МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Юго-Западный государственный университет»
(ЮЗГУ)

Кафедра Вычислительная техника

УТВЕРЖДАЮ
Проректор по учебной работе
____________ О.Г. Локтионова
«30» 06 2017

МНОГОМЕРНАЯ ОПТИМИЗАЦИЯ МЕТОДАМИ НУЛЕВОГО ПОРЯДКА

Методические указания к лабораторным занятиям
по дисциплине «Методы оптимизации» для студентов
для студентов направления подготовки 09.04.01

Курск 2017

УДК 519.8

Составитель Ж.Т. Жусубалиев

Рецензент
Доктор технических наук, профессор *С.А. Филист*

Многомерная оптимизация методами нулевого порядка: методические указания к лабораторным занятиям по дисциплине «Методы оптимизации» / Юго-Зап. гос. ун-т; сост.: Ж.Т. Жусубалиев. Курск, 2017. – 16 с.: ил. 4, табл. 1.– Библиогр.: с. 16.

Рассматриваются методы поиска безусловного минимума функции нескольких переменных методами нулевого порядка, а также алгоритмы их численной реализации с помощью ЭВМ. Приводятся задания для упражнений.

Методические указания соответствуют требованиям программы, утвержденной учебно-методическим объединением по направлению подготовки «Информатика и вычислительная техника».

Предназначены для студентов направления подготовки 09.04.01 дневной и заочной форм обучения.

Текст печатается в авторской редакции

Подписано в печать . Формат 60×84 $^{1}/_{16}$.
Усл. печ. л. . Уч.-изд. л. . Тираж 30 экз. Заказ . Бесплатно.
Юго-Западный государственный университет.
305040, г. Курск, ул. 50 лет Октября, 94..

1. Общие сведения о методах прямого поиска

Методы, ориентированные на решение задач безусловной оптимизации, можно разделить на три широких класса в соответствии с типом используемой при реализации того или иного метода информации.

1. Методы прямого поиска, основанные на вычислении только значений целевой функции.
2. Градиентные методы, в которых используются точные значения первых производных $f(X)$.
3. Методы второго порядка, в которых наряду с первыми производными используются также вторые производные функции $f(X)$.

В методах прямого (или методах нулевого порядка) используют информацию только о значениях этой функции. Многие из этих методов не имеют строгого теоретического обоснования и построены на основе эвристических соображений. Для применения методов прямого поиска достаточно располагать лишь возможностью вычисления значения целевой функции в любой точке ее области определения. Это обстоятельство существенно расширяет сферу применения методов прямого поиска.

2. Метод Хука-Дживса

Метод, разработанный Хуком и Дживсом, представляет собой комбинацию «исследующего» поиска с циклическим изменением переменных и ускоряющегося поиска по образцу с использованием определенных эвристических правил. Исследующий поиск ориентирован на выявление характера локального поведения целевой функции и определение направлений ее убывания. Полученная в результате исследующего поиска информация затем используется в процессе поиска по образцу.

Для проведения исследующего поиска необходимо задать множество направлений поиска в виде координатных направлений в пространстве управляемых переменных задачи. Затем вдоль каждого из координатных направлений проводится поиск с заданной величиной шага точки оптимума на основе методов одномерной оптимизации. Величина шага может быть различной для разных координатных направлений и переменной в процессе изменения соответствующей переменной. Исследующий поиск начинается в некоторой исходной точке. Если значение минимизируемой функции $f(X)$ в пробной точке не превышает значения функции в исходной точке, то шаг поиска рассматривается как успешный. В противном случае необходимо вернуться в предыдущую точку и сделать шаг в противоположном направлении с последующей проверкой значения целевой функции $f(X)$. После перебора всех N координат (N – размерность вектора X) исследующий поиск завершается. Полученную в результате точку называют базовой точкой (на рис. 1 в точке X_0 произведен исследующий поиск и получена базовая точка X_1).

