



**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 «Методы оптимизации»

Тема: «Исследование алгоритма имитации отжига»

Вариант 15

Выполнил: Петросян А.Р.,
студент группы ИУ8-32

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

г. Москва,
2020 г.

1. Цель работы

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

2. Постановка задачи

1. На интервале $[a,b]$ задана унимодальная функция одного переменного $f(x)$. Используя метод имитации отжига осуществить поиск минимума $f(x)$.

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

$$f(x) = x^2 \sin(x)$$

$$a = 9$$

$$b = 12$$

3. Ход работы

Рисунок 1 демонстрирует графики $f(x)$ унимодальной функции и мультимодальной функции $f(x) \cdot \sin 5x$.

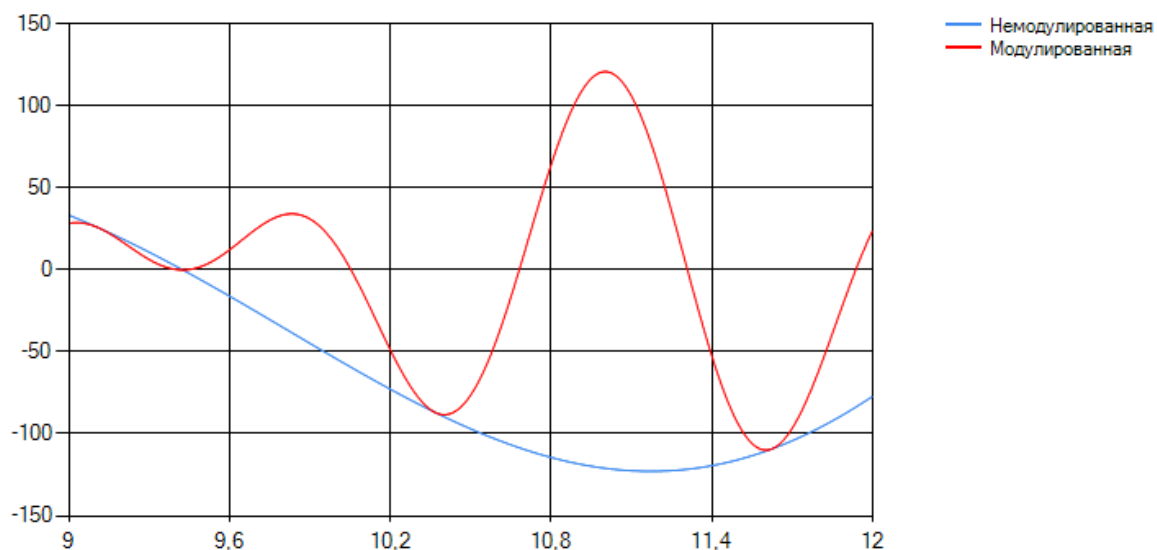


Рисунок 1 – Графики функций

На графике видно, что единственный минимум немодулированной функции находится около значения аргумента 11.2, а абсолютный минимум модулированной функции – около 11.5

Результаты численного эксперимента для представим в таблице 1, а для мультимодальной функции – в табл. 2.

Табл. 1. Результаты поиска экстремума $f(x)$

| N | T | x | f(x) |
|---|------|----------|------------|
| 1 | 10 | 10,5 | -96,98646 |
| 2 | 9,75 | 11,86854 | -90,51216 |
| 3 | 9,51 | 10,83299 | -115,80613 |
| 4 | 9,27 | 10,94289 | -119,58057 |
| 5 | 8,59 | 11,63709 | -108,49877 |
| 6 | 8,17 | 11,8437 | -92,77568 |

| | | | |
|----|------|----------|------------|
| 7 | 7,96 | 11,74765 | -100,78267 |
| 8 | 7,76 | 10,91928 | -118,88387 |
| 9 | 7,57 | 10,96984 | -120,29763 |
| 10 | 7,2 | 11,4823 | -116,53224 |
| 11 | 7,02 | 11,174 | -122,87606 |
| 12 | 6,67 | 11,096 | -122,50089 |
| 13 | 6,5 | 11,03689 | -121,70891 |
| 14 | 6,34 | 11,12214 | -122,71252 |
| 15 | 5,18 | 11,13015 | -122,76015 |
| 16 | 3,92 | 11,3963 | -119,58666 |
| 17 | 3,54 | 11,22571 | -122,6942 |
| 18 | 2,61 | 11,2464 | -122,52361 |
| 19 | 2,08 | 11,10554 | -122,58809 |
| 20 | 1,7 | 11,12467 | -122,72843 |
| 21 | 1,22 | 10,97087 | -120,32325 |
| 22 | 1,19 | 11,24543 | -122,53285 |
| 23 | 1,13 | 11,09929 | -122,53221 |
| 24 | 0,76 | 11,10039 | -122,54238 |
| 25 | 0,59 | 11,14879 | -122,83946 |
| 26 | 0,38 | 11,14138 | -122,8132 |
| 27 | 0,3 | 11,05349 | -121,97476 |
| 28 | 0,26 | 11,17467 | -122,87593 |
| 29 | 0,24 | 11,13832 | -122,80036 |
| 30 | 0,2 | 11,14616 | -122,83095 |
| 31 | 0,03 | 11,17323 | -122,87616 |
| 32 | 0 | 11,17164 | -122,8761 |

