



**Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)**

ФАКУЛЬТЕТ «Информатика и системы управления» (ИУ)

КАФЕДРА «Информационная безопасность» (ИУ8)

Отчёт

по лабораторной работе № 2

по дисциплине «Теория систем и системный анализ»

**Тема: «Исследование метода случайного поиска экстремума функции одного
переменного»**

Вариант 6

**Выполнил: Калинин Д. В.,
студент группы ИУ8-31**

**Проверил: Коннова Н.С.,
доцент каф. ИУ8**

1. Цель работы

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

2. Условие задачи

1. На интервале $[-5, 2]$ задана унимодальная функция одного переменного. $f(x) = (1 - x)^2 + e^x$. Используя метод случайного поиска, осуществить поиск минимума $f(x)$ с заданной вероятностью попадания в окрестность экстремума P при допустимой длине интервала неопределенности ε . Определить необходимое число испытаний N . Численный эксперимент выполнить для значений $P = 0.90, 0.91, \dots, 0.99$ и значений $\varepsilon = (b - a)q$, где $q = 0.005, 0.010, \dots, 0.100$.

Последовательность действий:

- определить вероятность P_1 непадания в ε -окрестность экстремума за одной испытание;
- записать выражение для вероятности P_N непадания в ε -окрестность экстремума за N испытаний;
- из выражения для P_N определить необходимое число испытаний N в зависимости от заданных $P_N = P$ и ε .

2. При аналогичных исходных условиях осуществить поиск минимума $f(x)$, модулированной сигналом $\sin 5x$, т.е. мультимодальной функции $f(x) \cdot \sin 5x$

3. Ход работы

Для наглядности построим графики данных функций (см. рисунки 1, 2).