Поиск по образцу заключается в реализации единственного шага из полученной базовой точки вдоль прямой, соединяющей эту точку с предыдущей базовой точкой. Новая точка определяется в соответствии с формулой:

$$\overline{X_{k+1}} = X_k + \left(X_k - X_{k-1}\right).$$

Как только движение по образцу не приводит к уменьшению целевой функции, точка $\overline{X_{k+1}}$ фиксируется в качестве временной базовой точки и вновь поводится исследующий поиск. Если в результате получается точка с меньшим значением целевой функции, чем в точке X_k, то она рассматривается как новая базовая точка X_{k+1}. Если исследующий поиск неудачен, необходимо вернуться с точку X_k и провести исследующий поиск с целью выявления нового направления минимизации. В конечном счете возникает ситуация, когда такой поиск не приводит к успеху. В этом случае требуется уменьшить величину шага путем введения некоторого множителя и возобновить исследующий поиск.

Поиск завершается, когда величина шага становится достаточно малой. Последовательность точек, получаемую в процессе реализации метода, записывается в следующем виде:

X_k – текущая базовая точка;

X_{k-1} – предыдущая базовая точка;

$\overline{X_{k+1}}$ – точка, построенная при движении по образцу;

X_{k+1} – следующая (новая) базовая точка.

Если исследующий поиск с данной величиной шага неудачен, то она уменьшается и процедура продолжается. Поиск заканчивается, когда текущая величина шага станет меньше некоторой величины.

Алгоритм метода Хука-Дживса включает следующие шаги:

Шаг 1. Задать начальную точку X_0, число $\varepsilon > 0$ для остановки алгоритма, начальные величины шагов по координатным направлениям $\alpha_1, \ldots, \alpha_n \geq \varepsilon$, коэффициент уменьшения шага $\lambda > 0$.

Шаг 2. Осуществить исследующий поиск.

Шаг 3. Проверить, найдена ли точка с меньшим значением целевой функции:

а) если да, то перейти к шагу 5;

б) если нет, то перейти к шагу 4.

Шаг 4. Проверить условие окончания поиска:

а) если $\|X_{k+1} - X_k\| \leq \varepsilon$, то поиск закончить: $X_* \cong X_k$;

б) если $\|X_{k+1} - X_k\| > \varepsilon$, уменьшить приращения $\alpha_i = \dfrac{\alpha_i}{\lambda}$, $i = \overline{1, N}$ и перейти к шагу 2.

Шаг 5. Провести поиск по образцу: $\overline{X_{k+1}} = X_k + (X_k - X_{k-1})$.

Шаг 6. Провести исследующий поиск, используя $\overline{X_{k+1}}$ в качестве базовой точки. X_{k+1} – полученная в результате точка.

Шаг 7. Проверить выполнение неравенства $f(X_{k+1}) < f(X_k)$:

а) если неравенство верное, то положить $X_{k-1} = X_k$, $X_k = X_{k+1}$, и перейти к шагу 5.

б) если неравенство неверное, то перейти к шагу 4.

Замечания

▪ В алгоритме можно использовать одинаковую величину шага по координатным направлениям, т.е. вместо $\alpha_1, \ldots, \alpha_n$ применять α.

▪ Существует модификация метода, где при исследующем поиске и поиске по образцу используется одномерная минимизация. Тогда если функция $f(X)$ дифференцируема, метод сходится к стационарной точке.

Пример. Для функции $f(X)=8x_1{}^2+4x_1x_2+5_2{}^2$ найти точку минимума, используя начальную точку $X_0=(-2;-5)^T$.

Решение.

Зададим начальную точку $X_0=(-2;-5)^T$, $\varepsilon=10^{-4}$ для остановки алгоритма, начальные величины шагов по координатным направлениям $\alpha_1=\alpha_2=\alpha=1$, коэффициент уменьшения шага $\lambda=2$.

Исследующий поиск вокруг точки $X_0=(-2;-5)^T$. $f(X_0)=197$. Фиксируя $x_2=-5$, изменяем x_1 на величину $\alpha=1$: $X_1{}'=(-1;-5)^T$, $f\left(X_1{}'\right)=153$, $f\left(X_1{}'\right)<f(X_0)$, следовательно, поиск удачен, фиксируем $x_1=-1$ и изменяем x_2 на величину $\alpha=1$: $X_2{}'=(-1;-4)^T$, $f\left(X_2{}'\right)=104$, $f\left(X_2{}'\right)<f\left(X_1{}'\right)$, следовательно, поиск удачен. $X_1=X_2{}'=(-1;-4)^T$, $f(X_1)=104$.

