование окрестности: не содержит ли она точку, приняв которую за следующую базовую можно приблизиться к оптимальному решению.

Если последовательно уменьшаемый шаг становится меньше некоторой заранее заданной величины ϵ и при этом отсутствует заметное изменение значения функции, поиск прекращается.

Достоинства метода Хука - Дживса:

1. Позволяет (поиском во время фазы исследования) восстанавливать направление движения вдоль оврага в тех случаях, когда вследствие искривления оврага установленная ранее конфигурация теряется.

2. Не требует знания целевой функции в явном виде (например, в технических задачах система может быть описана рядом связанных между собой уравнений подсистем; метод конфигураций требует лишь описания динамики подсистем и связей между ними).

3. Позволяет учитывать как ограничения, накладываемые на отдельные переменные, так и сложные ограничения, накладываемые на область поиска (например, изолированные запрещённые области).

4. Прост для программирования.

Недостаток метода конфигураций заключается в том, что он может “застревать”, т.е. остановиться вблизи локального минимума, особенно если поверхности постоянного значения целевой функции обладают острыми углами или очень изогнуты.

Симплекс-метод. Симплексом называется N -мерная замкнутая геометрическая фигура, ребра которой представляют собой прямые линии, пересекающиеся в $N + 1$ вершине. В двухмерном случае это треугольник, в трехмерном – тетраэдр. Схемы поиска с использованием симплексов основаны на слежении за изменением значений це-

левой функции в их вершинах. Главное в этих схемах – процесс отражения, т.е. нахождение вершины нового симплекса, расположенной симметрично относительно плоскости, проходящей через одну из сторон исходного симплекса. Выбор направления поиска вершины нового симплекса определяется положением той вершины исходного симплекса, в которой целевая функция имеет наихудшее значение. Новая точка называется *дополнением* наихудшей точки. Если в найденном дополнении значение целевой функции худшее, то алгоритм предусматривает возврат в исходную точку – вершину прежнего симплекса. Затем осуществляется переход к той вершине прежнего симплекса, где целевая функция имеет следующее по величине значение, и отыскивается точка, являющаяся ее дополнением. Поиск продолжается до тех пор, пока размеры симплекса по всем переменным не станут меньше заданной погрешности решения.

Такой алгоритм обеспечивает систематическое смещение центра симплекса в направлении экстремума целевой функции. Известен и более сложный метод – метод Нелдера - Мида, в котором помимо поиска вершин новых симплексов происходит сжатие или расширение их ребер.

Метод Нелдера - Мида. Один из наиболее быстрых и надежных неградиентных методов многомерной оптимизации. Идея метода состоит в следующем. В пространстве оптимизируемых параметров генерируется случайная точка и строится n -мерный симплекс с центром в этой точке и длиной стороны l . В каждой из вершин симплекса вычисляют значение оценки (целевой функции). Затем выбирают вершину с наибольшей оценкой, вычисляют центр тяжести остальных n вершин и оптимизируют шаг в направлении от наихудшей вершины к центру тяжести остальных вершин. Эту процедуру повторяют до тех пор, пока не окажется, что оптимизация не изменяет положение вершины. После этого выбирают вершину с наилучшей оценкой и вокруг нее снова строят симплекс с меньшими размерами (например $l/2$). Процедуру продолжают до тех пор, пока размер симплекса, который необходимо построить, не станет меньше требуемой точности.

Алгоритм метода Нелдера - Мида:

1. Выбирают начальную точку \bar{x}^0 и начальный симплекс. Вычисляют значения целевой функции в его вершинах. Из полученных значений выбирают наихудшее $P_{\text{худш}}$ и наилучшее $P_{\text{лучш}}$. Определяют положение центра симплекса \bar{P} , исключая наихудшую точку. Отраженную точку P^* находят по формуле

$$P^* = (1 + \alpha)\bar{P} - \alpha P_{\text{худш}},$$

где α – положительная постоянная, называемая *коэффициентом отражения*.

2. Если точке P^* соответствует лучшее из найденных значений целевой функции, то следующую точку определяют по выражению

$$P^{**} = \gamma P^* + (1 - \gamma)\bar{P},$$

где γ – заранее выбранная постоянная – *коэффициент расширения* ($\gamma > 1$).

3. Если значение целевой функции в точке P^{**} лучше ее значений во всех предыдущих точках, то его используют вместо прежнего наилучшего значения ($P_{\text{лучш}}$) и после проверки сходимости процесс повторяют. Если же P^{**} не лучше P^* , то $P_{\text{лучш}}$ заменяют на P^* и продолжают поиск.

4. Если ранее найденная точка P^* хуже $P_{\text{лучш}}$, но лучше $P_{\text{худш}}$, то точку $P_{\text{худш}}$ заменяют на P^* , в противном случае она остается неизменной. После этого по формуле

$$P^{**} = \beta P_{\text{худш}} + (1 - \beta)\bar{P}$$

определяют положение точки сжатия, где β – коэффициент сжатия ($1 > \beta > 0$).

5. Если точка P^{**} более удачная, чем $P_{\text{худш}}$, то в дальнейшем ею заменяют $P_{\text{худш}}$. В противном случае, прежде чем продолжить итерации, все P_i заменяют на новые:

$$P_{i \text{ нов}} = (P_{i \text{ ст}} + P_{\text{лучш}}) / 2.$$

Данный алгоритм обладает достаточной гибкостью, позволяющей учитывать топологические свойства поверхности, описываемой целевой функцией. Симплексы вытягиваются в направлении наклона плоскостей, их оси поворачиваются при встрече с впадиной на поверхности функции $f(\vec{x})$, вблизи экстремума они сжимаются.

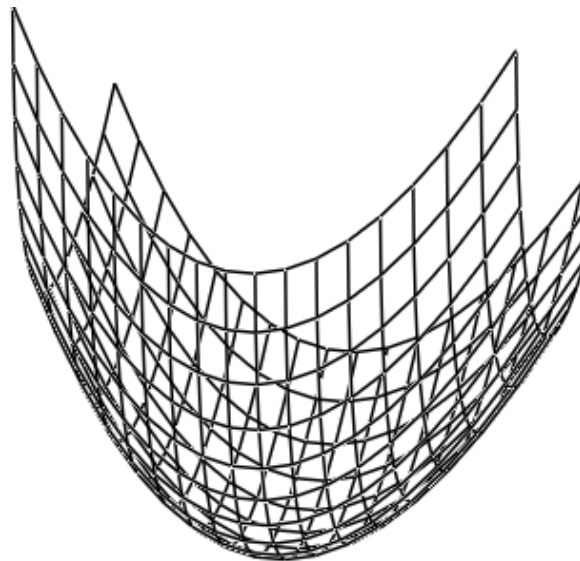


Рис. 2.3. Целевая функция

Процесс минимизации квадратичной целевой функции двух переменных $f(x, y) = (x - 1)^2 + y^2$ (рис. 2.3) по методу Нелдера - Мида показан на рис. 2.4.

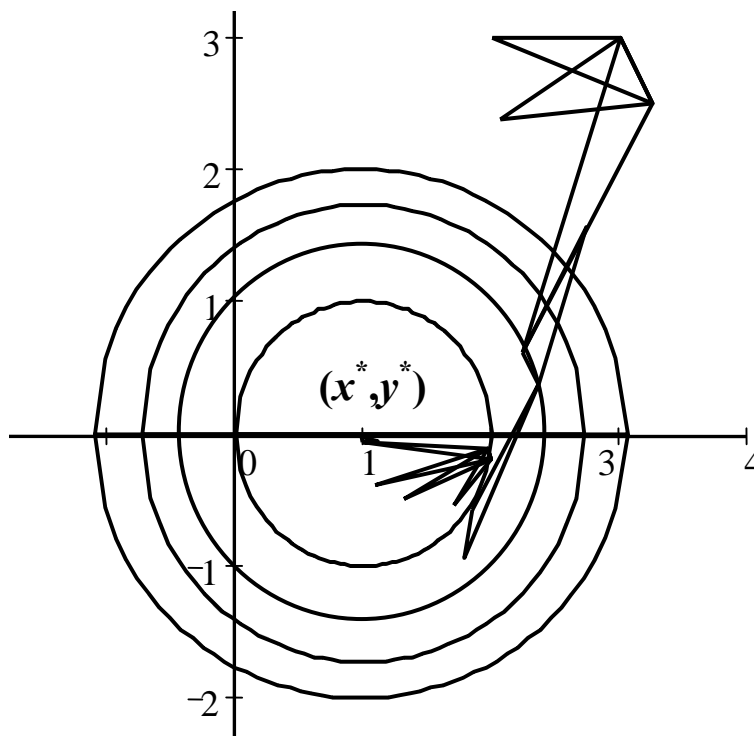


Рис. 2.4. Последовательность симплексов при поиске минимума квадратичной целевой функции $f(x, y)$

Градиентные методы – методы первого порядка, в которых для определения направления поиска используют градиент функции $\nabla f(\vec{x})$, $grad f$:

$$\nabla f(\vec{x}^0) = \vec{e}_i \sum_{i=1}^n \frac{\partial f(\vec{x})}{\partial x_i} \Big|_{\vec{x}^0},$$

где \vec{e}_i – единичные направляющие вектора вдоль осей координат.

Известно, что градиент указывает направление наискорейшего возрастания функции. Следовательно, противоположное направление – это направление наискорейшего убывания функции и именно вдоль вектора антиградиента $-\nabla f(\vec{x})$ ведется поиск минимума.

Все градиентные методы поиска экстремума строятся на двух свойствах градиента:

- так как в точке экстремума все производные равны нулю, то $\nabla f(\vec{x}^*) = \mathbf{0}$;

- градиент в любой точке \vec{x}^k всегда перпендикулярен линии уровня, проходящей через эту точку.

Следующую точку \vec{x}^{k+1} выбирают по отношению к предыдущей \vec{x}^k в направлении $-\nabla f(\vec{x}^k)$, т.е. $\vec{x}^{k+1} = \vec{x}^k - t\nabla f(\vec{x}^k)$, где t – положительная скалярная величина. В различных вариантах градиентного метода применяют различные способы выбора t .

Если t определяют путём решения одномерной задачи минимизации $\min f(\vec{x}^k - t\nabla f(\vec{x}^k))$, то вариант градиентного метода называется **методом наискорейшего спуска**. В качестве \vec{x}^{k+1} выбирают такую точку, расположенную в направлении $-\nabla f(\vec{x}^k)$, в которой целевая функция принимает минимальное значение. Затем находят значение вектора-градиента в точке \vec{x}^{k+1} и повторяют процедуру.

Рассмотрим задачу минимизации квадратичной целевой функции. Из геометрической интерпретации (рис. 2.5) видно, что луч $\vec{x} - t\nabla f(\vec{x})$, выходящий из исходной точки в направлении антиградиента, не приводит к минимизации за один шаг.