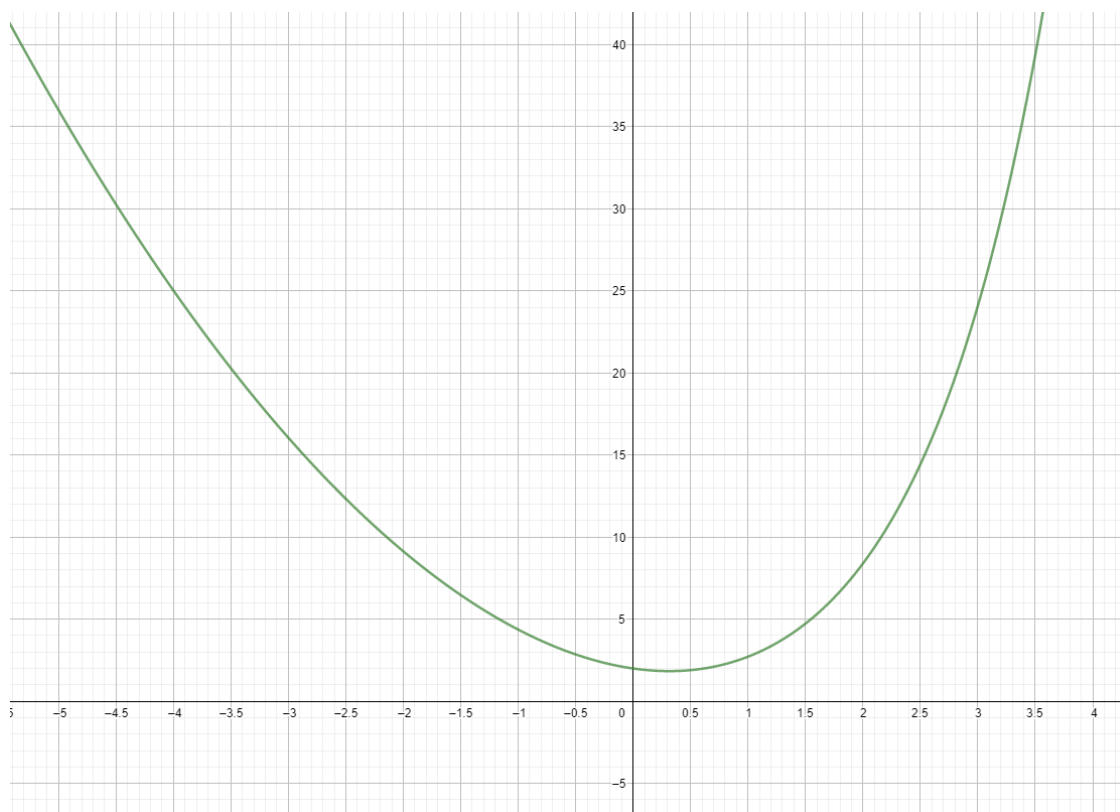


Рисунок 1 – График функции $f(x) = (1 - x)^2 + e^x$

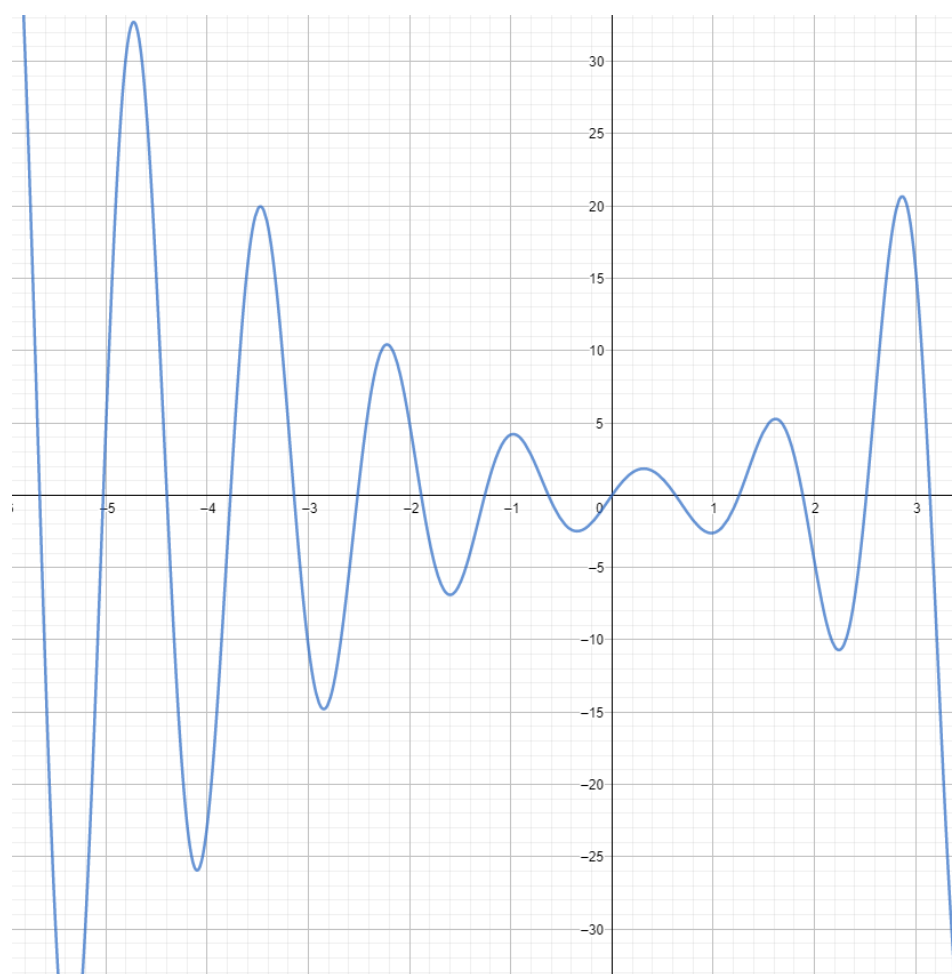


Рисунок 2 – График функции $f(x) \cdot \sin 5x$

Построим таблицу зависимости N от p и q .