Поиск по образцу. $\overline{X_2}=X_1+(X_1-X_0)=(0;-3)^T$, $f\left(\overline{X_2}\right)=45$.

Исследующий поиск вокруг точки $\overline{X_2}=(0;-3)^T$. В результате получаем точку $X_2=(1;-2)^T$, $f(X_2)=20$. Так как $f(X_2)<f(X_1)$, то поиск по образцу успешный и точка $X_2=(1;-2)^T$ становится новой базовой точкой при следующем проведении поиска по образцу.

Итерации продолжаются, пока уменьшение величины шага не укажет на окончание поиска в окрестности точка минимума $X_*=(0;\ 0)^T$, $f(X_*)=0$.

Метод Хука-Дживса характеризуется несложной стратегией поиска, относительной простотой вычислений и невысоким уровнем требований к объему памяти ЭВМ. Благодаря этому алгоритм Хука-Дживса находит широкое применение во всех областях инженерной практики.

3. Метод Розенброка

Направления исследующего поиска в методе Хука-Дживса фиксированы и совпадают с направлениями векторов стандартного базиса в R^n. Если выбор направления поиска проводить в процессе минимизации целевой функции путем построения на каждом шаге поиска нового ортонормированного базиса в R^n, то будет получен метод Розенброка. Такая стратегия поиска впервые была реализована в 1960 году

Суть метода Розенброка состоит в следующем. Задается начальная точка. Из нее осуществляется итеративный поиск направления убывания функции с помощью изменяемых дискретных шагов вдоль n линейно независимых и ортогональных[1] направлений. На первом шаге векторы направления поиска совпадают с векторами стандартного базиса в R^n. В случае удачного шага в исследуемом направлении его значение на следующей итерации увеличивается с помощью коэффициента растяжения, а в случае неудачи уменьшается за счет умножения на коэффициент сжатия (при этом направление поиска изменяется на противоположное).

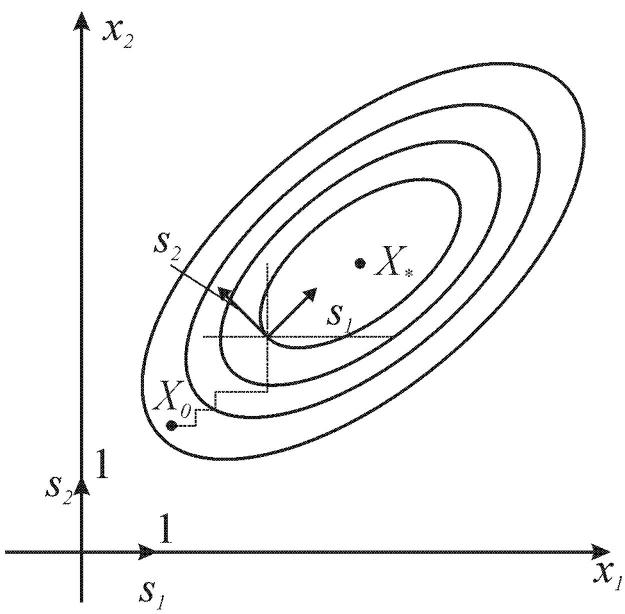

Рис. 1

Поиск в системе текущих направлений проводится до тех пор, пока все возможности уменьшения функции не будут исчерпаны. Если по каждому направлению поиска имеет место неудача, строится

[1] Пусть S_1, S_2, …, S_n – линейно независимые векторы, по норме равные единице. Они называются взаимно ортогональными, если для всех $i = 1, \ldots, n$ справедливо условие $S_i^T S_j = 0$, $j \neq i$.

новое множество линейно независимых и ортогональных направлений, и циклический поиск по отдельным направлениям продолжается. Новые направления поворачиваются по отношению к предыдущим так, что они оказываются вытянутыми вдоль оврага (рис. 1).

