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 № 3
по дисциплине «Теория систем и системный анализ»

Тема: «Исследование алгоритма имитации отжига**»**

Вариант 10

Выполнила: Минькова А.А.,
студентка группы ИУ8-31

Проверила: Коннова Н.С.,
доцент каф. ИУ8

г. Москва,
2020 г.

Цель работы

Изучение метода имитации отжига для поиска экстремума на примере унимодальной и мультимодальной функций одного переменного.

Условие задачи

На интервале [-2; 4] задана унимодальная функция одного переменного f(x) = (1-x)^2+exp(x). Используя метод имитации отжига осуществить поиск минимума f(x). При аналогичных исходных условиях осуществить поиск минимума f(x), модулированной сигналом sin(5x), т.е. мультимодальной функции f(x)*sin(5x).

Графики заданных функций

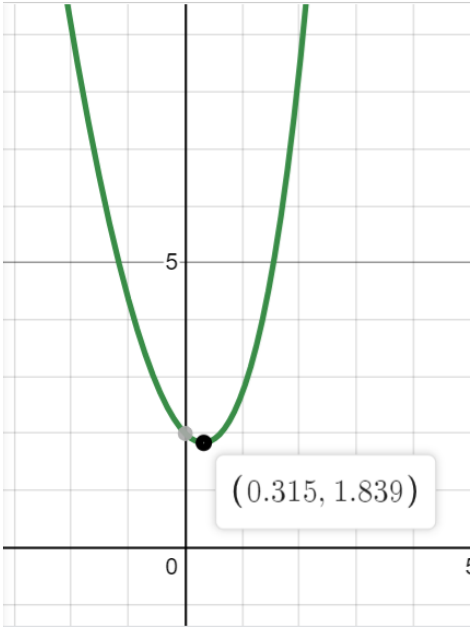

Рисунок 1- график f(x)

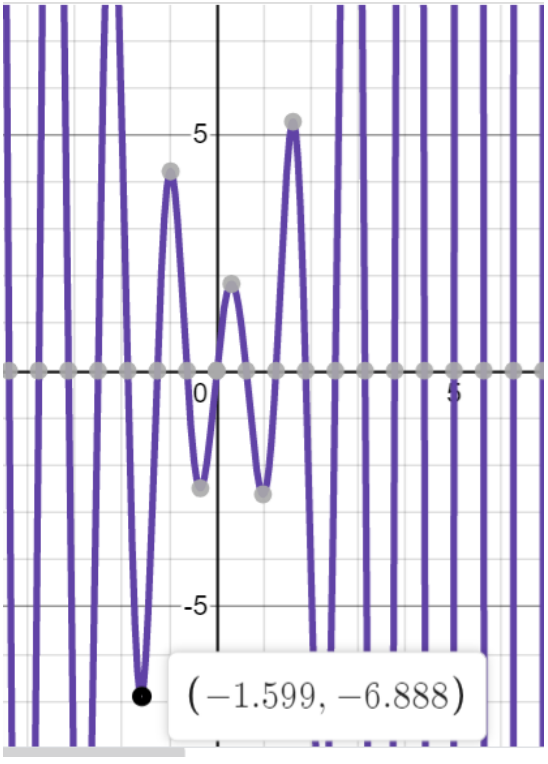

Рисунок 2- график f(x)*sin(5x)