Можно увидеть, что результат поиска совпадает с ожидаемым

Табл. 2. Результаты поиска экстремума $f(x) \cdot \sin 5x$

| N | T | x | f(x) |
|----|------|----------|------------|
| 1 | 10,0 | 10,50000 | -76,39674 |
| 2 | 9,27 | 11,73736 | -85,65767 |
| 3 | 8,38 | 11,47352 | -85,37838 |
| 4 | 8,17 | 10,42553 | -87,64828 |
| 5 | 7,38 | 10,46009 | -84,14160 |
| 6 | 4,45 | 11,50066 | -94,46307 |
| 7 | 3,63 | 11,55849 | -106,99693 |
| 8 | 0,35 | 11,64357 | -107,56873 |
| 9 | 0,21 | 11,58012 | -109,16600 |
| 10 | 0,03 | 11,58293 | -109,34545 |
| 11 | 0,02 | 11,61594 | -109,70129 |
| 12 | 0,00 | 11,59294 | -109,79341 |
| 13 | 0,00 | 11,60894 | -109,89363 |

Можно увидеть, что результат поиска также совпадает с ожидаемым

Рисунок 1 демонстрирует зависимости погрешности решения при поиске минимума немодулированной функции от числа точек N (номера итерации).

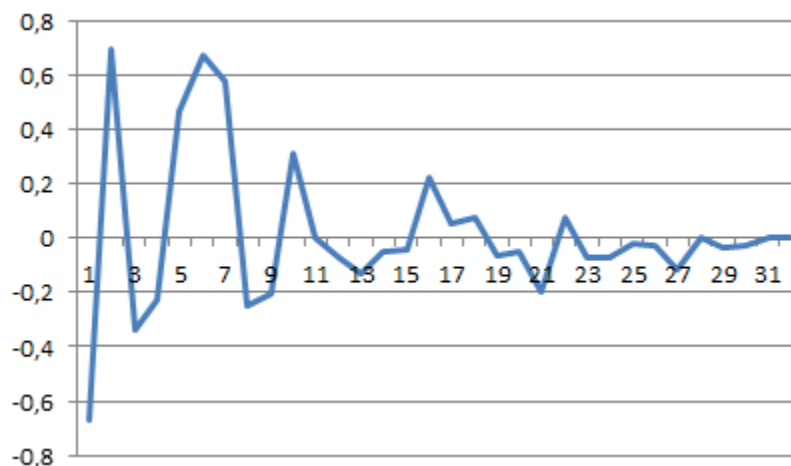


Рисунок 1 – График зависимостей погрешности от числа точек N

По графику можно заметить, что величина отклонения найденной точки экстремума от результата постепенно уменьшается с увеличением номера итерации, а также что для получения результата метод имитации отжига требует не очень большого количества шагов.

4. Выводы

В данной лабораторной работе были найдены минимумы двух унимодальных функций с помощью метода имитации отжига. Из приведенного выше хода работы можно сделать вывод, что для достижения заданной погрешности методу требуется не очень большое количество шагов, а также что находится именно точка абсолютного минимума на заданном отрезке. Преимущество используемого метода именно в том, что с большой вероятностью будет найдена именно точка абсолютного минимума.

