



**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 6

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

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

1. Цель работы

Изучение алгоритма имитации отжига на примере унимодальной и мультимодальной функций одного переменного.

2. Условие задачи

На интервале $[-5, 2]$ задана унимодальная функция одного переменного $f(x) = (1 - x)^2 + e^x$. 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 $f(x) \cdot \sin 5x$.

3. Ход работы

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

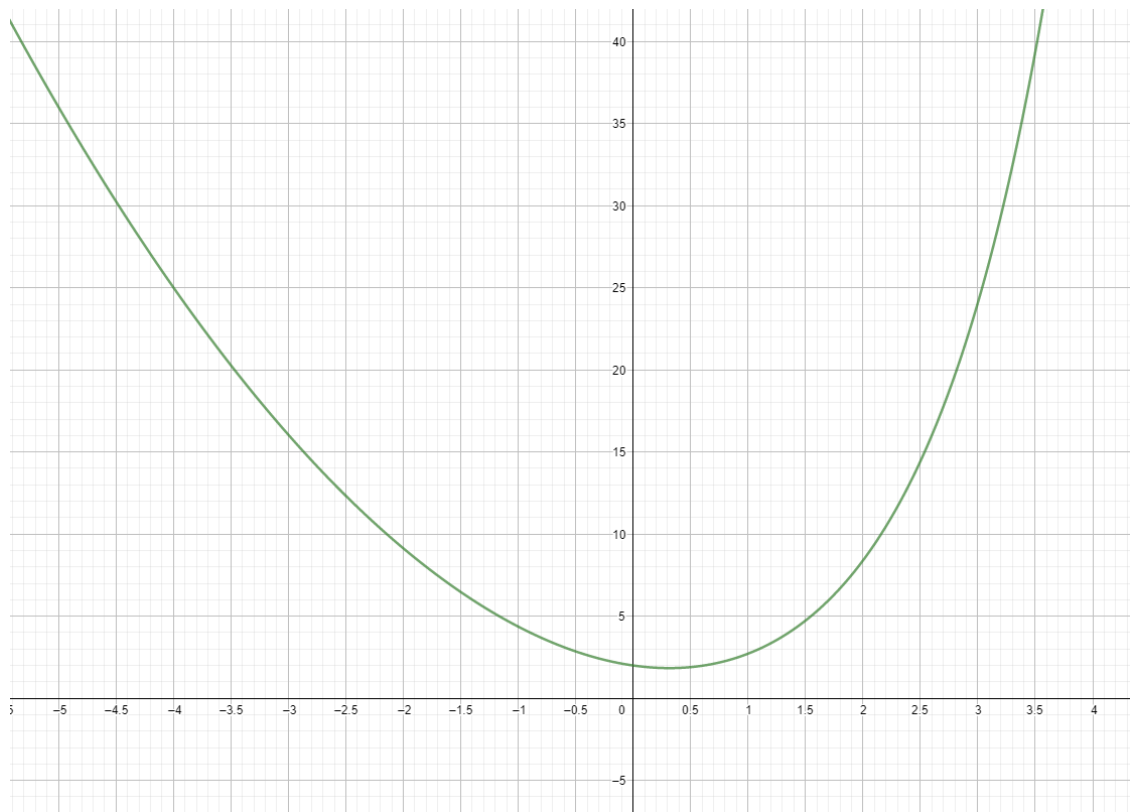


Рисунок 1 – График функции $f(x) = (1 - x)^2 + e^x$

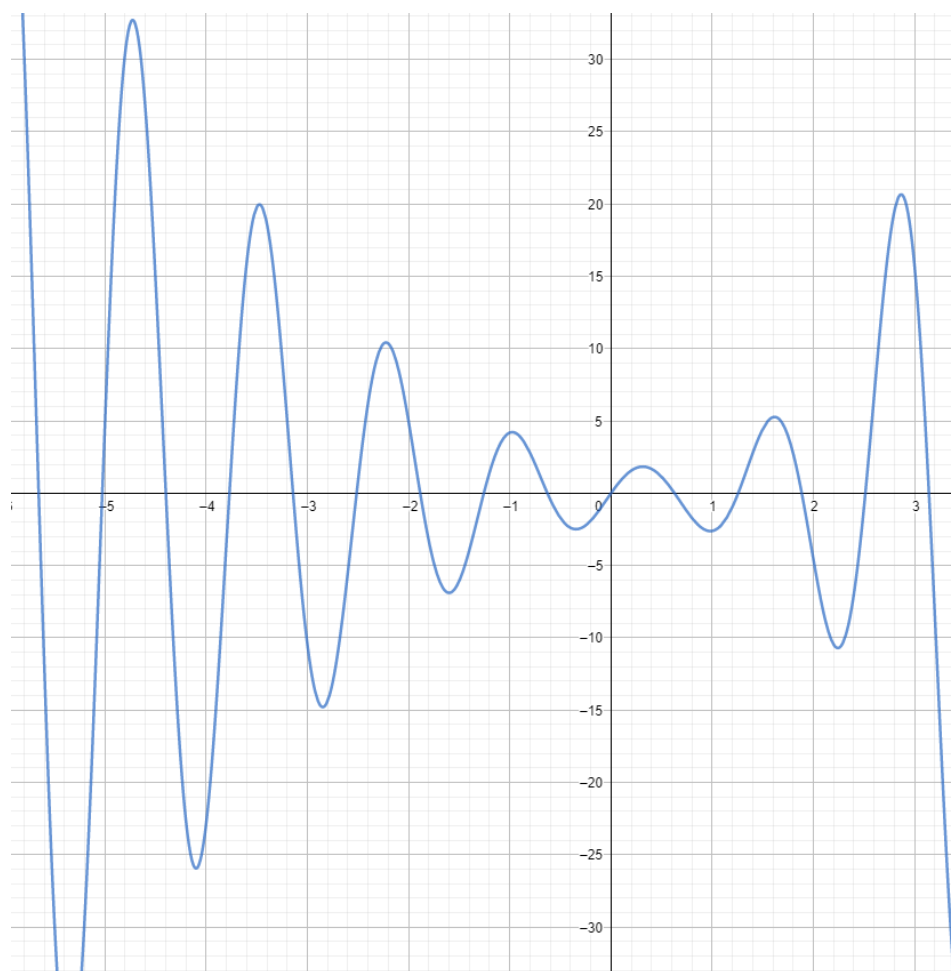


Рисунок 2 – График функции $f(x) \cdot \sin 5x$

Результаты поиска минимума алгоритмом имитации отжига для функций $f(x)$ и $f(x) \cdot \sin 5x$ представлены в таблицах 1 и 2 соответственно.

Таблица 1 – Алгоритм имитации отжига для функции $f(x) = (1 - x)^2 + e^x$

	N	T	x	f(x)
	0	1000.00000	-3.302	18.548
	1	950.00000	-0.859	3.881
	2	902.50000	1.101	3.018
	3	857.37500	-0.408	2.647
	4	814.50625	-0.017	2.018
	5	773.78094	-0.403	2.638
	6	735.09189	-1.165	5.001
	7	698.33730	-2.670	13.537
	8	663.42043	-3.446	19.803
	9	630.24941	-4.753	33.102
	10	598.73694	-4.067	25.691
	11	568.80009	-0.968	4.255
	12	540.36009	-1.758	7.778
	13	513.34208	-1.331	5.699
	14	487.67498	-4.727	32.812
	15	463.29123	-0.099	2.113
	16	440.12667	-3.885	23.880
	17	418.12034	-0.712	3.423
	18	397.21432	-4.500	30.260

	19	377.35360	-2.590	12.966
	20	358.48592	1.399	4.212
	21	340.56163	-4.078	25.801
	22	323.53354	-2.045	9.400