Алгоритм состоит из следующих шагов:

Шаг 1. Задать начальную точку X_0, число $\varepsilon > 0$ для остановки алгоритма, в качестве начальных линейно независимых и ортогональных направлений выбрать координатные направления:

$$S_1 = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix},\ S_2 = \begin{pmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{pmatrix},\ \ldots,\ S_n = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{pmatrix};$$

N – максимальное число неудачных серий шагов по всем направлениям на одной итерации. Положить $Y_1 = X_0$, $k = 1$, $i = 1$, $\alpha_i = \alpha_i^0$ для всех i.

Шаг 2. Определить значение α_i, минимизируя функцию $\varphi(\alpha) = f(Y_i + \alpha S_i)$. Сделать шаг по i-му направлению $Y_{i+1} = Y_i + \alpha S_i$:

Шаг 3. Проверить выполнение условий:

а) если $i < n$, то положить $i = i + 1$ и перейти к шагу 2 (сделать шаги по оставшимся направлениям);

б) если $i = n$, то $X_k = Y_{n+1}$ и перейти к шагу 4.

Шаг 4. Проверить условие окончания поиска:

а) если $\|X_k - X_{k-1}\| \le \varepsilon$, то поиск завершить: $X_* \cong X_k$;

б) если $\|X_k - X_{k-1}\| > \varepsilon$, перейти к шагу 5.

Шаг 5. Построить новый набор линейно независимых и взаимно ортогональных направлений поиска $S_1,\ S_2,\ \ldots,\ S_n$ с помощью процедуры Грамма-Шмидта:

$$a_i = \begin{cases} S_i,\ \alpha_i = 0, \\ \sum\limits_{j=1}^{n} \alpha_j S_j,\ \alpha_i \ne 0, \end{cases}\quad b_i = \begin{cases} a_i,\ i = 1, \\ a_i - \sum\limits_{j=1}^{i-1} (a_i,\ S_j) S_j,\ i \ge 2, \end{cases}\quad S_i = \frac{b_i}{\|b_i\|},\ i = 1, \ldots, n.$$

После вычисления векторов $S_1,\ S_2,\ \ldots,\ S_n$ переходим к шагу 2, полагая $k = k + 1$, $i = 1$.

Пример. Методом Розенброка найти минимум функции $f(X)=6x_1^{\,2}-4x_1x_2+3x_2^{\,2}+4\sqrt{5}(x_1-2x_2)+22$.

Решение.

Шаг 1. Зададим начальную точку $X_0=\begin{pmatrix}-2\\1\end{pmatrix}$, $S_1=\begin{pmatrix}1\\0\end{pmatrix}$, $S_2=\begin{pmatrix}0\\1\end{pmatrix}$, $\varepsilon=0{,}01$. Положим $Y_1=X_0$, $k=1$, $i=1$.

Результаты поиска приведены в таблице 1.

Таблица 1

k	$X_k^{\,T}$	$f(X_k)$
0	$(-2;\,1)$	57
1	$(-0{,}412;\,-3{,}259)$	-12,482
2	$(-2{,}020;\,-3{,}178)$	-23,817
3	$(-1{,}834;\,-4{,}234)$	-27,245
4	$(-2{,}159;\,-4{,}177)$	-27,794
5	$(-2{,}168;\,-4{,}444)$	-27,977
6	$(-2{,}219;\,-4{,}428)$	-27,996
7	$(-2{,}233;\,-4{,}472)$	-28,000

4. Метод Пауэлла

Если в методе Розенброка использовать факт, что минимум квадратичной функции может быть найден не более чем за n шагов при условии, что поиск ведется вдоль сопряженных относительно матрицы Гессе направлений, будет получен метод Пауэлла или метод сопряженных направлений. Так как достаточно большой класс целевых функций может быть представлен в окрестности точки минимума своей квадратичной аппроксимацией, описанная идея применяется и для неквадратичных функций.

Задается начальная точка и направления S_1, S_2, …, S_n, совпадающие с координатными. Находится минимум $f(X)$ при последовательном движении по $(n+1)$ направлениям с помощью одного из методов одномерной минимизации.