Результат работы программы

```
Variant 10
Function 1:
-----------------------------------------------
| N   | T         | x          | f(x)         |
-----------------------------------------------
| 1   | 10000     | -0.238306  | 2.32136      |
| 2   | 9500      | 2.79206    | 19.5261      |
| 3   | 9025      | -1.59247   | 6.92433      |
| 4   | 8573.75   | 1.3356     | 3.91493      |
| 5   | 8145.06   | 1.51203    | 4.79809      |
| 6   | 7737.81   | 2.1204     | 9.58976      |
| 7   | 7350.92   | 3.21926    | 29.9348      |
| 8   | 6983.37   | -0.357284  | 2.54179      |
| 9   | 6634.2    | 1.12745    | 3.10401      |
| 10  | 6302.49   | -1.94224   | 8.80019      |
| 11  | 5987.37   | 0.192163   | 1.86447      |
| 12  | 5688      | 1.67697    | 5.80759      |
| 13  | 5403.6    | -0.774745  | 3.61054      |
| 14  | 5133.42   | -1.14067   | 4.90207      |
| 15  | 4876.75   | 2.72282    | 18.1913      |
| 16  | 4632.91   | 1.98306    | 8.23139      |
| 17  | 4401.27   | -0.631244  | 3.19289      |
| 18  | 4181.2    | -1.32211   | 5.65876      |
| 19  | 3972.14   | 3.76078    | 50.6037      |
| 20  | 3773.54   | -1.36275   | 5.83853      |
| 21  | 3584.86   | -1.70502   | 7.49889      |
| 22  | 3405.62   | 1.99858    | 8.37575      |
| 23  | 3235.34   | 3.93936    | 60.0255      |
| 24  | 3073.57   | 0.184764   | 1.86754      |
| 25  | 2919.89   | 3.32532    | 33.215       |
```

243	0.0406549	0.0667769	1.93996
244	0.0386221	0.0667769	1.93996
245	0.036691	0.0667769	1.93996
246	0.0348565	0.0667769	1.93996
247	0.0331136	0.0667769	1.93996
248	0.031458	0.0667769	1.93996
249	0.0298851	0.0667769	1.93996
250	0.0283908	0.0667769	1.93996
251	0.0269713	0.217037	1.85542
252	0.0256227	0.217037	1.85542
253	0.0243416	0.420486	1.85854
254	0.0231245	0.420486	1.85854
255	0.0219683	0.420486	1.85854
256	0.0208699	0.420486	1.85854
257	0.0198264	0.420486	1.85854
258	0.018835	0.420486	1.85854
259	0.0178933	0.420486	1.85854
260	0.0169986	0.420486	1.85854
261	0.0161487	0.420486	1.85854
262	0.0153413	0.420486	1.85854
263	0.0145742	0.420486	1.85854
264	0.0138455	0.420486	1.85854
265	0.0131532	0.420486	1.85854
266	0.0124956	0.420486	1.85854
267	0.0118708	0.420486	1.85854
268	0.0112772	0.420486	1.85854
269	0.0107134	0.420486	1.85854
270	0.0101777	0.420486	1.85854

Result: Xmin = 0.420486, Fmin = 1.85854

Function 2:

N	T	x	f(x)
1	10000	-0.229392	-2.10235
2	9500	-1.73168	-5.29782
3	9025	1.58363	5.2025
4	8573.75	-1.60705	-6.88254
5	8145.06	2.29829	-10.2403
6	7737.81	-0.615278	-0.205213
7	7350.92	-1.83812	-1.90611
8	6983.37	0.756802	-1.31246
9	6634.2	3.12718	1.96804
10	6302.49	2.17361	-10.0847
11	5987.37	3.78861	4.85108
12	5688	2.73225	16.3271
13	5403.6	0.761473	-1.35783
14	5133.42	0.0709889	0.673053
15	4876.75	-1.40972	-4.19245
16	4632.91	0.754432	-1.28928
17	4401.27	-0.164419	-1.61478
18	4181.2	2.61649	8.04364
19	3972.14	2.70522	14.6323
20	3773.54	2.42947	-5.45063
21	3584.86	2.0551	-6.70624
22	3405.62	-1.96867	3.63865
23	3235.34	3.19734	-8.06003
24	3073.57	1.64614	5.21141
25	2919.89	3.64843	-25.9284
26	2773.9	-0.896143	3.89654

245	0.036691	3.48757	-38.4032
246	0.0348565	3.48978	-38.4149
247	0.0331136	3.48978	-38.4149
248	0.031458	3.48978	-38.4149
249	0.0298851	3.48978	-38.4149
250	0.0283908	3.48978	-38.4149
251	0.0269713	3.48978	-38.4149
252	0.0256227	3.48978	-38.4149
253	0.0243416	3.48978	-38.4149
254	0.0231245	3.48978	-38.4149
255	0.0219683	3.48978	-38.4149
256	0.0208699	3.48978	-38.4149
257	0.0198264	3.48978	-38.4149
258	0.018835	3.48978	-38.4149
259	0.0178933	3.48978	-38.4149
260	0.0169986	3.48978	-38.4149
261	0.0161487	3.48978	-38.4149
262	0.0153413	3.48978	-38.4149
263	0.0145742	3.48978	-38.4149
264	0.0138455	3.48978	-38.4149
265	0.0131532	3.48978	-38.4149
266	0.0124956	3.48978	-38.4149
267	0.0118708	3.48978	-38.4149
268	0.0112772	3.48978	-38.4149
269	0.0107134	3.48978	-38.4149
270	0.0101777	3.48978	-38.4149