Приближение к точке минимума происходит зигзагообразно, взаимно перпендикулярными шагами. Каждый спуск кончается в точке, где его направление касается линии (поверхности) уровня, поскольку необходимое условие экстремума $f'(t) = 0$ одномерной функции совпадает с уравнением касательной к линии уровня. Следующий спуск идёт по направлению антиградиента к линии уровня в данной точке, т.е. перпендикулярно предыдущему направлению. В двухмерном случае это означает, что метод наискорейшего спуска совпадает с методом покоординатного спуска, если систему координат повернуть так, чтобы ось $0x'$ стала параллельна $\nabla f(\bar{x})$ в начальной точке.

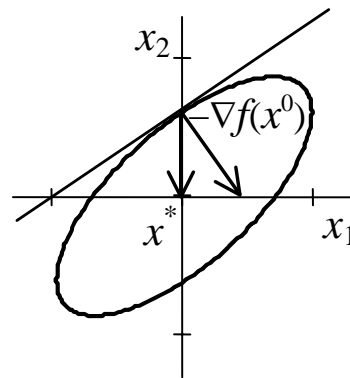


Рис. 2.5. Выбор направления поиска

Метод Ньютона – метод второго порядка, является прямым обобщением метода отыскания корня уравнения $\phi(t) = 0$, где $\phi(t)$ – функция скалярной переменной (метод касательных). При решении n -мерной задачи минимизации требуется определить корни уравнения $\nabla f(\bar{x}) = 0$. Разложим это уравнение в ряд Тейлора:

$$0 = \nabla f(\bar{x}^{(k+1)}) \approx \nabla f(\bar{x}^{(k)}) + \mathbf{H}(\bar{x}^{(k)})(\bar{x}^{(k+1)} - \bar{x}^{(k)}),$$

где \mathbf{H} – матрица Гессе:

$$\mathbf{H}(\bar{x}^{(k)}) = \left[\left. \frac{\partial^2 f(\bar{x})}{\partial x_i \partial x_j} \right|_{\bar{x}^{(k)}} \right], \quad i, j = 1, \dots, n.$$

Последовательность $\{\bar{x}^{(k)}\}$ рассчитывают по формуле

$$\bar{x}^{(k+1)} = \bar{x}^{(k)} - \mathbf{H}^{-1}(\bar{x}^{(k)}) \nabla f(\bar{x}^{(k)})$$

при условии, что существует обратная матрица Гессе \mathbf{H}^{-1} .

Недостаток метода Ньютона – его трудоемкость, связанная с необходимостью вычисления вторых производных целевой функции и с обращением $(n \times n)$ матрицы Гессе. Метод может расходиться, если целевая функция не является строго выпуклой и начальное приближение достаточно далеко отстоит от минимума.

Постановка задачи

Разработать и реализовать в среде MathCAD алгоритмы минимизации функции двух переменных (согласно заданному варианту):

- методом наискорейшего спуска;
- методом покоординатного спуска.

Сравнить эффективность работы данных алгоритмов. Критерии эффективности выбрать самостоятельно.

Точность вычислений для всех вариантов $\varepsilon \leq 0,01$.

Контрольные вопросы

1. Назовите типы рельефа функций. Как влияет топология поверхности целевой функции на выбор алгоритма оптимизации?
2. Напишите общую формулу выбора очередного приближения.
3. Как определяется длина шага и направление поиска в изученных методах многомерной минимизации?
4. В каких случаях приемлемые направления поиска являются недопустимыми?
5. Перечислите достоинства и недостатки известных методов многомерной минимизации.
6. Как проверить, найден ли глобальный минимум целевой функции или же алгоритм “завис” вблизи локального минимума?

Варианты заданий

Номер варианта	$f(x, y)$	Номер варианта	$f(x, y)$
1	$x^2 + 4x + y^2 + 2y + 1$	5	$x^2 + 6x + y^2 - 4y + 5$
2	$x^2 - 4x + y^2 + 2y + 2$	6	$x^2 + 6x + y^2 - 2y + 6$
3	$x^2 - 2x + y^2 + 4y + 3$	7	$x^2 - 4x + y^2 + 6y + 7$
4	$x^2 - 2x + y^2 - 4y + 4$	8	$x^2 - 8x + y^2 - 6y + 8$

ОПТИМИЗАЦИЯ ПАРАМЕТРОВ РАДИОЭЛЕКТРОННЫХ УСТРОЙСТВ

Цель работы:

- изучить оптимизационные методы поиска экстремума с учетом ограничений, налагаемых на проектные параметры;
- научиться формировать простые и составные целевые функции;
- получить практические навыки реализации оптимизационных алгоритмов в системе MathCAD;
- определить оптимальное значение проектного параметра по критерию соответствия требуемой амплитудно-частотной характеристике в заданном диапазоне частот.

Теоретические сведения

В задачах синтеза пассивных цепей величины отдельных параметров L , R и C ограничены: $g_i(R, C, L) \leq 0$. Например, ограничение $1 \leq RC \leq 100$ можно записать в виде $(100 - RC)(1 - RC) \leq 0$. Таким образом, задача оптимизации формулируется как оптимизация заданной целевой функции (в зависимости от выбранного критерия качества) при условии выполнения ряда заданных ограничений.

Рассмотрим задачу построения системы, обладающей заданной выходной реакцией (например, частотная характеристика). Требуемая частотная характеристика $F_d(\omega)$ должна быть задана либо аналитически, либо в виде таблицы. Фактическую частотную характеристику при значениях параметров L , R , C вычисляют как функцию $F(\omega; L, R, C)$ при дискретных значениях частоты ω_i , $i = 1, \dots, m$. Как выразить математически, насколько эти две кривые (F_d и F) близки друг к другу? Как и при измерении расстояния между двумя кривыми, можно ввести различные меры близости двух функций:

$$1) d_p(F_d, F) = \left[\sum_{i=1}^m (F_{di} - F_i)^p \right]^{1/p}, \text{ где } p - \text{положительное целое}$$

число;

$$2) d_\infty(F_d, F) = \max_{1 \leq i \leq m} |F_{di} - F_i| - \text{это } \lim d_p \text{ при } p \rightarrow \infty.$$

При малых значениях p две кривые отделены друг от друга большим расстоянием в том случае, если они различаются на большом интервале ω (рис. 3.1, кривые 1, 2). Когда p увеличивается, расстояние между двумя кривыми всё в большей мере будет определяться максимальным расхождением двух характеристик (рис. 3.1, кривые 1, 3).

Таким образом, простая целевая функция для рассматриваемой задачи имеет вид

$$f(L, R, C) = \sum_{i=1}^m \phi[F_d(\omega_i), F(\omega_i; L, R, C)],$$

она указывает, в какой степени расчётные величины обеспечивают получение требуемой частотной характеристики. Можно задать также другие величины (критерии качества проектируемой системы): потери мощности, величины постоянных напряжений и т.п. Однако такая функция не учитывает ограничения, налагаемые на проектные параметры, поэтому может быть найдено недопустимое решение.

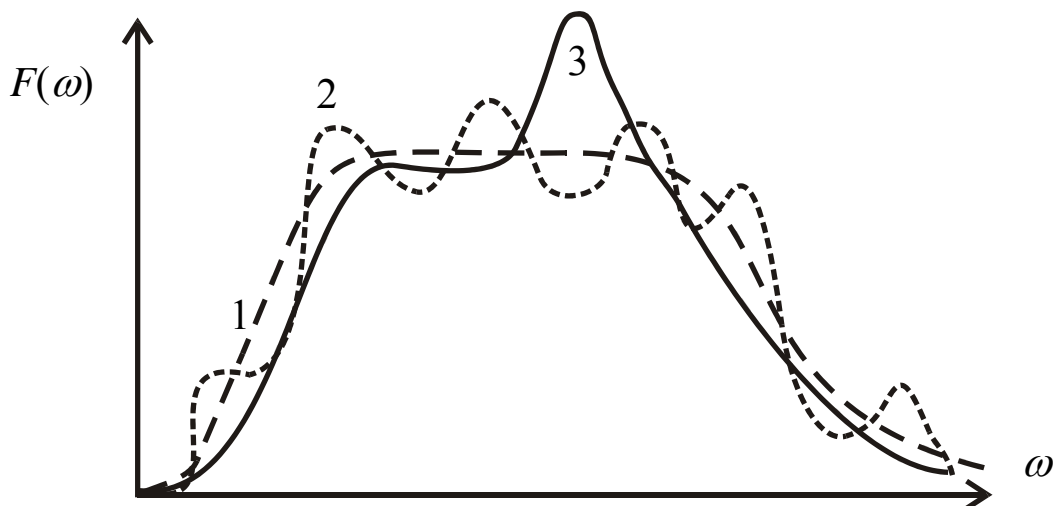


Рис. 3.1. К определению мер близости функций:
1 – требуемая характеристика,
2, 3 – различные фактические кривые

Рассмотрим численные методы отыскания экстремума нелинейной функции при нелинейных ограничениях:

$$\begin{cases} \min f(\bar{x}), & \bar{x} = (x_1, \dots, x_n); \\ \varphi_i(\bar{x}) = 0, & i = 1 \dots m; \\ \psi_j(\bar{x}) \leq 0, & j = 1 \dots p. \end{cases} \quad (3.1)$$

Условия типа равенств выделяют в пространстве \mathbf{R}^n некоторую $(n-m)$ -мерную поверхность, поэтому должно выполняться условие $m < n$. Условия типа неравенств выделяют n -мерную область, ограниченную гиперповерхностями $\psi_j(\bar{x}) = 0$, поэтому число этих условий может быть любым. Таким образом, задача (3.1) – это задача отыскания экстремума в $(n-m)$ -мерной области, причём в общем случае минимум может реализовываться как внутри области, так и на её границе. Как правило, вид дополнительных условий достаточно сложен, поэтому исключить их и вести поиск прямо в $(n-m)$ -мерной области не удаётся, чаще всего приходится вести поиск во всём n -мерном пространстве и при этом учитывать ограничения.

В численных методах для нелинейных задач с ограничениями есть два подхода: первый – состоит в том, чтобы, непосредственно контролируя соблюдение ограничений задачи, двигаться к её оптимуму по последовательности допустимых или “почти допустимых” точек. Это *методы спуска* (или градиентные методы). Вторым подходом является сведение задачи на условный экстремум к последовательности задач безусловной оптимизации для специальных вспомогательных функций. Это *методы штрафных функций*, они отличаются простотой реализации и сильными свойствами сходимости.

Рассмотрим задачу поиска минимума $f(\bar{x})$ на некотором ограниченном множестве $X \in \mathbf{R}^n$. Формально она эквивалентна задаче безусловной минимизации $\tilde{f}(\bar{x}) = f(\bar{x}) + \delta(\bar{x}|X)$, где $\delta(\bar{x}|X)$ – так называемая индикаторная функция

$$\delta(\bar{x}|X) = \begin{cases} 0, & \bar{x} \in X; \\ +\infty, & \bar{x} \notin X. \end{cases} \quad (3.2)$$

Если $\bar{x} \in X$, то $\tilde{f}(\bar{x}) \equiv f(\bar{x})$, если $\bar{x} \notin X$, то при таких \bar{x} $f(\bar{x})$ не может иметь минимум; следовательно, $\min_{\bar{x} \in X} \tilde{f}(\bar{x}) \equiv \min_{\bar{x} \in X} f(\bar{x})$.