	23	307.35687	-0.059	2.065
	24	291.98902	-0.987	4.322
	25	277.38957	0.651	2.040
	26	263.52009	0.647	2.034
	27	250.34409	-4.400	29.172
	28	237.82689	-3.608	21.263
	29	225.93554	-4.967	35.617
	30	214.63876	-1.717	7.560
	31	203.90683	0.568	1.952
	32	193.71148	-3.317	18.672
	33	184.02591	-1.101	4.749
	34	174.82461	0.877	2.419
	35	166.08338	-3.506	20.330
	36	157.77921	-4.491	30.158
	37	149.89025	-2.915	15.383
	38	142.39574	-2.950	15.651
	39	135.27595	-3.911	24.140
	40	128.51216	-4.111	26.135
	41	122.08655	1.721	6.112
	42	115.98222	-2.630	13.248
	43	110.18311	-3.279	18.345
	44	104.67395	-1.153	4.952
	45	99.44026	0.454	1.873
	46	94.46824	-1.798	7.996
	47	89.74483	1.624	5.462
	48	85.25759	-1.604	6.980
	49	80.99471	-3.347	18.931
	50	76.94498	-0.194	2.250
	51	73.09773	-3.994	24.956
	52	69.44284	-3.349	18.950
	53	65.97070	0.997	2.711
	54	62.67216	-0.242	2.327
	55	59.53856	-1.636	7.142
	56	56.56163	0.502	1.900
	57	53.73355	-0.333	2.494
	58	51.04687	0.275	1.842
	59	48.49453	-0.541	2.957
	60	46.06980	-0.541	2.957
	61	43.76631	0.339	1.840
	62	41.57799	-4.902	34.842
	63	39.49909	-0.029	2.030
	64	37.52414	-4.490	30.149
	65	35.64793	-0.461	2.764
	66	33.86554	-0.719	3.442
	67	32.17226	1.473	4.585
	68	30.56365	1.473	4.585
	69	29.03546	-1.518	6.558
	70	27.58369	-2.823	14.672
	71	26.20451	-2.522	12.484
	72	24.89428	-2.967	15.785
	73	23.64957	-1.806	8.038
	74	22.46709	-2.273	10.817
	75	21.34373	1.644	5.590
	76	20.27655	1.644	5.590
	77	19.26272	0.736	2.157
	78	18.29958	0.736	2.157
	79	17.38460	0.736	2.157
	80	16.51537	0.511	1.906
	81	15.68961	1.545	4.984
	82	14.90513	-0.909	4.049