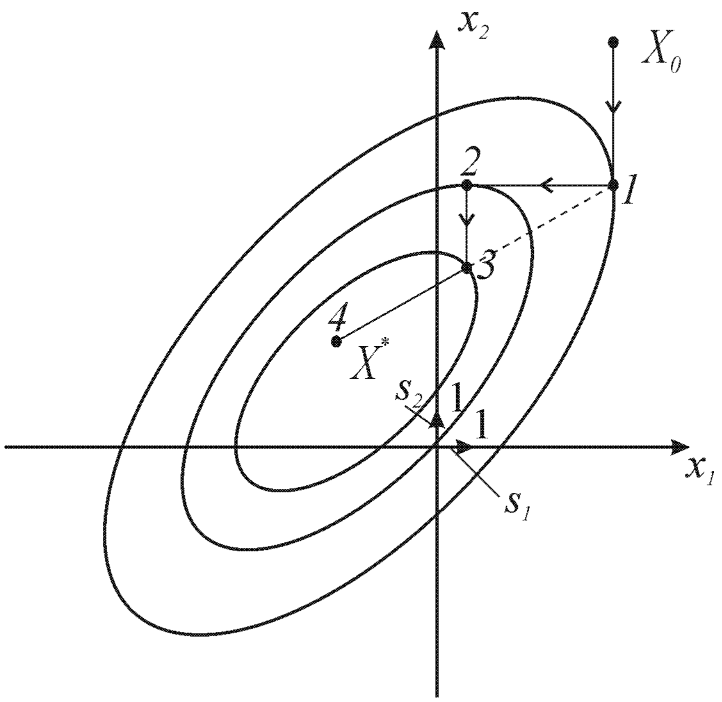

Рис. 2

При этом полученная ранее точка минимума берется в качестве исходной для поиска по следующему направлению, а направление S_n используется как при первом $(S_0 = S_n)$, так и при последнем поиске. Находится новое направление поиска, сопряженное[2] с S_n. Оно проходит через точки, полученные при первом поиске. Заменяется S_1 на S_2, S_2 на S_3 и т.д. Направление S_n заменяется сопряженным направлением, после чего повторяется поиск по $(n+1)$ направлениям, уже не содержащим старого направления S_1. Для квадратичных функций последовательность n^2 одномерных поисков приводит к точке минимума (если все операции выполнены точно).

Построение сопряженного направления для квадратичной функции при $n = 2$ изображено на рис. 2. Оно проходит через точки 1 и 3.

Алгоритм включает следующие шаги.

---

[2] Пусть H – симметрическая матрица размера $n \times n$. Векторы S_1, S_2, …, S_n называются H-сопряженными или просто сопряженными, если $S_i^T H S_j = 0$ при всех $i \neq j$.

Шаг 1. Задать начальную точку X_0, число $\varepsilon > 0$ для остановки алгоритма, в качестве начальных линейно независимых и ортогональных направлений выбрать координатные направления:

$$S_1 = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix},\ S_2 = \begin{pmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{pmatrix},\ \ldots,\ S_n = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{pmatrix};$$

N – максимальное число неудачных серий шагов по всем направлениям на одной итерации. Положить $Y_1 = X_0$, $k = 1$, $i = 1$, $\alpha_i = \alpha_i^0$ для всех i.

Шаг 2. Определить значение α_i, минимизируя функцию $\varphi(\alpha) = f(Y_i + \alpha S_i)$. Сделать шаг по i-му направлению $Y_{i+1} = Y_i + \alpha S_i$.

Шаг 3. Проверить выполнение условий:

а) если $i < n$, то положить $i = i + 1$ и перейти к шагу 2 (сделать шаги по оставшимся направлениям);

б) если $i = n$, то $X_k = Y_{n+1}$ и перейти к шагу 4.

Шаг 4. Полагаем $S = Y_{n+1} - Y_1$ и находим значение α_i, минимизируя функцию $\varphi(\alpha) = f(Y_i + \alpha S_i)$. Вычисляем $Z_j = Y_{n+1} + \alpha S$.