Определение (3.2) функции $\delta(\bar{x}|X)$ не является конструктивным. Однако, если X задано ограничениями типа равенств или неравенств, можно построить конкретные штрафные функции $\delta_k(\bar{x}|X)$, такие, что $\forall \bar{x} \in \mathbf{R}^n \quad \lim_{k \rightarrow \infty} \delta_k(\bar{x}|X) = \delta(\bar{x}|X)$.

Тогда $\min_{\bar{x} \in X} f(\bar{x}) = \lim_{k \rightarrow \infty} \min_{\bar{x} \in X} \{f(\bar{x}) + \delta_k(\bar{x}|X)\}$.

Существует два типа методов штрафных функций – внутренние (барьерные) и внешние.

Методы барьерных функций. Если допустимое множество X задано только ограничениями-неравенствами $\psi_j(\bar{x}) \leq 0$, причём существуют точки, в которых все $\psi_j(\bar{x}) < 0$, то в качестве внутренних штрафных функций применяют функции вида

$$\delta_k(\bar{x}|X) = -r_k \sum_{j=1}^p \frac{1}{\psi_j(\bar{x})}$$

или

$$\delta_k(\bar{x}|X) = -r_k \sum_{j=1}^p \max\{-\ln[-\psi_j(\bar{x})]; 0\},$$

где r_k – параметры штрафа – положительные числа, такие, что,

$\lim_{k \rightarrow \infty} r_k = 0$, например: $r_k = k^{-1}$, $r_k = 10^{-k}$.

Алгоритм

1. Выбирают внутреннюю точку \bar{x}^0 , в которой $\psi_j(\bar{x}^0) < 0$, и последовательность коэффициентов $r_k \rightarrow 0$.

2. При $k = 1, 2, \dots$ начиная с \bar{x}^{k-1} решают любым допустимым способом задачу безусловной минимизации функции

$$\tilde{f}(\bar{x}) = f(\bar{x}) + \delta_k(\bar{x}|X).$$

В результате находят очередное приближение \bar{x}^k . Предел $\lim_{k \rightarrow \infty} \bar{x}^k = \bar{x}^*$ совпадает с точкой минимума исходной функции, причём все приближения \bar{x}^k будут находиться строго внутри X .

Методы внешних штрафных функций применимы к более широкому классу задач с ограничениями как в форме неравенств, так и равенств.

Штрафы $\delta_k(\bar{x}|X)$, сходящиеся при $k \rightarrow \infty$ к индикаторной функции, строят так, чтобы $\forall k$ выполнялись условия: $\delta_k(\bar{x}|X) = 0$ для $\bar{x} \in X$ и $\delta_k(\bar{x}|X) > 0$ для $\bar{x} \notin X$.

Обычно полагают $\delta_k(\bar{x}|X) = r_k \Phi(\bar{x})$, $r_k \rightarrow \infty$ при $k \rightarrow \infty$.

В качестве $\Phi(\bar{x})$ берут функции вида

$$\Phi(\bar{x}) = \sum_{j=1}^p [\psi_j^+(\bar{x})]^\xi + \sum_{i=1}^m |\varphi_i(\bar{x})|^\xi,$$

где ξ – фиксированное натуральное число (обычно 1 или 2), $\psi_j^+(\bar{x})$ – “срезка” функции $\psi_j(\bar{x})$, т.е. $\psi_j^+(\bar{x}) = \max\{\psi_j(\bar{x}); 0\}$.

Алгоритм решения задач с использованием метода внешних штрафных функций аналогичен алгоритму метода барьерных функций, за исключением того, что движение происходит извне, т.е. из-за границ допустимой области. Это различие показано на рис. 3.2. Для примера рассмотрим задачу минимизации функции $f(x) = x$ при $x \geq 0$. Очевидно, что минимум этой функции достигается на границе допустимого множества $x = 0$. В качестве ограничителя области выступает функция $\psi_1(x) = -x$. По методу барьерных функций выберем

$r_k = k^{-1}$, $\delta_k(x|X) = -r_k \frac{1}{\psi_1(x)}$. Тогда $\tilde{f}_k(x) = x + \frac{1}{kx}$. Точку минимума

определим из условия $\tilde{f}'_k(x) = 0$, $x^k = 1/\sqrt{k} \rightarrow 0$ при $k \rightarrow \infty$.

В методе штрафных функций положим $r_k = k$, $\xi = 2$. Тогда $\tilde{f}_k(x) = x + k[\max\{-x; 0\}]^2$, $\tilde{f}'_k(x) = 1 - 2k[\max\{-x; 0\}] = 0$ и $x^k = -1/2k$.

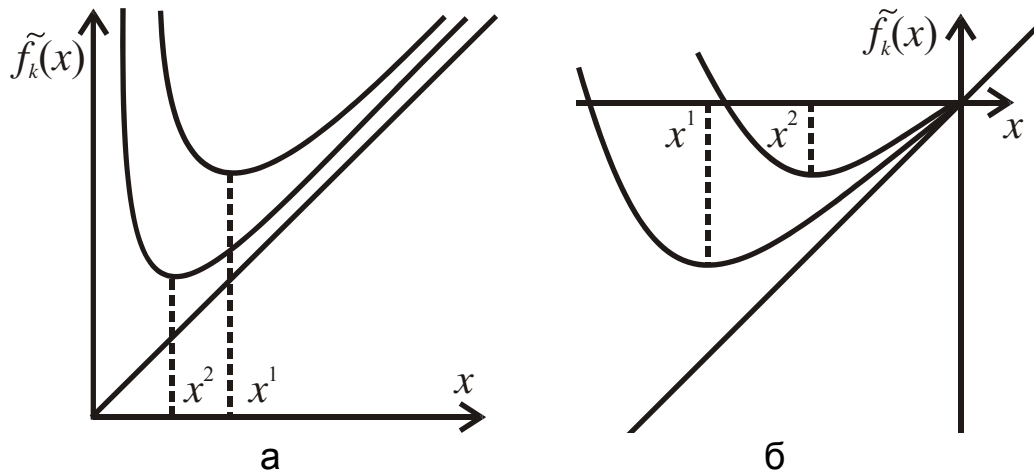


Рис. 3.2. Примеры последовательностей вспомогательных функций для методов внутренних (а) и внешних (б) штрафных функций

Полученная на k -й итерации точка экстремума может лежать за пределами X . Поэтому после завершения счета необходимо проверить допустимость решения $\bar{x}^k \in X$ и, если $\bar{x}^k \notin X$, скорректировать полученную точку, например, спроектировать её на границу X .

Таким образом, в методе штрафных функций ограниченную функцию $f(\bar{x})$ заменяют составной неограниченной целевой функцией $\tilde{f}(\bar{x})$, которая образуется путем сложения целевой функции рассматриваемой задачи и штрафной функции, учитывающей заданные ограничения. Если все ограничения удовлетворяются, то функции $\tilde{f}(\bar{x})$ и $f(\bar{x})$ имеют один и тот же минимум. Если хотя бы одно условие не удовлетворяется, то составная целевая функция $\tilde{f}(\bar{x})$ приобретает бесконечно большие значения, весьма далекие от минимального значения $f(\bar{x})$. Тем самым на каждое решение, не удовлетворяющее поставленным условиям, «налагается штраф».

Простыми примерами составных целевых функций являются функции вида

$$\tilde{f}(\bar{x}) = f(\bar{x}) + W \sum_{j=1}^p A_j |\psi_j(\bar{x})|;$$

$$\tilde{f}(\bar{x}) = f(\bar{x}) + W \sum_{j=1}^p A_j [\psi_j(\bar{x})]^2,$$

где W – большое число, например 10^{20} ; $A_j = \begin{cases} 1 & \text{при } \psi_j(\vec{x}) > 0, \\ 0 & \text{при } \psi_j(\vec{x}) \leq 0. \end{cases}$

Постановка задачи

1. Объект исследования задан передаточной функцией $K(p)$ (варианты согласно табл. 3.1); p – оператор Лапласа:

$$p = i2\pi f.$$

2. Выходной параметр объекта – значение коэффициента усиления. Так как объект динамический, то значение выходного параметра зависит от частоты f входного сигнала и представляет собой амплитудно-частотную характеристику (АЧХ). Дискретные значения требуемой АЧХ приведены в табл. 3.2.

В прил. 2 представлены примеры программ MathCAD для оптимизации RC -цепочки по критерию соответствия требуемой АЧХ в диапазоне частот от 0 до 1 кГц. Данные программы реализуют алгоритм *метода Хука - Дживса* для безусловной минимизации функций одной и двух переменных.

Исходные данные для проектирования RC -цепочки:

- требуемые значения АЧХ (при частоте $f_i = 0, 100, \dots, 1000$ Гц) заданы в виде массива данных $AK_i, i = 0, \dots, 10$;
- фактическая АЧХ

$$K(f, R, C) = \left| \frac{1}{1 + i2\pi fRC} \right|,$$

для случая оптимизации по одному параметру (R) значение емкости принято равным $C = 10^{-6}$ Ф.

Предварительно должны быть выбраны базисная точка (Bx) и шаг изменения H для проектного параметра (параметров).

Процесс поиска завершается, когда будет достигнута требуемая точность вычисления минимума целевой функции (или когда длина шага будет уменьшена до заданного малого значения).

Порядок выполнения

1. Для заданного варианта устройства сформировать целевую функцию как функцию значений проектных параметров по критерию совпадения расчетной АЧХ ($K0_j, j = 0, \dots, 100$) с требуемой формой частотной характеристики (массив $AK_i, i = 0, \dots, 10$). Критерий совпадения кривых требует минимального отклонения заданной кривой от фактической. Целевую функцию в этом случае можно определить по формуле наименьших квадратов:

$$TF = \sum_j (AK_j - K0_j)^2. \quad (3.3)$$

Задача вычисления целевой функции включает в себя вопросы аппроксимации: чем адекватнее аппроксимационная кривая АЧХ, построенная на узловых точках AK_i , тем больше вероятность адекватной оценки отклонения фактической АЧХ от требуемой. Промежуточные точки таблично заданной функции рассчитывают по интерполяционной формуле Лагранжа

$$P_n(x) = \sum_{k=0}^n \frac{(x-x_0)(x-x_1)\dots(x-x_{k-1})(x-x_{k+1})(x-x_n)}{(x_k-x_0)(x_k-x_1)\dots(x_k-x_{k-1})(x_k-x_{k+1})(x_k-x_n)} f(x_k).$$

2. По методу штрафных функций сформировать составную целевую функцию, учитывающую ограничения на множество допустимых значений проектных параметров.

3. Найти значение проектных параметров, при котором значение составной целевой функции минимально, используя различные оптимизационные алгоритмы.

Контрольные вопросы

1. Как формируют и аппроксимируют целевые функции?
2. Изложите идею методов условной оптимизации.
3. В чем сходство и различие между штрафными и барьерными функциями?
4. Как выбрать наиболее эффективный алгоритм для оптимизации при наличии ограничений на проектные параметры?