Зависимость N от p и q										
$q \backslash p$	0.90	0.91	0.92	0.93	0.94	0.95	0.96	0.97	0.98	0.99
0.005	459	480	503	530	561	597	642	699	780	918
0.010	229	239	251	264	279	298	320	348	389	458
0.015	152	159	167	175	186	198	212	232	258	304
0.020	113	119	125	131	139	148	159	173	193	227
0.025	90	95	99	105	111	118	127	138	154	181
0.030	75	79	82	87	92	98	105	115	128	151
0.035	64	67	70	74	78	84	90	98	109	129
0.040	56	58	61	65	68	73	78	85	95	112
0.045	50	52	54	57	61	65	69	76	84	100
0.050	44	46	49	51	54	58	62	68	76	89
0.055	40	42	44	47	49	52	56	61	69	81
0.060	37	38	40	42	45	48	52	56	63	74
0.065	34	35	37	39	41	44	47	52	58	68
0.070	31	33	34	36	38	41	44	48	53	63
0.075	29	30	32	34	36	38	41	44	50	59
0.080	27	28	30	31	33	35	38	42	46	55
0.085	25	27	28	29	31	33	36	39	44	51
0.090	24	25	26	28	29	31	34	37	41	48
0.095	23	24	25	26	28	30	32	35	39	46
0.100	21	22	23	25	26	28	30	33	37	43

Рисунок 3 – Таблица зависимости N от p и q

Рассчитаем минимумы данных функций, используя полученные ранее данные.

Результаты расчёта приведены на рисунках 4, 5.

Точки минимума функции в зависимости от p и q											
q\p	0.90	0.91	0.92	0.93	0.94	0.95	0.96	0.97	0.98	0.99	
0.005	1.8396	1.8395	1.8395	1.8407	1.8395	1.8395	1.8395	1.8395	1.8395	1.8395	1.8395
0.010	1.8396	1.8397	1.8395	1.8398	1.8397	1.8396	1.8396	1.8423	1.8397	1.8418	
0.015	1.8405	1.8410	1.8433	1.8395	1.8395	1.8403	1.8395	1.8418	1.8395	1.8395	
0.020	1.8397	1.8401	1.8399	1.8459	1.8397	1.8411	1.8412	1.8398	1.8395	1.8397	
0.025	1.8486	1.8490	1.8403	1.8401	1.8408	1.8395	1.8403	1.8396	1.8395	1.8404	
0.030	1.8403	1.8408	1.8395	1.8404	1.8637	1.8427	1.8406	1.8446	1.8408	1.8399	
0.035	1.8533	1.8419	1.9647	1.8395	1.8395	1.8532	1.8396	1.8397	1.8447	1.8401	
0.040	1.8471	1.8575	1.8411	1.8794	1.8407	1.8442	1.8776	1.8423	1.8443	1.8464	
0.045	1.8550	1.8397	1.8641	1.8566	1.8414	1.8561	1.8857	1.8570	1.8395	1.8426	
0.050	1.8463	1.8551	1.8397	1.8397	1.8504	1.8588	1.8412	1.8537	1.8642	1.8442	
0.055	2.3228	1.8396	1.8398	1.8422	1.8761	1.8410	1.8417	1.8563	1.8601	1.8411	
0.060	1.8523	1.9809	1.8403	1.8442	1.8686	1.8429	1.8793	1.8451	1.8417	1.8720	
0.065	1.8573	1.8439	2.0865	1.8428	1.8773	1.8407	1.8397	1.8397	1.8421	1.8397	
0.070	1.8424	1.8405	1.8464	1.8397	1.8426	1.8440	1.8405	1.8457	1.8520	1.8649	
0.075	1.8462	1.8404	1.8840	1.8402	1.8406	1.8413	1.8804	1.8432	1.8415	1.8634	
0.080	1.8495	1.8414	1.8745	1.8499	1.8860	1.8554	1.8456	1.8407	1.8497	1.8395	
0.085	1.8411	1.8398	1.8437	1.8405	1.8434	1.8642	1.8492	1.8425	1.8398	1.8458	
0.090	1.8849	2.0152	1.8395	1.8669	1.8457	1.8427	1.8960	1.8802	1.8395	1.8481	
0.095	1.8466	1.8427	1.8717	1.8634	1.9201	1.8848	1.8576	1.8452	1.8520	1.8395	
0.100	1.8913	1.8967	1.8402	1.9518	1.8645	1.8399	1.9265	1.8421	1.8693	1.8433	