а) если $j < n$, то для всех $j = \overline{1, N-1}$ заменяем S_i на S_{i+1}, $S_n = S$, $Y_1 = Z_j$, $i = 1$, $j = j + 1$ и перейти к шагу 2;

б) если $j = n$, то перейти к шагу 5.

Шаг 5. $X_k = Z_n$. Проверить условие окончания поиска:

а) если $\|X_k - X_{k-1}\| \le \varepsilon$, то поиск завершить: $X_* \cong X_k$;

б) если $\|X_k - X_{k-1}\| > \varepsilon$, $Y_1 = X_k$, $j = 1$, $i = 1$, $k = k + 1$ и перейти к шагу 2.

5. Методы случайного поиска

Методы прямого поиска, в которых поиск ведется на основе рекурсивного перебора значений целевой функции в направлениях случайного заданного множества, называются методами случайного поиска.

5.1. Адаптивный метод случайного поиска

Итерационный расчет проводится по формуле:

$$X_{k+1} = X_k + \alpha_k S_k,$$

где $\alpha_k > 0$ – величина шага; S_k – случайный вектор единичной длины, определяющий направление поиска; k – номер итерации.

На текущей итерации при помощи генерирования случайных векторов S_k получаются точки, лежащие на гиперсфере радиуса α_k с центром в точке X_k (рис. 3). Если значение функции в полученной точке не меньше, чем в центре, шаг считается неудачным (точки Y_1, Y_2 при поиске из X_0; точки Y_1, Y_3 при поиске из X_1). Если число неудачных шагов из текущей точки достигает некоторого числа M, дальнейший поиск продолжается из той же точки, но с меньшим шагом до тех пор, пока он не станет меньше заранее заданной величины R. Если же значение функции в полученной точке меньше, чем в центре, шаг считается удачным, и в найденном направлении делается увеличенный шаг, играющий роль ускоряющего шага. Если при этом значение функции снова меньше, чем в центе, направление считается удачным, и дальнейший поиск продолжается из этой точки (точки $Z_3 = X_1$ при поиске из X_0, $Z_4 = X_2$ при поиске из X_1). Если же значение функции стало не меньше, чем в центре, направление считается неудачным, и поиск продолжается из старого центра (в точке Y_2 при поиске из X_1 функция меньше, чем в X_1, а в точке Z_2 уже не меньше, поэтому направление $(Z_2 - X_1)$ неудачное).

Алгоритм включает следующие шаги:

Шаг 1. Задать начальную точку X_0, коэффициенты расширения $l \geq 1$ и сжатия $0 < \gamma < 1$, M – максимальное число неудачно выполненных испытаний на текущей итерации, $\alpha_0 = 1$ – начальную величину шага, R – минимальную величину шага, N – максимальное число итераций. Положить $k = 0$, $j = 1$.

Шаг 2. Получить случайный вектор $S_j = \left(s_{1j}, s_{2j}, ..., s_{nj}\right)^T$, где s_{ij} – случайная величина, равномерно распределенная на интервале $[-1, 1]$.

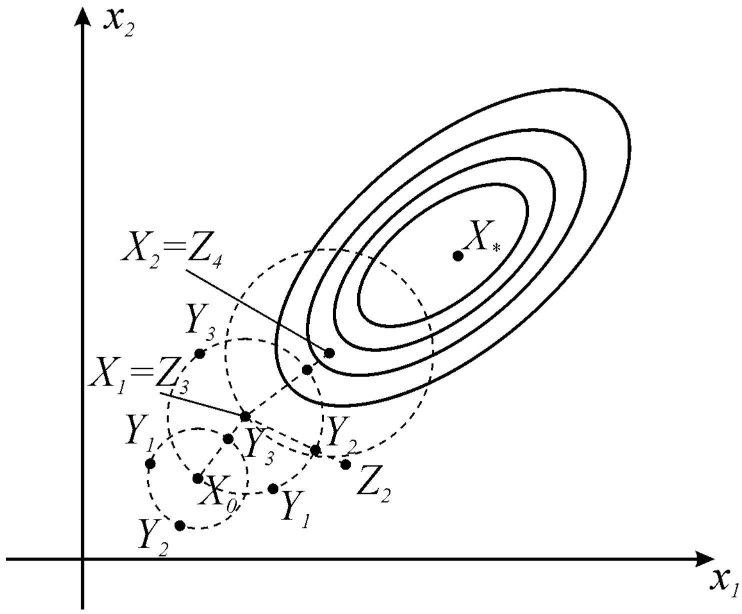

Рис. 3

Шаг 3. Вычислить $Y_j = X_k + \alpha_k \dfrac{S_j}{\left\| S_j \right\|}$.