Варианты заданий

Таблица 3.1

Номер варианта	Передаточная функция объекта	Параметры
1	$K(p) = \frac{pR_1R_2C_1 + R_2}{pR_1R_2C_1 + R_1 + R_2}$	$R_1 = 10^4, R_2 = 10^3, C_1 = x$
2	$K(p) = \frac{pR_2C_1}{p(R_1 + R_2)C_1 + 1}$	$R_1 = 10, R_2 = 1000, C_1 = x$
3	$K(p) = \frac{pR_2C_1 + 1}{p(R_1 + R_2)C_1 + 1}$	$R_1 = 10^4, R_2 = 10^3, C_1 = x$
4	$K(p) = \frac{1}{p^2C_1L_1 + pR_1C_1 + 1}$	$R_1 = 1, C_1 = 10^{-4}, L_1 = x$
5	$K(p) = \frac{p^2C_1L_1 + pR_1C_1}{p^2C_1L_1 + pR_1C_1 + 1}$	$R_1 = 5, C_1 = 10^{-4}, L_1 = x$
6	$K(p) = \frac{pL_1 + R_2}{pL_1 + R_1 + R_2}$	$R_1 = 10, R_2 = 2, L_1 = x$
7	$K(p) = \frac{R_2}{pL_1 + R_1 + R_2}$	$R_1 = 1, R_2 = 10, L_1 = x$
8	$K(p) = \frac{pR_1C_1}{pR_1(C_1 + C_2) + 1}$	$R_1 = 2 \cdot 10^3, C_1 = 10^{-7}, C_2 = x$

Таблица 3.2

Номер варианта	Значения АЧХ на частотах f_i				
	0 Гц	250 Гц	500 Гц	750 Гц	1000 Гц
1	0,09	0,97	0,99	1,00	1,00
2	0,00	0,84	0,94	0,97	0,98
3	1,00	0,15	0,11	0,10	0,09
4	1,00	5,21	0,29	0,11	0,06
5	0,00	1,82	1,19	1,08	1,04
6	0,17	0,98	0,99	1,00	1,00
7	0,91	0,52	0,30	0,21	0,16
8	0,00	0,94	0,98	0,99	1,00

МЕТОД ВЕТВЕЙ И ГРАНИЦ

Цель работы:

- ознакомиться со спецификой численных методов дискретной оптимизации и структурой метода ветвей и границ;
- получить решение типовой задачи о коммивояжере с помощью метода ветвей и границ.

Теоретические сведения

Задача

$$f(\vec{x}) \rightarrow \min (\max), \vec{x} \in X \quad (4.1)$$

называется задачей дискретной оптимизации, если либо само допустимое множество $X \in R^n$ дискретно (т.е. является конечным или счетным множеством без предельных точек), либо только некоторые из координат вектора $\vec{x} = (x_1, \dots, x_n)$ принимают дискретные значения на числовой оси. Часто допустимое множество задачи дискретной оптимизации имеет вид

$$X = \{ \vec{x} \in D \mid \varphi_i(\vec{x}) = 0, i = 1, \dots, k; \psi_i(\vec{x}) \leq 0, i = k + 1, \dots, m \},$$

где $D = D_1 \times D_2 \times \dots \times D_n$, причем $D_j \subset \{0, \pm 1, \pm 2, \dots\} \quad \forall j \in J$ и $D_j = R \quad \forall j \notin J$, J – некоторое подмножество множества $\{1, \dots, n\}$.

Например $\forall j \in J: D_j = \{0, \pm 1, \pm 2, \dots\}$, $D_j = \{0, 1, 2, \dots\}$, $D_j = \{0, 1\}$.

Это означает, что координата x_j может принимать только целые, натуральные или булевы значения.

Задачу (4.1) с указанным множеством X называют задачей дискретного программирования. Если целевая функция $f(\vec{x})$ и функции $\varphi_i(\vec{x})$, $\psi_j(\vec{x})$, задающие ограничения, – линейны, то задачу (4.1) при $J = \{1, \dots, n\}$ называют целочисленной задачей линейного программирования (при $J \neq \{1, \dots, n\}$ – частично целочисленной). Если

$D_j = \{0, 1\}$, то это (частично) булева задача линейного программирования. С задачами дискретной оптимизации часто сталкиваются во всех сферах практической деятельности.

Пример 1. Выбор оптимальной структуры автоматической системы управления (АСУ). Пусть АСУ состоит из n звеньев и предназначена для решения m задач; a_{ij} – затраты ресурсов на выполнение i -й задачи в j -м звене; b_j – ресурсы j -го звена. Структура АСУ задается матрицей $\{x_{ij}\}$, $i = 1, \dots, m$, $j = 1, \dots, n$, где $x_{ij} = 1$, если i -я задача выполняется в j -м звене, и $x_{ij} = 0$ – в противном случае.

Необходимо выбрать структуру АСУ, для которой совокупные затраты минимальны. Это приводит к булевой задаче линейного программирования

$$\sum_{i=1}^m \sum_{j=1}^n a_{ij} x_{ij} \rightarrow \min, \quad \sum_{i=1}^m a_{ij} x_{ij} \leq b_j, \quad \sum_{j=1}^n x_{ij} = 1, \quad x_{ij} \in \{0, 1\}.$$

Целевая функция описывает суммарные затраты. Ограничения-неравенства задают границы затрат (затраты в каждом j -м звене не могут превосходить имеющихся ресурсов). Ограничения-равенства означают, что АСУ должна решать все m задач.

Пример 2. Задача о коммивояжере. Имеется n городов, пронумерованных числами $1, \dots, n$. Для любой пары городов (i, j) задано расстояние $c_{ij} \geq 0$ между ними (c_{ij} может означать не только расстояние, но и время, путевые расходы и пр., поэтому в общем случае не предполагается, что $c_{ij} = c_{ji}$). Выехав из исходного города, коммивояжер должен вернуться в него, побывав во всех остальных городах ровно один раз. В качестве исходного может быть выбран любой город. Требуется найти маршрут минимальной длины, т.е. минимизировать функцию $f(x) = c_{i_1 i_2} + c_{i_2 i_3} + \dots + c_{i_n i_1}$ на множестве X допустимых маршрутов $x = (i_1, \dots, i_n, i_1)$, где (i_1, \dots, i_n) – произвольная перестановка чисел $1, \dots, n$. Множество X содержит $n!$ элементов, поэтому решить задачу методом полного перебора при достаточно больших n нереально.

Решение задач дискретной оптимизации связано с трудностями принципиального характера. Полный перебор точек допустимого множества, как правило, неосуществим из-за слишком большого объема вычислительной работы. Из-за дискретности допустимого множества неприменимы многие приемы, разработанные в математическом программировании, например, движение по направлению антиградиента, переход из одной вершины многогранного множества допустимых точек в другую и т.д. Невозможно использовать здесь и теорию условий оптимальности. Поэтому для решения задач дискретной оптимизации разработаны специальные методы, к числу которых относится **метод ветвей и границ**. Различные варианты этого метода используют специфику конкретных задач и заметно отличаются друг от друга. Однако все они основаны на последовательном разбиении допустимого множества на подмножества (ветвлении) и вычислении оценок (границ), позволяющем отбрасывать подмножества, заведомо не содержащие решений задачи.

Допустим, необходимо решить задачу отыскания минимума функции $f(\bar{x})$, $\bar{x} \in X$. В зависимости от специфики задачи выбирают конкретный способ вычисления **оценок снизу** $d(X')$ целевой функции на множествах $X' \subset X$ (в частности, может быть $X' = X$):

$$f(\bar{x}) \geq d(X'), \bar{x} \in X'.$$

Оценку снизу часто вычисляют путем *релаксации*, т.е. замены задачи минимизации целевой функции по множеству X' задачей минимизации по некоторому более широкому множеству. Например, релаксация целочисленной или частично целочисленной задачи может состоять в отбрасывании требования целочисленности.

Также надо определить **правило ветвления**, состоящее в *выборе разветвляемого подмножества* X' из числа подмножеств, на которые к данному шагу разбито X , и *выборе способа разбиения* X' на непересекающиеся подмножества.

Обычно из числа кандидатов на ветвление выбирают множество X' с наименьшей оценкой, так как именно в нем естественно искать минимум в первую очередь. При этом рассматривают только такие способы получения оценок снизу, в которых оценки

для подмножеств, образовавшихся после разветвления X' , не меньше $d(X')$.

При решении релаксированной задачи может оказаться, что:

1) допустимое множество пусто и $X' = \emptyset$;

2) значение $d(X')$ минимума больше или равно наименьшему из вычисленных значений целевой функции (текущему значению рекорда) и, следовательно, минимум функции $f(\bar{x})$, $\bar{x} \in X$ достигается вне множества X' ;

3) найденная точка минимума принадлежит множеству X' и, следовательно, является точкой минимума целевой функции на X' .

Во всех трех случаях X' исключается из числа кандидатов на дальнейшее ветвление. В третьем случае текущее значение рекорда полагается равным минимуму из предыдущего текущего значения и вычисленного значения $\min_{\bar{x} \in X'} f(\bar{x})$.

С помощью метода ветвей и границ можно решать задачу приближенно. При этом на каждом шаге искомое значение минимума не больше текущего значения рекорда и не меньше наименьшей из оценок снизу для подзадач, входящих на данном шаге в число кандидатов на ветвление.

Решение задачи о коммивояжере методом ветвей и границ

Задача состоит в минимизации функции

$$f(x) = c_{i_1 i_2} + c_{i_2 i_3} + \dots + c_{i_n i_1} \quad (4.2)$$

на множестве X допустимых маршрутов $x = (i_1, \dots, i_n, i_1)$, где (i_1, \dots, i_n) – произвольная перестановка чисел $1, \dots, n$. Далее допустимые маршруты будем называть *маршрутами*, а переезды коммивояжера $(i_1, i_2), (i_2, i_3), \dots, (i_n, i_1)$ – *дугами маршрута*. Матрицу расстояний обозначим через $c = \{c_{ij}\}$, $i, j = 1, \dots, n$, $c_{ij} \geq 0$. Чтобы запретить переезды вида (i, i) , положим $c_{ii} = +\infty$, $i = 1, \dots, n$.

Если считать города вершинами графа, а переезды (i, j) – его ребрами, то требование найти минимальный путь, проходящий только один раз через каждый город и приводящий обратно в исходный пункт, можно рассматривать как задачу нахождения гамильтонова контура минимальной длины.

Для практической реализации идеи метода ветвей и границ применительно к задаче коммивояжера нужно найти способ определения нижних границ подмножества и разбиения множества гамильтоновых контуров на подмножества (ветвления).

Операция приведения. Введем следующие обозначения: $g_i = \min_{j=1 \div n} c_{ij}$, $h_j = \min_{i=1 \div n} (c_{ij} - g_i)$, $\bar{c}_{ij} = c_{ij} - g_i - h_j$. Матрица $\bar{c} = \{\bar{c}_{ij}\}$ называется *приведенной*, операция ее построения – *приведением* матрицы c , а величины g_i , h_j – *константами приведения*.