Рисунок 4 – Результаты расчёта для функции $f(x) = (1 - x)^2 + e^x$

Точки минимума функции в зависимости от p и q											
q\p	0.90	0.91	0.92	0.93	0.94	0.95	0.96	0.97	0.98	0.99	
0.005	-25.9438	-25.9355	-25.9429	-25.9214	-25.9434	-25.9332	-25.9413	-25.9437	-25.9361	-25.9430	
0.010	-25.9025	-25.9439	-25.0093	-25.8282	-25.9401	-25.3693	-25.6039	-25.9441	-25.8598	-25.9441	
0.015	-25.8975	-25.9268	-25.9382	-25.9374	-25.7929	-25.9419	-25.8860	-25.9132	-25.9032	-25.7390	
0.020	-25.9258	-25.8241	-25.9423	-25.6557	-25.9198	-25.8882	-25.6101	-25.9395	-25.9239	-25.9191	
0.025	-25.9434	-25.4686	-25.8700	-25.1976	-25.7907	-25.2417	-25.7807	-25.9346	-25.8727	-25.8819	
0.030	-25.9408	-25.9167	-25.6775	-23.9982	-25.7385	-14.7586	-25.8830	-25.7186	-23.8133	-25.9350	
0.035	-24.8013	-25.8212	-23.0877	-25.8537	-24.6637	-25.9379	-25.9252	-24.7514	-25.8870	-25.6507	
0.040	-25.8981	-25.7873	-25.8452	-25.9194	-25.8972	-25.9202	-25.9153	-25.9441	-25.9039	-25.6325	
0.045	-14.7811	-25.5421	-25.8903	-25.5404	-25.1364	-24.8158	-25.8425	-25.8806	-25.8789	-25.9212	
0.050	-25.9269	-25.8090	-23.9510	-25.9199	-25.7944	-25.9108	-22.9600	-25.8939	-25.9159	-25.9361	
0.055	-22.0953	-25.8271	-25.8272	-21.1053	-24.7906	-25.9304	-25.9168	-25.8405	-25.6983	-25.9384	
0.060	-12.1766	-21.9269	-19.9327	-20.1980	-22.4952	-20.5651	-25.9440	-25.7119	-25.7392	-24.9669	
0.065	-13.3129	-22.4741	-22.9583	-24.3897	-19.1908	-24.5950	-25.5871	-24.3761	-25.7848	-23.4485	
0.070	-25.7869	-25.1626	-16.3741	-14.7331	-14.4939	-21.7244	-24.1691	-13.8884	-24.5762	-25.9416	
0.075	-25.4085	-5.8678	-14.4839	-20.4053	-25.8348	-25.8715	-21.8276	-20.3127	-22.3046	-25.9090	
0.080	-23.2182	-14.0208	-15.9603	-25.5004	-14.4606	-25.8442	-22.8627	-25.3241	-25.3903	-25.7907	
0.085	-13.0581	-22.0034	-6.7316	-25.9330	-25.8544	-22.4486	-20.7804	-22.9228	-25.9397	-20.1039	
0.090	-23.1638	-21.6795	-25.0204	-17.1200	-25.5529	-21.7188	-25.5700	-10.8382	-14.3583	-21.0052	
0.095	-25.5003	-25.9439	-14.5632	-25.9355	-16.7039	-25.8758	-25.9389	-24.2863	-25.5425	-24.5453	
0.100	-23.5097	-14.7818	-25.9402	-20.5874	-25.2032	-22.0860	-23.7375	-25.9424	-25.5800	-22.1681	

