**Министерство образования Российской Федерации**

**МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**им. Н.Э. БАУМАНА**

Факультет: Информатика и системы управления Кафедра: Информационная безопасность (ИУ8)

МЕТОДЫ ОПТИМИЗАЦИИ

Лабораторная работа №2 на тему:

«Исследование метода случайного поиска экстремума функции одного переменного»

Вариант 5

**Преподаватель:**

Коннова Н. С.

**Студент**:

Григорьев А. С.

**Группа:**

ИУ8-32

Москва 2020

# Цель работы

# Изучение метода случайного поиска экстремума на примере унимодальной и мультимодальной функций одного переменного.

# Постановка задачи

1. На интервале [-2, 4] задана унимодальная функция одного переменного f (x) = -cos (0,5x) - 1. Используя метод случайного поиска осуществить поиск минимума f (x) с заданной вероятностью попадания в окрестность экстремума P при допустимой длине интервала неопределенности *e*. Определить необходимое число испытаний N. Численный эксперимент выполнить для значений P = 0,90, 0,91, ..., 0,99 и значений *e* = (4 – (-2)) \* q, где q = 0,005, 0,010, ..., 0,100.

# Последовательность действий:

# - определить вероятность непопадания в *e* -окрестность экстремума за одно испытание;

# - записать выражение для вероятности непопадания в *e* -окрестность экстремума за N испытаний;

# - из выражения для определить необходимое число испытаний N в зависимости от заданных = P и *e*.

# 2. При аналогичных исходных условиях осуществить поиск минимума f (x), модулированной сигналом sin5x, т. е. мультимодальной функции f (x) \* sin5x.

# Ход работы

# В данном методе P – это вероятность того, что найденная точка минимума находится в интервале неопределенности, а q – это вероятность попадания в интервал неопределенности для отдельно взятой точки. Тогда вероятность непопадания в интервал неопределенности за одно испытание будет равна . Вероятность непопадания в интервал неопределенности за N испытаний будет равна . Отсюда же N = .

**Рис. 1** График функции f(x) = -cos (0,5x) - 1

# 

**Табл. 1** Зависимость N от P и q.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **q/P** | **0,9** | **0,91** | **0,92** | **0,93** | **0,94** | **0,95** | **0,96** | **0,97** | **0,98** | **0,99** |
| **0.005** | 460 | 481 | 504 | 531 | 562 | 598 | 643 | 700 | 781 | 919 |
| **0.01** | 230 | 240 | 252 | 265 | 280 | 299 | 321 | 349 | 390 | 459 |
| **0.015** | 153 | 160 | 168 | 176 | 187 | 199 | 213 | 233 | 259 | 305 |
| **0.02** | 114 | 120 | 126 | 132 | 140 | 149 | 160 | 174 | 194 | 228 |
| **0.025** | 91 | 96 | 100 | 106 | 112 | 119 | 128 | 139 | 155 | 182 |
| **0.03** | 76 | 80 | 83 | 88 | 93 | 99 | 106 | 116 | 129 | 152 |
| **0.035** | 65 | 68 | 71 | 75 | 79 | 85 | 91 | 99 | 110 | 130 |
| **0.04** | 57 | 59 | 62 | 66 | 69 | 74 | 79 | 86 | 96 | 113 |
| **0.045** | 51 | 53 | 55 | 58 | 62 | 66 | 70 | 77 | 85 | 101 |
| **0.05** | 45 | 47 | 50 | 52 | 55 | 59 | 63 | 69 | 77 | 90 |
| **0.055** | 41 | 43 | 45 | 48 | 50 | 53 | 57 | 62 | 70 | 82 |
| **0.06** | 38 | 39 | 41 | 43 | 46 | 49 | 53 | 57 | 64 | 75 |
| **0.065** | 35 | 36 | 38 | 40 | 42 | 45 | 48 | 53 | 59 | 69 |
| **0.07** | 32 | 34 | 35 | 37 | 39 | 42 | 45 | 49 | 54 | 64 |
| **0.075** | 30 | 31 | 33 | 35 | 37 | 39 | 42 | 45 | 51 | 60 |
| **0.08** | 28 | 29 | 31 | 32 | 34 | 36 | 39 | 43 | 47 | 56 |
| **0.085** | 26 | 28 | 29 | 30 | 32 | 34 | 37 | 40 | 45 | 52 |
| **0.09** | 25 | 26 | 27 | 29 | 30 | 32 | 35 | 38 | 42 | 49 |
| **0.095** | 24 | 25 | 26 | 27 | 29 | 31 | 33 | 36 | 40 | 47 |
| **0.1** | 22 | 23 | 24 | 26 | 27 | 29 | 31 | 34 | 38 | 44 |