Приведенную матрицу \bar{c} получают путем вычитания из всех элементов каждой строки исходной матрицы c минимального элемента этой строки и из всех элементов получившейся матрицы – минимального элемента этого столбца. Очевидно, что $\bar{c}_{ij} \geq 0$ и в каждой строке и каждом столбце \bar{c} имеется хотя бы один нулевой элемент.

Положим

$$d(X) = \sum_{i=1}^n g_i + \sum_{j=1}^n h_j.$$

Для любого маршрута $x = (i_1, \dots, i_n, i_1) \in X$

$$f(x) = c_{i_1 i_2} + c_{i_2 i_3} + \dots + c_{i_n i_1} = \bar{c}_{i_1 i_2} + \dots + \bar{c}_{i_n i_1} + d(X) \geq d(X), \quad (4.3)$$

так как в сумме (4.2) содержится ровно по одному элементу из каждой строки и из каждого столбца матрицы c .

Неравенство (4.3) показывает, что $d(X)$ есть оценка снизу для X . Кроме того, после операции приведения длины всех маршрутов уменьшаются на одну и ту же величину $d(X)$. Поэтому оптимальный маршрут, найденный с использованием приведенной матрицы, оптимален и для исходной задачи.

Ветвление можно представить в виде дерева (рис. 4.1), каждая вершина которого соответствует некоторому множеству маршрутов, являющемуся подмножеством множества X . Начальная вершина соответствует множеству всех маршрутов X . На каждом шаге из числа кандидатов на ветвление выбирают множество X' с минимальной оценкой, которое разветвляется на два подмножества X'_1 и X'_2 и исключается из числа кандидатов на ветвление.

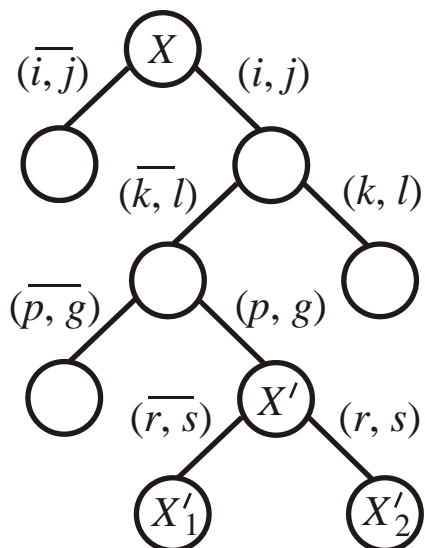


Рис. 4.1. Граф маршрутов

Подмножество X'_1 состоит из всех маршрутов множества X' , не содержащих некоторой дуги (r, s) ; подмножество X'_2 – из всех маршрутов, содержащих эту дугу. Ребро дерева, соединяющее вершины X' и X'_1 , помечают $(\overline{r, s})$, ребро, соединяющее X' и X'_2 , – (r, s) . По пометкам на ребрах пути, ведущего в данную вершину, можно восстановить соответствующее ей множество маршрутов. Для вершины X' (см. рис. 4.1) таким множеством будет подмножество всех маршрутов из X , которые содержат дуги (i, j) , (p, q) и не содержат (k, l) .

Пусть \bar{c}' – приведенная матрица, найденная для вершины X' . Элемент матрицы \bar{c}' , стоящий на пересечении строки и столбца, соответствующих городам μ и ν , обозначим $\bar{c}'(\mu, \nu)$. Это не обязательно элемент, стоящий на пересечении μ -й строки и ν -го столбца матрицы \bar{c}' , так как в процессе построения \bar{c}' из исходной приведенной матрицы \bar{c} вычеркивают некоторые строки и столбцы. Матрицу \bar{c}' строят так, что отличный от $+\infty$ элемент $\bar{c}'(\mu, \nu)$ содержится в ней только тогда, когда в X' найдется маршрут, содержащий дугу (μ, ν) .

Способ выбора дуги (r, s) основан на стремлении сделать оценку $d(X'_2)$ поменьше, а оценку $d(X'_1)$ – побольше, чтобы увеличить вероятность выбора для дальнейшего ветвления множества X'_2 .

Стремление к уменьшению $d(X'_2)$ приведет к выбору такой дуги (r, s) , для которой

$$\bar{c}'(r, s) = 0, \quad (4.4)$$

поскольку все маршруты множества X'_2 содержат дугу (r, s) .

Стремление увеличить $d(X'_1)$ приведет к выбору среди дуг, удовлетворяющих условию (4.4), той, для которой максимально значение функции θ , определяемое по формуле

$$\theta(\mu, \nu) = \min_{\rho: \rho \neq \nu} \bar{c}'(\mu, \rho) + \min_{\sigma: \sigma \neq \mu} \bar{c}'(\sigma, \nu),$$

$$\theta(r, s) = \max_{\bar{c}'(\mu, \nu) = 0} \theta(\mu, \nu).$$

Здесь μ, σ – строки, ν, ρ – столбцы матрицы \bar{c}' (строки соответствуют пунктам отправления, а столбцы – пунктам прибытия).

Смысл введения функции θ состоит в том, что величина $\theta(\mu, \nu)$ является оценкой снизу для длины (по приведенной матрице расстояний) любого маршрута из X' , не содержащего дуги (μ, ν) .

Построение приведенных матриц и вычисление оценок снизу. Матрица расстояний для вершины $X'_1(\overline{r, s})$:

$$c'_1(\mu, \nu) = \begin{cases} \bar{c}(\mu, \nu), & (\mu, \nu) \neq (r, s), \\ +\infty, & (\mu, \nu) = (r, s). \end{cases}$$

Приведенную матрицу \bar{c}'_1 получают из c'_1 с помощью операции приведения.

Оценка функции (4.2) на множестве X'_1 : $d(X'_1) = d(X') + \theta(r, s)$, $\theta(r, s) = \tau$, где τ – сумма констант приведения.

Рассмотрим множество X'_2 . Все маршруты из этого множества содержат дугу (r, s) . Найдем максимальный связанный путь, который принадлежит всем маршрутам X'_2 . Пусть этот путь начинается в городе m и заканчивается в городе t (может быть, $m = r$ или $t = s$, или то и другое одновременно). Чтобы запретить подцикл, начинающийся

и заканчивающийся в m , положим $c'_2(t, m) = +\infty$, остальные элементы c'_2 полагаем равными соответствующим элементам c' . При этом строку, соответствующую городу r , и столбец, соответствующий городу s , в матрицу c'_2 не включаем, поскольку все матрицы из X'_2 содержат (r, s) .

Приведенную матрицу расстояний \bar{c}'_2 для вершины X'_2 получают из c'_2 с помощью операции приведения.

Оценка функции (4.2) на множестве X'_2 : $d(X'_2) = d(X') + \tau$.

Формирование списка кандидатов на ветвление. После вычисления каждой из оценок $d(X'_i)$, $i = 1, 2$ следует проверить, не состоит ли множество X'_i из единственного маршрута. Если в каждой строке и каждом столбце \bar{c}'_i оказалось только по одному элементу, отличному от $+\infty$, то множество X'_i содержит единственный маршрут, длина которого равна $d(X'_i)$. В этом случае текущее значение рекорда (наименьшее из уже вычисленных значений целевой функции) полагается равным минимальному из предыдущего текущего значения и $d(X'_i)$.

Если X'_i содержит более одного маршрута и $d(X'_i)$ меньше текущего значения рекорда, то X'_i включают в число кандидатов на ветвление.

Начальное значение рекорда равно $+\infty$, а начальный список кандидатов на ветвление состоит из X . Расчет останавливают, если наименьшая из оценок снизу кандидатов на ветвление не меньше текущего значения рекорда.

Пример решения задачи о коммивояжере

Дана матрица расстояний 6×6 (табл. 4.1). Необходимо с помощью метода ветвей и границ (алгоритм Литтла) решить задачу о коммивояжере (см. пример 2).

Таблица 4.1

$i \backslash j$	1	2	3	4	5	6
1	∞	7	16	21	2	17
2	13	∞	21	15	43	23
3	25	3	∞	31	17	9
4	13	10	27	∞	33	12
5	9	2	19	14	∞	51
6	42	17	5	9	23	∞

1. К таблице присоединяем столбец U_i , в котором записываем минимальные элементы соответствующих строк (табл. 4.2). Вычитаем значения U_{ij} из элементов каждой j -й строки матрицы.

Таблица 4.2

$i \backslash j$	1	2	3	4	5	6	U_i
1	∞	7	16	21	2	17	2
2	13	∞	21	15	43	23	13
3	25	3	∞	31	17	9	3
4	13	10	27	∞	33	12	10
5	9	2	19	14	∞	51	2
6	42	17	5	9	23	∞	5

2. К полученной матрице присоединяем строку V_j , в которой записываем минимальные элементы столбцов (табл. 4.3). Вычитаем элементы V_{ij} из соответствующих столбцов матрицы.

Таблица 4.3

$i \backslash j$	1	2	3	4	5	6
1	∞	5	14	19	0	15
2	0	∞	8	2	30	10
3	22	0	∞	28	14	6
4	3	0	17	∞	23	2
5	7	0	17	12	∞	49
6	37	12	0	4	18	∞
V_j	0	0	0	2	0	2

3. В результате получаем матрицу, приведенную по строкам и столбцам, которая представлена в виде табл. 4.4.

Таблица 4.4

$i \backslash j$	1	2	3	4	5	6
1	∞	5	14	17	0^{19}	13
2	0^3	∞	8	0^2	30	8
3	22	0^4	∞	26	14	4
4	3	0^0	17	∞	23	0^4
5	7	0^7	17	10	∞	47
6	37	12	0^8	2	18	∞

4. Находим константу приведения

$$\tau = \sum_{i=1}^6 U_i + \sum_{j=1}^6 V_j = 35 + 4 = 39.$$

Таким образом, нижняя граница множества всех гамильтоновых контуров – число $\tau = 39$.

5. Находим степени нулей приведенной матрицы. Для этого мысленно заменяем в ней каждый исследуемый нуль на знак « ∞ » и устанавливаем сумму минимальных элементов строки и столбца, соответствующих этому нулю. Степени нулей записаны в правых верхних углах ячеек, для которых $c_{ij} = 0$ (см. табл. 4.4).

6. Определяем максимальную степень нуля. Она равна 19 и соответствует ячейке (1, 5). Таким образом, кандидатом на включение в гамильтонов контур является дуга (1, 5).

7. Разбиваем множество всех гамильтоновых контуров Ω^0 на два подмножества: Ω_{15}^1 и Ω_{15}^1 . Подмножество Ω_{15}^1 содержит дугу (1, 5), Ω_{15}^1 – не содержит. Матрицу Ω_{15}^1 с дугой (1, 5) получаем из табл. 4.4 путем вычеркивания строки 1 и столбца 5. Чтобы не допускать образования негамильтонова контура, заменяем элемент (5, 1) на знак « ∞ » (табл. 4.5).

Таблица 4.5

$i \backslash j$	1	2	3	4	6
2	0	∞	8	0	8
3	22	0	∞	26	4
4	3	0	17	∞	0
5	∞	0	17	10	47
6	37	12	0	2	∞

8. Матрицу гамильтоновых контуров Ω_{15}^1 получаем из табл. 4.4 путем замены элемента (1, 5) на знак « ∞ » (табл. 4.6).