Result: Xmin = 3.48978, Fmin = -38.4149

Выводы

В результате проделанной работы был исследован метод имитации отжига. Он является эффективным алгоритмом случайного поиска глобального минимума, в чем мы убедились на примере данной унимодальной и мультимодальной функции одного переменного. Метод эффективно работает для обеих функций.

Ответ на контрольный вопрос

В чем состоит сущность метода имитации отжига? Какова область применимости данного метода?

Метод имитации отжига основан на том, что локальное (субоптимальное) решение, найденное в процессе решения задачи оптимизации, также можно рассматривать как дефектное решение. Улучшить это решение (приблизиться к глобальному оптимуму) можно путём его случайных флюктуаций, амплитуда которых уменьшается с ростом номера итераций. Принципиальным в алгоритме SA является то, что, в отличие от большинства других стохастических алгоритмов поисковой оптимизации, он допускает шаги, приводящие к увеличению значений фитнес-функции. Метод имитации отжига применяется для решения разных оптимизационных задач – финансовых, компьютерной графики, комбинаторных, и т.д., используется в нейронных сетях.

Приложение. Исходный код программы

```
#include
<cmath>
            #include <ctime>
            #include <iomanip>
            #include <iostream>

            using std::cout;

            double randomInRange(const double a, const double b)
            {
                return a + rand() * 1./RAND_MAX * (b - a);
            }

            void printTableHead()
            {
                cout << std::left << std::string(47, '-') << '\n'
                     << "| " << std::setw(4) << "N"
                     << "| " << std::setw(10) << "T"
                     << "| " << std::setw(11) << "x"
                     << "| " << std::setw(13) << "f(x)" << "|\n"
                     << std::string(47, '-') << '\n';
            }

            void printLine(const int iteration, const double T, const double value, const double
            functionValue)
            {
                cout << "| " << std::setw(4) << iteration
                     << "| " << std::setw(10) << T
                     << "| " << std::setw(11) << value
                     << "| " << std::setw(13) << functionValue << "|\n";
            }

            template<class Function>
            auto SAMethod(const double a, const double b, Function func)
            {
                printTableHead();

                const double T_min = .01;
                double T_i = 10000.;
                double x_i = randomInRange(a,b);
                int i = 0;
                while (T_i > T_min) {
                    ++i;
                    double x_new = randomInRange(a, b);
                    double delta_f = func(x_new) - func(x_i);
                    if (delta_f <= 0) {
```

```
            x_i = x_new;
        } else {
            double randomProb = randomInRange(0, 1);
            double probability = exp(-delta_f/T_i);
            if (randomProb < probability) {
                x_i = x_new;
            }
        }
        printLine(i, T_i, x_i, func(x_i));
        T_i *= .95;
    }

    cout << std::string(47, '-') << '\n';
    return std::pair{x_i, func(x_i)};
}

double MyFunction(const double x)
{
    return pow((1-x),2)+exp(x);
}

double MyFunctionSin(const double x) {
    return MyFunction(x) * sin(5*x);
}

const double A = -2.;
const double B = 4.;

int main()
{
    cout << "Variant 10\nFunction 1:" <<std::endl;
    srand(time(nullptr));
    auto result_1 = SAMethod(A, B, MyFunction);
    cout << "Result: Xmin = " << result_1.first
         << ", Fmin = " << result_1.second << '\n';

    cout << "\nFunction 2:"<<std::endl;
    auto result_2 = SAMethod(A, B, MyFunctionSin);
    cout << "Result: Xmin = " << result_2.first
         << ", Fmin = " << result_2.second << '\n';

    return 0;
}
```