Приложение 1. Исходный код программы

```
#pragma endregion

//Делегат для передачи функции
delegate double function(double x);
//Не модулированная функция
double F0(double x)
{
    return x*x*Math::Sin(x);
}
//Модулированная
double F(double x)
{
    return F0(x) * Math::Sin(5*x);
}

double A, B; //Интервал
ref struct Log //Единичная запись в журнале
{
    double T; //Температура
    double X; //Аргумент
    double F; //Функция
};
List<Log^>^ list; //Журнал операций

private: System::Void MyForm_Load(System::Object^ sender, System::EventArgs^ e) {
    list = gcnew List<Log^>();
    button_Click(sender, e); //Вызвать расчет для данных по умолчанию
}

private: System::Void button_Click(System::Object^ sender, System::EventArgs^ e) {
    //Прочитать данные
    if (!ReadData()) return; //В данных - ошибки
    //Заполнить график
    ShowInChart();
    //Выполнить расчет
    //function^ f = gcnew function(this, &MyForm::F0);
    double X = FindExtremum(gcnew function(this, &MyForm::F0));
    labelMinF0->Text = X.ToString("0.000");
    SaveToFile("F0.txt");
    X = FindExtremum(gcnew function(this, &MyForm::F));
    labelMinF->Text = X.ToString("0.000");
    //Вывести результат в файл
    SaveToFile("F.txt");
}

//Прочитать одно значение
// ^ = ссылка на экземпляр класса
// % = передача параметра по ссылке (может быть изменен в функции)
bool ReadTextBox(TextBox^ textbox, double % value)
{
    textbox->BackColor = Color::Red;
    if (!double::TryParse(textbox->Text, value)) return false;
    textbox->BackColor = Color::White;
    return true;
}
//Прочитать исходные данные
bool ReadData()
{
    bool result = true;
```

```

        result &= ReadTextBox(textBoxA, A);
        result &= ReadTextBox(textBoxB, B);
        if (A > B)
        {
            MessageBox::Show("A>B");
            return false;
        }
        return result;
    }

void ShowInChart()
{
    int N = 1000; //Точек на графике
    double h = (B - A) / (N - 1);
    chart->Series[0]->Points->Clear();
    chart->Series[1]->Points->Clear();
    chart->ChartAreas[0]->AxisX->Minimum = A;
    chart->ChartAreas[0]->AxisX->Maximum = B;

    for (int i = 0; i <= N; i++)
    {
        double x = A + h*i;
        chart->Series[0]->Points->AddXY(x, F0(x));
        chart->Series[1]->Points->AddXY(x, F(x));
    }
}

double TMax = 10; //Начальная "температура"
double TMin = 1E-6; //Конечная температура
Random^ random = gcnew Random(); //Датчик случайных чисел

```

```

double FindExtremum(function^ F)
{
    double T = TMax; //Номер итерации
    double Alpha = 0.975; //Коэффициент уменьшения
    double X = (B + A) / 2; //Начальное значение приближения
    double f1 = F(X); //Начальное значение функции
    bool NeedToLog = true;
    while (T > TMin)
    {
        if (NeedToLog)
        {
            //Записывается в журнал
            Log^ log = gcnew Log();
            log->T = T;
            log->X = X;
            log->F = f1;
            list->Add(log);
            NeedToLog = false;
        }
        //Выбирается пробная точка
        double xs = random->NextDouble()*(B - A) + A;

        //Улучшение алгоритма - уменьшение величины шага с понижением
        //температуры
        double xs = X + (2*random->NextDouble()-1) * (B - A)*(T / TMax);
        //if (xs<A || xs>B) continue;

        T = T * Alpha; //Следующая итерация
        double f = f1; //Значение функции на предыдущем шаге

        f1 = F(xs);
        //Разница
        double df = f1 - f;
    }
}

```


точку

```
        if (df <= 0) //Если разница отрицательная - безусловный переход в новую
        {
            X = xs;
            NeedToLog = true;
            continue; //и продолжить выполнение цикла
        }
        //Если разница положительная
        double P = Math::Exp(-df / T); //Вероятность перехода
        if (random->NextDouble() <= P)
        {
            X = xs; //Переход
            NeedToLog = true;
        }
        else
            f1 = f; //Иначе - остаемся на месте
    }
    return X;
}

//Сохранить журнал в файл, очистить
void SaveToFile(String^ filename)
{
    StreamWriter^ writer = gcnew StreamWriter(filename);
    writer->WriteLine("N\tT\tx\tf(x)\t"); //Заголовок
    for (int k = 0; k < list->Count; k++)
    {
        Log^ log = list[k];
        writer->WriteLine(
            (k+1).ToString()+"\t"+
            log->T.ToString("0.00")+"\t" +
            log->X.ToString("0.00000") + "\t" +
            log->F.ToString("0.00000") + "\t"
        );
    }

    writer->Close();
    list->Clear();
}

};
```