Таблица 4.6

$i \backslash j$	1	2	3	4	5	6	
1	∞	5	14	17	∞	13	5
2	0	∞	8	0	30	8	
3	22	0	∞	26	14	4	
4	3	0	17	∞	23	0	
5	7	0	17	10	∞	47	
6	37	12	0	2	18	∞	
					14		

9. Выполняем дополнительное приведение матрицы контуров Ω_{15}^1 : $\tau_{15}^1 = 0$. Нижняя граница множества Ω_{15}^1 равна $d(\Omega_{15}^1) = 39$.

10. Находим константу приведения для множества контуров Ω_{15}^1 : $\tau_{15}^1 = 5 + 14 = 19$. Следовательно, нижняя граница множества Ω_{15}^1 равна $d(\Omega_{15}^1) = 39 + 19 = 58$.

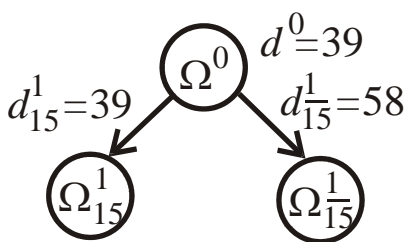


Рис. 4.2. Ветвление исходного множества

11. Сравниваем нижние границы подмножеств Ω_{15}^1 и Ω_{15}^1 (рис. 4.2).

Так как $d(\Omega_{15}^1) = 39 < d(\Omega_{15}^1) = 58$, то дальнейшему ветвлению подвергается подмножество Ω_{15}^1 .

Переходим к ветвлению подмножества Ω_{15}^1 . Его приведенная матрица представлена в табл. 4.7.

Таблица 4.7

$i \ j$	1	2	3	4	6
2	0^3	∞	8	0^2	8
3	22	0^4	∞	26	4
4	3	0^0	17	∞	0^4
5	∞	0^{10}	17	10	47
6	37	12	0^{10}	2	∞

12. Определяем степени нулей матрицы. Претендентами на включение в гамильтонов контур будут несколько дуг: (5, 2) и (6, 3). Для дальнейших расчетов выберем дугу (6, 3). Разбиваем множество Ω_{15}^1 на два подмножества Ω_{63}^2 и Ω_{63}^2 (табл. 4.8, 4.9). Чтобы не допустить образования негамильтонова контура, в табл. 4.8 заменяем элемент (3, 6) на знак « ∞ ».

Таблица 4.8

$i \ j$	1	2	4	6
2	0	∞	0	8
3	22	0	26	∞
4	3	0	∞	0
5	∞	0	10	47

Таблица 4.9

$i \ j$	1	2	3	4	6	
2	0	∞	8	0	8	
3	22	0	∞	26	4	
4	3	0	17	∞	0	
5	∞	0	17	10	47	
6	37	12	∞	2	∞	2
			8			

13. Определяем константы приведения для этих матриц: $\tau_{63}^2 = 0$, $\tau_{63}^2 = 10$. Следовательно, $d(\Omega_{63}^2) = 39$, $d(\Omega_{63}^2) = 39 + 10 = 49$.

Так как $d(\Omega_{63}^2) < d(\Omega_{63}^2)$, то дальнейшему ветвлению подлежит подмножество Ω_{63}^2 . Находим степени нулей матрицы (табл. 4.10).

Таблица 4.10

$i \ j$	1	2	4	6
2	0^3	∞	0^2	8
3	22	0^{22}	26	∞
4	3	0^0	∞	0^8
5	∞	0^{10}	10	47

14. Претендентом на включение в гамильтонов контур является дуга (3, 2). Разбиваем множество Ω_{63}^2 на два подмножества Ω_{32}^3 (табл. 4.11) и Ω_{32}^3 (табл. 4.12).

Таблица 4.11

$i \ j$	1	4	6	
2	0	0	8	
4	3	∞	0	
5	∞	10	47	10

Таблица 4.12

$i \ j$	1	2	4	6	
2	0	∞	0	8	
3	22	∞	26	∞	22
4	3	0	∞	0	
5	∞	0	10	47	

Очевидно, $d(\Omega_{32}^3) = 39 + 10 = 49$, $d(\Omega_{32}^3) = 39 + 22 = 61$. Следовательно, ветвлению нужно подвергнуть подмножество Ω_{32}^3 .

15. Переходим к ветвлению подмножества Ω_{32}^3 . Его приведенная матрица представлена в табл. 4.13.

Таблица 4.13

$i \quad j$	1	4	6
2	0^3	0^0	8
4	3	∞	0^{11}
5	∞	0^{37}	37

16. Определяем степени нулей. Претендентом на включение в гамильтонов контур является дуга (5, 4). Разбиваем множество Ω_{32}^3 на два подмножества: Ω_{54}^4 (табл. 4.14) и Ω_{54}^4 (табл. 4.15).

Таблица 4.14

$i \quad j$	1	6
2	0	8
4	3	0

Таблица 4.15

$i \quad j$	1	4	6	
2	0	0	8	
4	3	∞	0	
5	∞	∞	37	37

17. Находим оценки снизу: $d(\Omega_{54}^4) = 49$, $d(\Omega_{54}^4) = 49 + 37 = 86$. Следовательно, очередному ветвлению нужно подвергнуть подмножество Ω_{54}^4 . Но его матрица имеет размерность 2×2 . Поэтому в гамильтонов контур следует включить дуги, соответствующие в матрице Ω_{54}^4 нулевым элементам. Это дуги (2, 1) и (4, 6).

На рис. 4.3 изображено дерево ветвлений.

Определим полученный гамильтонов контур. В него вошли дуги $\{(1, 5), (6, 3), (3, 2), (5, 4), (2, 1), (4, 6)\}$.

Длина контура

$$c_{15} + c_{54} + c_{46} + c_{63} + c_{32} + c_{21} = 2 + 14 + 12 + 5 + 3 + 13 = 49.$$

Так как границы оборванных ветвей больше длины контура $1 - 5 - 4 - 6 - 3 - 2 - 1$, то этот контур имеет наименьшую длину.

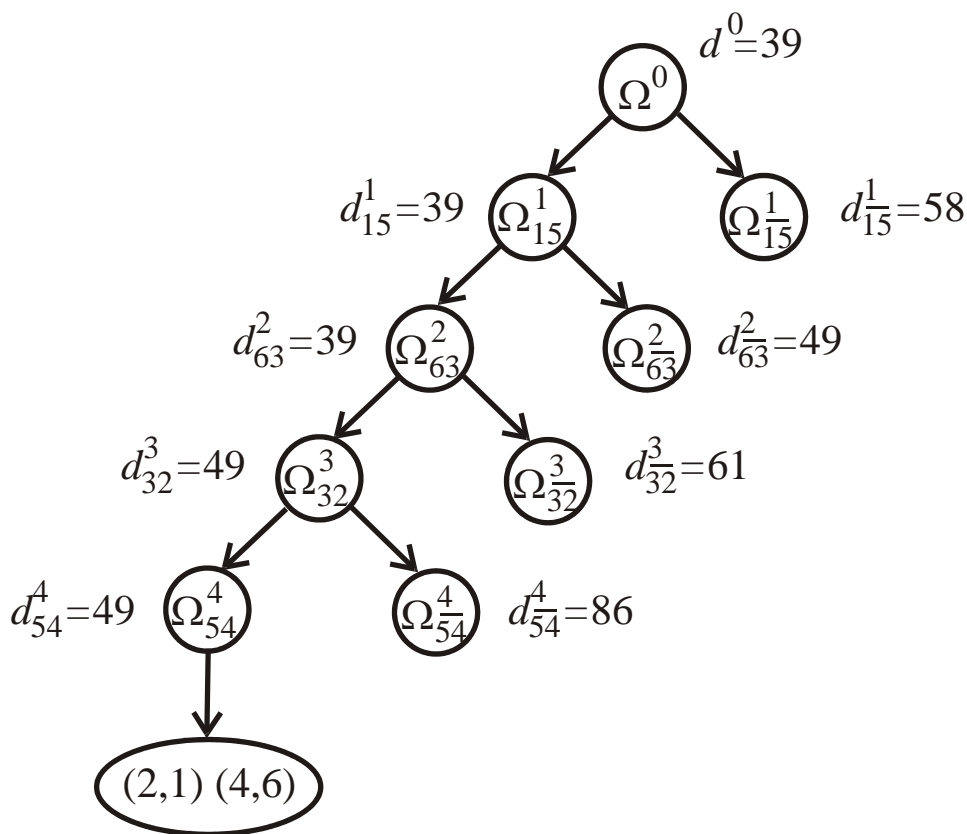


Рис. 4.3. Дерево ветвлений

Постановка задачи

Решить задачу о коммивояжере методом ветвей и границ. Матрицу расстояний $\{c_{ij}\}$ выбрать согласно заданному варианту.

Контрольные вопросы

1. В чем заключается специфика задач дискретной оптимизации? Приведите примеры таких задач.
2. В чем состоит идея метода ветвей и границ?
3. Что означает понятие «релаксация задачи»?
4. Сформулируйте задачу о коммивояжере в терминах теории графов.

Варианты заданий

Вариант № 1						
<i>i \ j</i>	1	2	3	4	5	6
1	∞	7	16	21	5	17
2	13	∞	21	15	43	23
3	30	25	∞	4	17	9
4	13	10	27	∞	33	12
5	9	40	19	14	∞	2
6	42	17	5	10	23	∞

Вариант № 5						
<i>i \ j</i>	1	2	3	4	5	6
1	∞	34	16	21	25	17
2	11	∞	21	15	43	3
3	30	5	∞	25	17	19
4	7	15	27	∞	33	12
5	9	40	2	14	∞	18
6	42	17	15	3	23	∞

Вариант № 2						
<i>i \ j</i>	1	2	3	4	5	6
1	∞	17	16	21	35	5
2	35	∞	10	15	43	23
3	30	25	∞	27	7	11
4	13	10	27	∞	33	12
5	2	40	19	14	∞	10
6	42	17	10	3	23	∞

Вариант № 6						
<i>i \ j</i>	1	2	3	4	5	6
1	∞	4	16	21	25	17
2	40	∞	21	15	43	23
3	30	25	∞	5	17	22
4	13	35	27	∞	7	12
5	9	40	2	14	∞	29
6	2	17	15	10	23	∞

Вариант № 3						
<i>i \ j</i>	1	2	3	4	5	6
1	∞	27	16	21	17	7
2	13	∞	21	15	43	23
3	17	25	∞	35	5	10
4	43	20	27	∞	33	13
5	9	40	19	2	∞	24
6	40	15	5	10	23	∞

Вариант № 7						
<i>i \ j</i>	1	2	3	4	5	6
1	∞	27	16	7	35	17
2	13	∞	21	15	43	23
3	30	25	∞	15	5	28
4	13	10	2	∞	33	12
5	9	40	19	14	∞	2
6	42	2	15	10	23	∞

Вариант № 4						
<i>i \ j</i>	1	2	3	4	5	6
1	∞	34	16	21	25	17
2	11	∞	21	15	43	23
3	30	25	∞	5	17	9
4	13	7	27	∞	33	12
5	9	40	19	14	∞	2
6	42	17	5	10	23	∞