**Табл. 2** Зависимость и от соответсвующих q и N

# 

# Случайно выбираем N точек в заданном отрезке [-2, 4], определим значение унимодальной функции в этих точках и среди них найдем наименьшее значение.

**Табл. 3** Результаты поиска экстремума f(x) в зависимости от P и q.

# 

# Аналогичные вычисления требуется проделать для мультимодальной функции.

**Рис. 2** График функции f(x) \* sin5x

# 

# Табл. 4 Результаты поиска экстремума f(x) \* sin5x в зависимости от P и q.

# 

# 

# Выводы

В ходе проделанной работы был освоен метод случайного для нахождения экстремума унимодальной и мультимодальной функций одного переменного. Было выяснено, что применимость метода случайного поиска не зависит от того, является ли функция унимодальной или мультимодальной. Для увеличения вероятности попадания в заданный интервал или для уменьшения интервала неопределенности необходимо увеличивать число случайных точек.

# Приложение А.

*Файл ‘lab2.cpp’.*

Вариант 5. '''

#include <iostream>

#include <cmath>

#include <algorithm>

#include <array>

#include <ctime>

#include <vector>

double f(double x) {

return -cos(0.5 \* x) - 1;

}

double g(double x) {

return sin(5\*x)\*f(x);

}

double fRand(double fMin, double fMax)

{

double f = (double)rand() / RAND\_MAX;

return fMin + f \* (fMax - fMin);

}

int main() {

srand(time(0));

const double a = -2;

const double b = 4;

double x = 0;

double N = 0;

double P1 = 0;

double Pn = 0;

std::array<double, 10> P;

std::array<double, 20> q;

std::vector<double> func\_v;

P[0] = 0.9;

q[0] = 0.005;

for (int i = 1; i < 10; i++) {

P[i] = P[i - 1] + 0.01;

}

for (int i = 1; i < 20; i++) {

q[i] = q[i - 1] + 0.005;

}

std::cout << "f(x)" << std::endl;

for (int i = 0; i < 10; i++) {

std::cout << "P = " << P[i] << std::endl;

for (int j = 0; j < 20; j++) {

N = ceil(log(1 - P[i]) / log(1 - q[j]));

P1 = 1 - q[j];

Pn = pow(P1, N);

std::cout <<" P1 = " << P1 << " Pn = " << Pn << std::endl;

std::cout << " q = "<< q[j] << " N = " << N;

for (int k = 1; k <= N; k++) {

x = fRand(a, b);

func\_v.push\_back(f(x));

}

auto result = std::min\_element(func\_v.begin(), func\_v.end());

std::cout << " Fmin = " << \*result << std::endl;

func\_v.clear();

}

}

std::cout << "f(x) \* sin5x" << std::endl;

for (int i = 0; i < 10; i++) {

std::cout << "P = " << P[i] << std::endl;

for (int j = 0; j < 20; j++) {

N = ceil(log(1 - P[i]) / log(1 - q[j]));

P1 = 1 - q[j];

Pn = pow(P1, N);

std::cout << " P1 = " << P1 << " Pn = " << Pn << std::endl;

std::cout << " q = " << q[j] << " N = " << N;

for (int k = 1; k <= N; k++) {

x = fRand(a, b);

func\_v.push\_back(g(x));

}

auto result = std::min\_element(func\_v.begin(), func\_v.end());

std::cout << " Fmin = " << \*result << std::endl;

func\_v.clear();

}

}

}