Шаг 4. Проверить выполнение условий:

а) если $f(Y_j) < f(X_k)$, шаг удачный. Положить $Z_j = X_k + \alpha(Y_j - X_k)$. Определить, является ли текущее направление $Y_j - X_k$ удачным:

– если $f(Z_j) < f(X_k)$, направление поиска удачное. Положить $X_{k+1} = Z_j$, $\alpha_{k+1} = l\alpha_k$, $k = k+1$ и проверить условие окончания. Если $k < N$, положить $j = 1$ и перейти к шагу 2. Если $k = N$, поиск завершить: $X_* \cong X_k$;

– если $f(Z_j) \geq f(X_k)$, направление поиска неудачное. Перейти к шагу 5;

б) если $f(Y_j) \geq f(X_k)$, шаг неудачный и перейти к шагу 5.

Шаг 5. Оценить число неудачных шагов из текущей точки:

а) если $j < M$, следует положить $j = j+1$ и перейти к шагу 2;

б) если $j = M$, проверить условие окончания:

– если $\alpha_k \leq R$, процесс закончить: $X_* \cong X_k$, $f(X_*) \cong f(X_k)$;

– если $\alpha_k > R$, положить $\alpha_k = \gamma\alpha_k$, $j = 1$ и перейти к шагу 2.

Замечания

- величина s_{ij}, равномерно распределенная на интервале $[-1, 1]$, генерируется обычно с помощью датчиков псевдослучайных чисел на ЭВМ. Вырабатывается случайная величина η_{ij}, равномерно распределенная на интервале $[-1, 1]$, а затем используется линейное преобразование: $s_{ij} = 2\eta_{ij} - 1$;
- если выполнено условие окончания $\alpha_k \le R$, то в качестве ответа можно использовать любую точку внутри шара с радиусом α_k и центром в точке X_k.

5.2. Метод случайного поиска с возвратом при неудачном шаге

Задается начальная точка X_0. Каждая последующая точка находится по формуле

$$X_{k+1} = X_k + \alpha_k S_k,$$

где $\alpha_k > 0$ – величина шага; S_k – случайный вектор единичной длины, определяющий направление поиска; k – номер итерации. На текущей итерации при помощи генерирования случайных векторов S_k получаются точки, лежащие на гиперсфере радиуса α_k с центром в точке X_k (рис. 4). Если значение функции в полученной точке не меньше, чем в центре, шаг считается неудачным (точки Y_1, Y_2 при поиске из X_0; точки Y_1, Y_2, Y_3 при поиске из X_1), происходит возврат в текущий центр и поиск продолжается. Если число неудачных шагов из текущей точки достигает некоторого числа M, дальнейший поиск продолжается из той же точки, но с меньшим шагом до тех пор, пока он не станет меньше заранее заданной величины R. Если же значение функции в полученной точке меньше, чем в центре, шаг считается удачным и дальнейший поиск продолжается из этой точки.