Вариант № 8						
<i>i \ j</i>	1	2	3	4	5	6
1	∞	22	6	20	15	38
2	43	∞	21	15	3	23
3	30	25	∞	5	17	10
4	13	10	27	∞	33	12
5	10	40	14	24	∞	2
6	4	17	35	10	23	∞

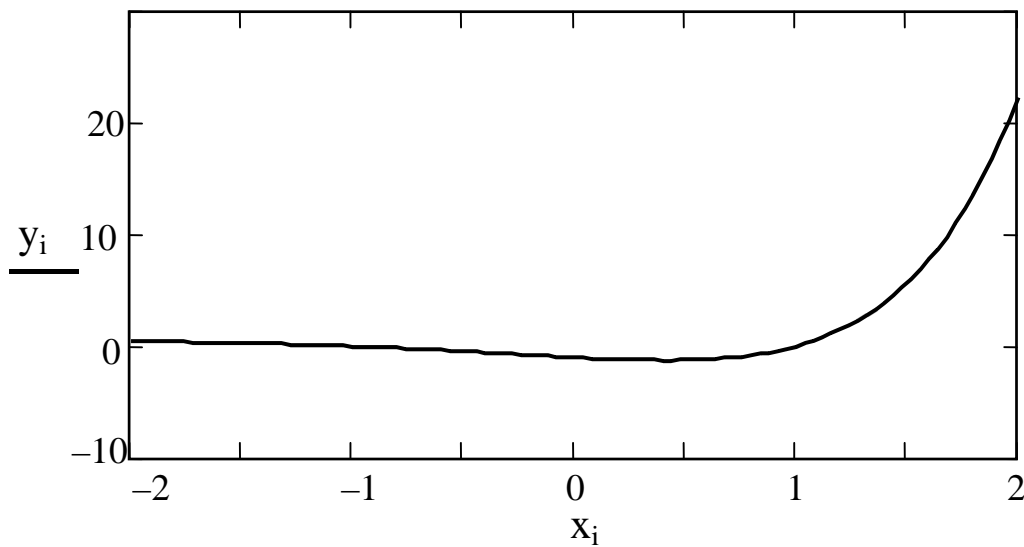
**ПРИМЕРЫ РЕАЛИЗАЦИИ МЕТОДОВ
ОДНОМЕРНОЙ МИНИМИЗАЦИИ**

Построение графика целевой функции $f(x)$ на интервале $[-2, 2]$.

Исходные данные:

$$n := 100 \quad i := 0..n \quad f(x) := \exp(x) \cdot (x^2 - 1) \quad \min := -2 \quad \max := 2$$

$$x_i := \min + \frac{\max - \min}{n} \cdot i \quad y_i := f(x_i)$$



Метод дихотомии (половинного деления)

```

half := | a ← -2
        | b ← 2
        | ξ ← 0.01
        | xs ← (b + a) / 2
        | while b - a > 0.1
        |   | a ← if(f(xs + ξ) > f(xs - ξ), a, xs - ξ)
        |   | b ← if(f(xs + ξ) < f(xs - ξ), b, xs + ξ)
        |   | xs ← (b + a) / 2
        | xs
    
```

half = 0.404
f(half) = -1.253

Метод золотого сечения

```

Golden := | a ← -2
          | b ← 2
          | L ←  $\frac{b - a}{1.618}$ 
          | while b - a > 0.1
          |   | x2 ← a + L
          |   | x1 ← b - L
          |   | a ← if(f(x1) > f(x2), x1, a)
          |   | b ← if(f(x1) > f(x2), b, x2)
          |   | L ←  $\frac{b - a}{1.618}$ 
          |   | xs ←  $\frac{b + a}{2}$ 
          | xs
    
```

Golden = 0.43

f(Golden) = -1.253

Метод парабол

a := -2 b := 2 $Dy(w) := \frac{d}{dw}f(w)$

$D2y(w) := \frac{d^2}{dw^2}f(w)$

```

Parabol := | i ← 0
           | xi ← 0
           | while |Dy(xi)| > 0.001
           |   | i ← i + 1
           |   | xi ← xi-1 -  $\frac{Dy(xi-1)}{D2y(xi-1)}$ 
           |   | xi
    
```

Parabol = 0.414 f(Parabol) = -1.254

$\xi := 0.05$ f(Parabol - ξ) = -1.248 f(Parabol + ξ) = -1.248

**ПРИМЕР ПРОГРАММЫ ОПТИМИЗАЦИИ ОБЪЕКТА ПО КРИТЕРИЮ
СООТВЕТСТВИЯ ЗАДАННОЙ ВЫХОДНОЙ ХАРАКТЕРИСТИКЕ**

Исходные данные:

$$i := 0..10 \quad j := 0..100 \quad n := 0..10 \quad AK_i :=$$

$$df := 10.0 \quad f_j := j \cdot df \quad ff_i := i \cdot 10 \cdot df$$

Расчет промежуточных значений требуемой АЧХ

$$FL(p, k) := \prod_i \text{if}[i = k, 1, (p - ff_i)] \quad FLA_n := \frac{AK_n}{FL(ff_n, n)}$$

$$S(d) := \sum_n (FLA_n \cdot FL(d, n)) \quad K(j) := S(f_j)$$

Выбор базисной точки и шага изменения параметра

$$Vx := 500.0 \quad H := 10.0 \quad x := Vx$$

Определение фактической частотной характеристики

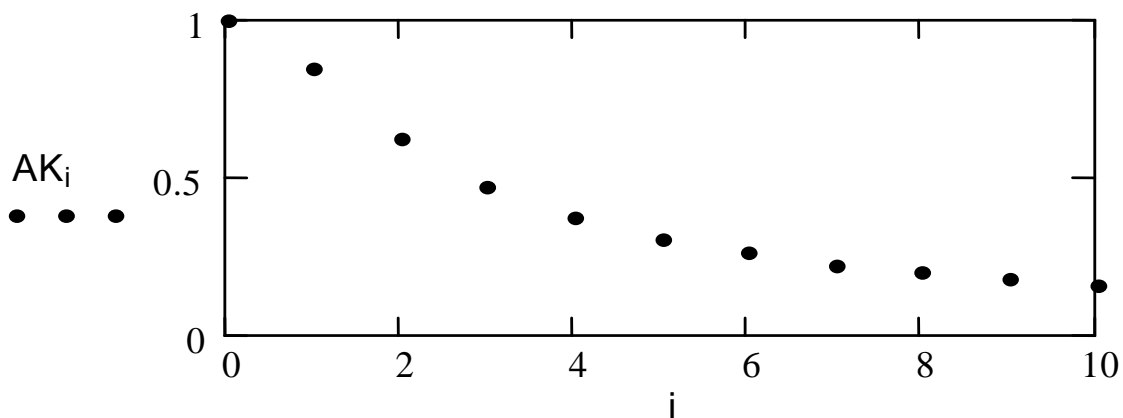
$$K0(j, x) := \left| \frac{1}{1 + 6.28i \cdot f_j \cdot x \cdot 10^{-6}} \right|$$

Задание целевой функции

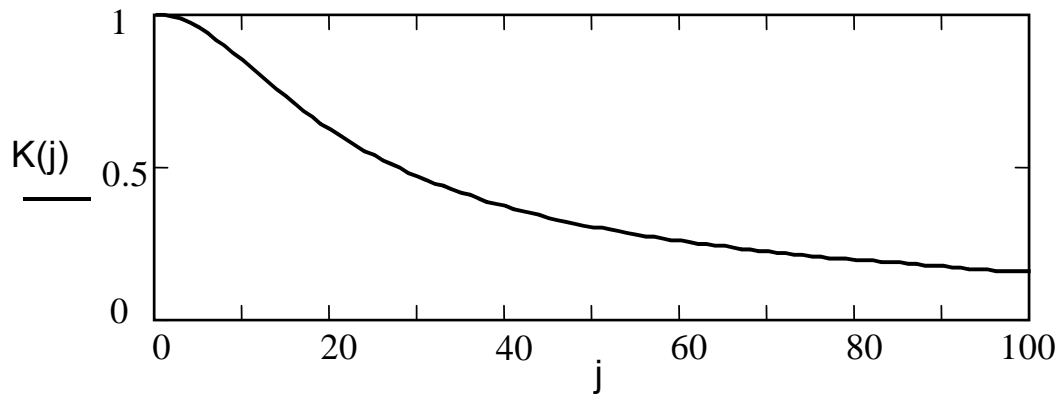
$$k := 0, 20..200 \quad T(j, x) := (K(j) - K0(j, x))^2 \quad TF(x) := \sum_j T(j, x)$$

$$ax_k := Vx + k \cdot H \quad AK_k := TF(ax_k)$$

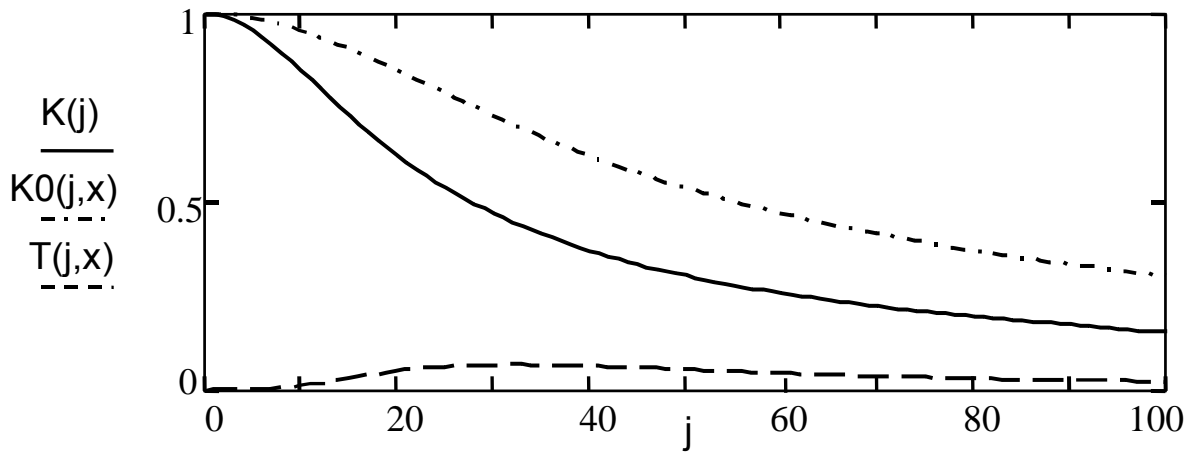
Построение графика дискретных данных требуемой АЧХ