Рисунок 5 – Результаты расчёта для функции $f(x) \cdot \sin 5x$

Ссылка на репозиторий с выполненной работой: <https://github.com/shreddered/lab-02>

4. Выводы

Из полученных таблиц и графиков видно, 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. Исходный код программы

Файл source/algorithms/searcher.d

```
module algorithms.searcher;

interface ISearcher {
    public void setInterval(in double a, in double b);
    public void search(const double eps = double.nan);
}
```

Файл source/algorithms/package.d

```
module algorithms;  
  
public import algorithms.fibonacci;  
public import algorithms.random_searcher;
```

Файл source/algorithms/random_searcher.d

```
module algorithms.random_searcher;

import algorithms.searcher;

import std.algorithm : each, map, min, reduce;
import std.array : array;
import std.math : log;
import std.random : uniform;
import std.range : generate, iota, takeExactly;
import std.stdio : write, writef, writefln, writeln;

class RandomSearcher(alias func) : ISearcher {
    private double _a, _b;

    public override void setInterval(in double a, in double b) {
        _a = a;
        _b = b;
    }

    private void print(in double[] p, in double[] q, in double[][] mins) {
        // table header
        writeln("Точки минимума функции в зависимости от p и q");
        writeln("+-----+-----+-----+-----+-----",
            "+-----+-----+-----+-----+-----",
            "+-----+");
        write("| q\\P ");
        p.each!((a) => writef("| %-2.2f ", a));
        writeln('|');
        writeln("+-----+-----+-----+-----+-----",
            "+-----+-----+-----+-----+-----",
            "+-----+");

        foreach(i, seq; mins) {
            writef("| %-1.3f ", q[i]);
            foreach(j, elem; seq) {
                writef("|%9,2.4f", elem);
            }
            writeln('|');
        }
        // table bottom
        writeln("+-----+-----+-----+-----+-----",
            "+-----+-----+-----+-----+-----",
            "+-----+");
    }

    private void printN(in double[] p, in double[] q, in ulong[][] n) {
        writeln("Зависимость N от p и q");
        // table header
        writeln("+-----+-----+-----+-----+-----",
            "+-----+-----+-----+-----+-----",
            "+-----+");
        write("| q\\P ");
        p.each!((a) => writef("| %-2.2f ", a));
        writeln('|');
        writeln("+-----+-----+-----+-----+-----",
            "+-----+-----+-----+-----+-----",
            "+-----+");

        foreach(i, seq; n) {
            writef("| %-1.3f ", q[i]);
            foreach(j, elem; seq) {
                writef("|%9d", elem);
            }
        }
    }
}
```

```

    }
    writeln('|');
}

// table bottom
writeln("+-----+-----+-----+-----+",
        "+-----+-----+-----+-----+",
        "+-----+");
}

public override void search(const double eps) {
    enum double[] p = iota(0.9, 1.0, 0.01).array;
    enum double[] q = iota(0.005, 0.105, 0.005).array;

    // computes 2D range (matrix) n = log(1 - p)/log(1 - q) at compile
time
    // first mapping maps every q to another mapping
    // which maps every p to log(1 - p)/log(1 - q)
    enum ulong[][] n = q.map!((q_) => p
        .map!((p_) => cast(ulong) (log(1 - p_)/log(1 - q_)))
        .array)
        .array;
    printN(p, q, n);
    writeln;
    // helper function for generating n random numbers
    alias generateNRandomNumbers = (ulong _n) => generate(() =>
uniform!"[]"(_a, _b)).takeExactly(_n);

    // 2D range of minimums
    double[][] mins = n.map!((str) => str
        // generate n random numbers in [a, b] for each n
        .map!generateNRandomNumbers
        // for each sequence select function's minimum
        .map!((seq) => seq.map!func // function values
            .reduce!min)
        .array)
        .array;
    print(p, q, mins);
}
}

```