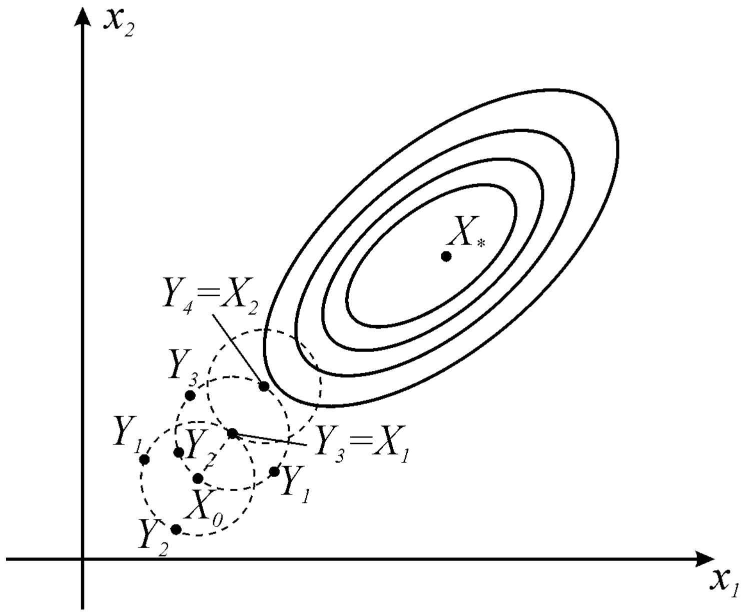

Рис. 4

Алгоритм включает следующие шаги.

1. Задать начальную точку X_0, коэффициент сжатия $0<\gamma<1$, M – максимальное число неудачно выполненных испытаний на текущей итерации, α_0 – начальную величину шага, R – минимальную величину шага, N – максимальное число итераций. Положить $k=0$, $j=1$.

2. Получить случайный вектор $S_j=\left(s_{1j},\, s_{2j},\, ...,\, s_{nj}\right)^T$, где s_{ij} – случайная величина, равномерно распределенная на интервале $[-1, 1]$.

3. Вычислить $Y_j = X_k + \alpha_k \dfrac{S_j}{\left\|S_j\right\|}$.

4. Проверить выполнение условий:

а) если $f(Y_j)< f(X_k)$, шаг удачный. Положить $X_{k+1}=Y_j$, $\alpha_{k+1}=\alpha_k$, $k=k+1$ и поверить условие окончания. Если $k<N$ положить $j=1$ и перейти к шагу 2. Если $k=N$, поиск завершить: $X_* \cong X_k$;

б) если $f(Y_j)\geq f(X_k)$, шаг неудачный и перейти к шагу 5.

5. Оценить число неудачных шагов из текущей точки:

а) если $j<M$, следует положить $j=j+1$ и перейти к шагу 2;

б) если $j = M$, проверить условие окончания:

– если $\alpha_k \le R$, процесс закончить: $X_* \cong X_k$, $f(X_*) \cong f(X_k)$;

– если $\alpha_k > R$, положить $\alpha_k = \gamma\alpha_k$, $j = 1$ и перейти к шагу 2.

6. Задачи для упражнений

1. Найти минимум функции $f(X) = x_1^3 + x_2^2 - 3x_1 - 2x_2 + 2$ методами конфигураций, сопряженных направлений.

2. Найти минимум функции $f(X) = x_1^4 + x_2^4 + 2x_1^2 x_2^2 - 4x_1 + 3$ методами конфигураций, сопряженных направлений.

3. Методами конфигураций, сопряженных направлений найти минимум функции $f(X) = \left(x_1^2 + x_2 - 11\right)^2 + \left(x_1 + x_2^2 - 7\right)^2$.

4. Методами конфигураций, сопряженных направлений, Розенброка найти минимум функции
$f(X) = 1 - 2x_1 - 2x_2 - 4x_1x_2 + 10x_1^2 + 2x_2^2$.

5. Найти минимум функции $f(X) = 4(x_1 - 5)^2 + (x_2 - 6)^2$ методами конфигураций, сопряженных направлений.

---

1. Пантелеев, А.В. Методы оптимизации в примерах и задачах [Текст]: учеб. пособие / А.В. Пантелеев, Т.А. Летова; 2-е изд., исправл.; М.: Высш. шк., 2005. 544 с.: ил.

2. Аттетков, А.В. Методы оптимизации [Текст]: учебник для вузов / А.В. Аттетков, С.В. Галкин, В.С. Зарубин / Под ред. В.С. Зарубина, А.П. Крищенко; 2-е изд.; М.: Изд-во МГТУ им. Н.Э. Баумана, 2003. 440 с.