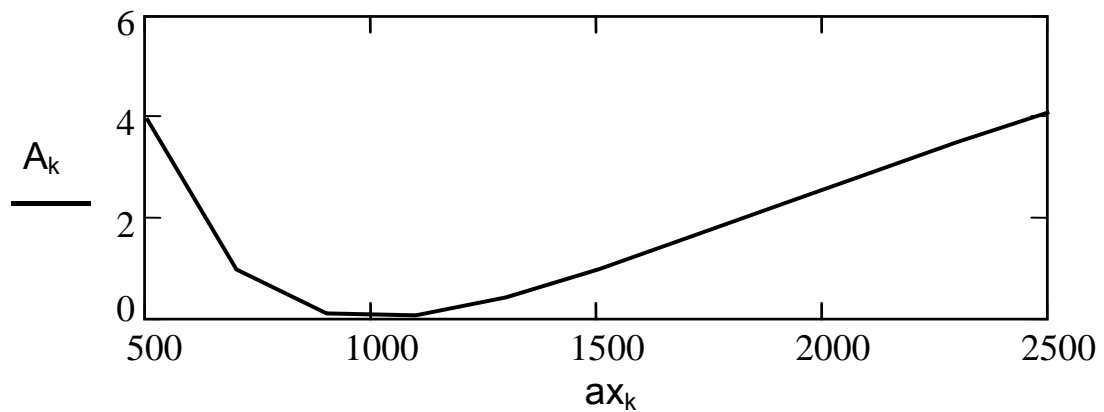
Полиномиальная интерполяция требуемой характеристики



Построение графиков требуемой АЧХ, фактической АЧХ (при базисном значении проектного параметра x) и целевой функции



Зависимость целевой функции от величины проектного параметра x



Программа поиска оптимального проектного параметра по методу Хука - Дживса

```

OptimX := | x ← Bx
          | while TF(x) > 0.1
          |   | x ← | ((x + H)) if TF(x + H) < TF(x)
          |   |   | ((x - H)) if TF(x - H) < TF(x)
          |   |   | x otherwise
          |   | if x ≠ Bx
          |   |   | Px ← [ Bx + 2·(x - Bx) ]
          |   |   | Bx ← x
          |   |   | H ← H
          |   |   | H ←  $\frac{H}{10}$  otherwise
          |   |   | x ← | Px if TF(Px) < TF(x)
          |   |   |   | Bx otherwise
  
```

Вывод оптимального значения параметра: $\text{OptimX} = 1.12 \times 10^3$

Соответствующее значение целевой функции: $\text{TF}(\text{OptimX}) = 0.085$

Оптимизация по двум параметрам $\vec{x} = (R, C)$

Выбор начального приближения Bx_i и длины шагов по координатам H_i осуществляется из практических соображений с учетом приемлемых величин проектных параметров R и C, например:

$Bx_i :=$

100
$5 \cdot 10^{-7}$

$H_i :=$

50
10^{-8}

$x := Bx$

```

OptimX := | x ← Bx
          | while TF(x) > 0.001
          |   | T ← TF(x)
          |   | X0 ←  $\begin{pmatrix} x_0 \\ x_1 + H_1 \end{pmatrix}$ 
          |   | X1 ←  $\begin{pmatrix} x_0 \\ x_1 - H_1 \end{pmatrix}$ 
          |   | X2 ←  $\begin{pmatrix} x_0 + H_0 \\ x_1 \end{pmatrix}$ 
          |   | X3 ←  $\begin{pmatrix} x_0 - H_0 \\ x_1 \end{pmatrix}$ 
          |   | x0 ←  $\begin{cases} (x_0 + H_0) & \text{if } TF(X2) < T \\ (x_0 - H_0) & \text{if } TF(X3) < T \\ x_0 & \text{otherwise} \end{cases}$ 
          |   | x1 ←  $\begin{cases} (x_1 + H_1) & \text{if } TF(X0) < T \\ (x_1 - H_1) & \text{if } TF(X1) < T \\ x_1 & \text{otherwise} \end{cases}$ 
          |   | for i ∈ 0..1
          |   |   | if x ≠ Bx
          |   |   |   | Px ← [ Bx + 2·(x - Bx) ]
          |   |   |   | Bx ← x
          |   |   | HH ← H
          |   |   | HHi ←  $\begin{cases} H_i & \text{if } x_i \neq Bx_i \\ H_i \cdot 0.1 & \text{otherwise} \end{cases}$ 
          |   |   | H ← HH
          |   |   | x ←  $\begin{cases} Px & \text{if } TF(Px) < TF(x) \\ Bx & \text{otherwise} \end{cases}$ 

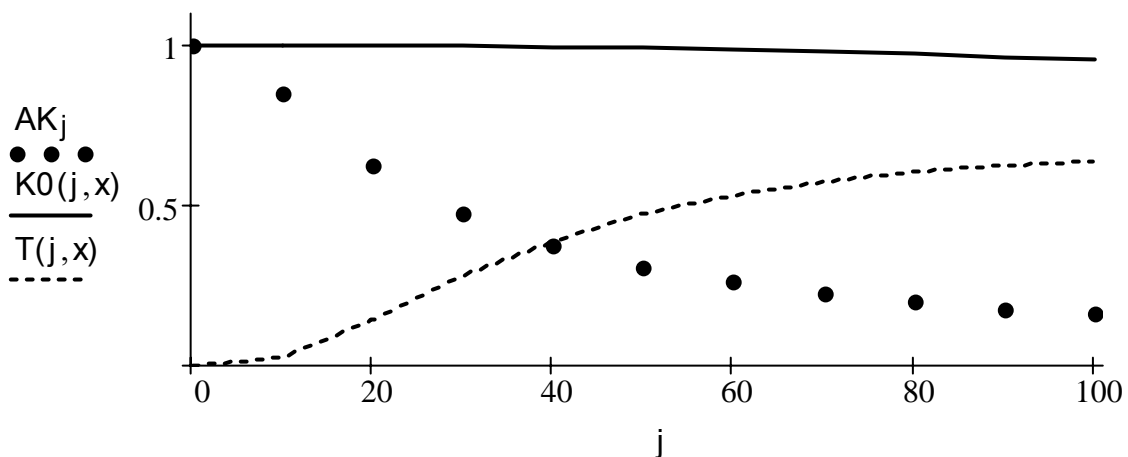
```

$$x = \begin{pmatrix} 1 \times 10^3 \\ 5 \times 10^{-7} \end{pmatrix} \quad \text{OptimX} = \begin{pmatrix} 1.3 \times 10^3 \\ 7.4 \times 10^{-7} \end{pmatrix}$$

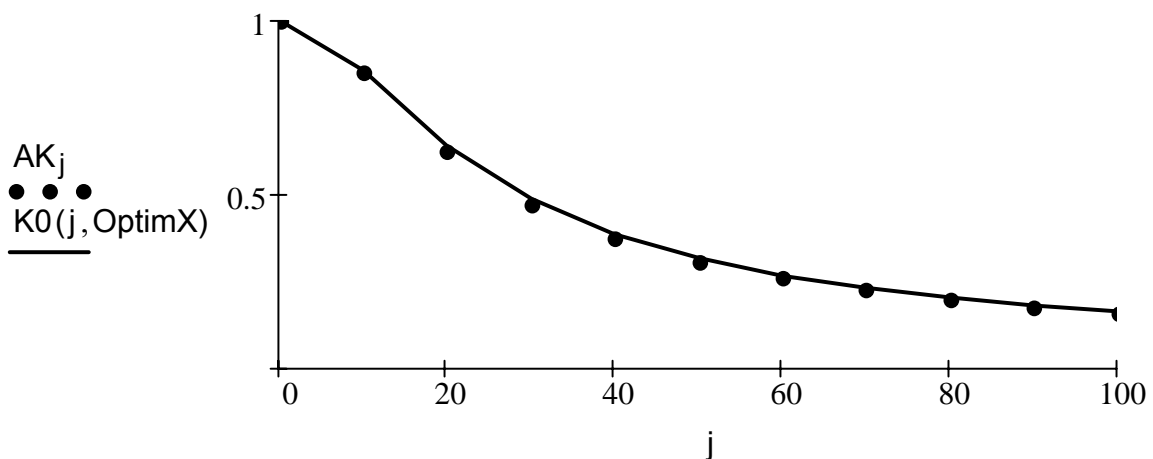
Соответствующее значение целевой функции

$$TF(\text{OptimX}) = 1.062 \times 10^{-3}$$

Построение графиков требуемой АЧХ (данные изображены точками), АЧХ для начального приближения ($R = 100$, $C = 5 \times 10^{-7}$) и целевой функции



Сравнение требуемой АЧХ с частотной характеристикой, полученной после оптимизации ($R^* = 1,3 \times 10^3$, $C^* = 7,4 \times 10^{-7}$)



Библиографический список

Акимов О.Е. Дискретная математика. Логика, группы, графы / О.Е. Акимов. – М. : Изд. дом "Лаборатория базовых знаний", 2003. – 376 с.

Аоки А. Введение в методы оптимизации / А. Аоки. – М. : Наука, 1977. – 344 с.

Банди Б. Методы оптимизации. Вводный курс: пер. с англ. / Б. Банди. – М. : Радио и связь, 1988. – 128 с.

Дегтярев Ю.И. Методы оптимизации: учеб. пособие для вузов / Ю.И. Дегтярев. – М. : Сов. радио, 1980. – 272 с.

Дмитришин Р.В. Оптимизация электронных схем на ЭВМ / Р.В. Дмитришин. – К. : Техніка, 1980. – 224 с.

Новиков Ф.А. Дискретная математика для программистов / Ф.А. Новиков. – СПб. : Изд. дом "Питер", 2002. – 304 с.

Соловьев А.А. Вариационное исчисление и методы оптимизации: учеб. / А.А. Соловьев. – Саров : РФЯЦ-ВНИИЭФ, 2004. – 241 с.

Сухарев А.Г. Курс методов оптимизации / А.Г. Сухарев, А.В. Тимохов, В.В. Федоров. – М. : Наука, 1986. – 328 с.

Шуп Т. Решение инженерных задач на ЭВМ: практическое руководство: пер. с англ. / Т. Шуп; под ред. В.Б. Миносцева. – М. : Мир, 1982. – 238 с.

Содержание

Введение.....	3
Лабораторная работа № 1. Сравнение эффективности численных методов одномерной безусловной минимизации	10
Лабораторная работа № 2. Многомерная безусловная минимизация	19
Лабораторная работа № 3. Оптимизация параметров радиоэлектронных устройств	31
Лабораторная работа № 4. Метод ветвей и границ	40
Приложение 1. Примеры реализации методов одномерной минимизации	56
Приложение 2. Пример программы оптимизации объекта по критерию соответствия заданной выходной характеристике	58
Библиографический список	63

Васильева Ирина Карловна
Ельцов Павел Евгеньевич

ЧИСЛЕННЫЕ МЕТОДЫ ОПТИМИЗАЦИИ

Редактор В.И. Филатова

Св. план, 2009

Подписано в печать 16.05.2009

Формат 60×84¹/₁₆. Бум. офс. № 2. Офс. печ.

Усл. печ. л. 3,6. Уч.-изд. л. 4. Т. 50 экз. Заказ 217. Цена свободная

Национальный аэрокосмический университет им. Н.Е. Жуковского

"Харьковский авиационный институт"

61070, Харьков-70, ул. Чкалова, 17

<http://www.khai.edu>

Издательский центр "ХАИ"

61070, Харьков-70, ул. Чкалова, 17

izdat@khai.edu