	83	14.15987	-0.909	4.049
	84	13.45188	-0.721	3.449
	85	12.77928	-2.448	11.972
	86	12.14032	1.571	5.139
	87	11.53330	-2.750	14.126
	88	10.95664	-0.644	3.228
	89	10.40880	-0.644	3.228
	90	9.88836	-0.644	3.228
	91	9.39395	1.478	4.612
	92	8.92425	1.478	4.612
	93	8.47804	1.263	3.604
	94	8.05413	-0.879	3.947
	95	7.65143	0.934	2.549
	96	7.26886	0.875	2.415
	97	6.90541	-0.539	2.952
	98	6.56014	-1.347	5.770
	99	6.23214	-1.347	5.770
	100	5.92053	-2.690	13.687
	101	5.62450	-0.474	2.796
	102	5.34328	1.208	3.389
	103	5.07611	1.208	3.389
	104	4.82231	1.208	3.389
	105	4.58119	1.439	4.411
	106	4.35213	1.400	4.216
	107	4.13453	1.971	8.118
	108	3.92780	1.971	8.118
	109	3.73141	1.771	6.470
	110	3.54484	1.771	6.470
	111	3.36760	1.571	5.140
	112	3.19922	-1.742	7.695
	113	3.03926	1.669	5.752
	114	2.88729	0.478	1.885
	115	2.74293	0.478	1.885
	116	2.60578	0.478	1.885
	117	2.47549	-0.272	2.379
	118	2.35172	0.922	2.520
	119	2.23413	0.602	1.984
	120	2.12243	0.602	1.984
	121	2.01631	0.602	1.984
	122	1.91549	0.602	1.984
	123	1.81972	0.525	1.916
	124	1.72873	0.860	2.383
	125	1.64229	0.860	2.383
	126	1.56018	0.860	2.383
	127	1.48217	0.860	2.383
	128	1.40806	0.860	2.383
	129	1.33766	-0.641	3.219
	130	1.27078	-1.127	4.847
	131	1.20724	-1.320	5.648
	132	1.14687	-0.902	4.023
	133	1.08953	-0.902	4.023
	134	1.03505	0.836	2.333
	135	0.98330	0.836	2.333
	136	0.93414	0.836	2.333
	137	0.88743	0.582	1.965
	138	0.84306	0.582	1.965
	139	0.80091	0.057	1.948
	140	0.76086	0.057	1.948
	141	0.72282	0.057	1.948
	142	0.68668	0.057	1.948
	143	0.65234	0.057	1.948
	144	0.61973	0.057	1.948
	145	0.58874	0.029	1.972
	146	0.55930	0.029	1.972

147	0.53134	0.029	1.972
148	0.50477	0.029	1.972
149	0.47953	0.029	1.972
150	0.45555	0.758	2.192
151	0.43278	0.758	2.192
152	0.41114	0.758	2.192
153	0.39058	0.758	2.192
154	0.37105	0.758	2.192
155	0.35250	0.758	2.192
156	0.33487	0.758	2.192
157	0.31813	0.758	2.192
158	0.30222	0.883	2.431
159	0.28711	0.582	1.964
160	0.27276	0.582	1.964
161	0.25912	0.582	1.964
162	0.24616	0.582	1.964
163	0.23386	0.582	1.964
164	0.22216	0.582	1.964
165	0.21105	-0.066	2.073
166	0.20050	-0.066	2.073
167	0.19048	-0.066	2.073
168	0.18095	-0.066	2.073
169	0.17191	-0.066	2.073
170	0.16331	-0.066	2.073
171	0.15514	-0.066	2.073
172	0.14739	-0.066	2.073
173	0.14002	-0.066	2.073
174	0.13302	0.185	1.867
175	0.12637	0.185	1.867
176	0.12005	0.185	1.867
177	0.11405	0.185	1.867
178	0.10834	0.185	1.867
179	0.10293	0.185	1.867
+-----+-----+-----+-----+			

Таблица 2 – Алгоритм имитации отжига для функции $f(x) \cdot \sin 5x$

	N	T	x	f(x)
+-----+-----+-----+-----+				
	0	1000.00000	-2.063	7.396
	1	950.00000	0.218	1.647
	2	902.50000	-1.043	3.965
	3	857.37500	-1.872	-0.559
	4	814.50625	-2.582	-4.339
	5	773.78094	-3.071	-5.751
	6	735.09189	-0.303	-2.433
	7	698.33730	-3.162	1.785
	8	663.42043	-3.723	5.188
	9	630.24941	-4.658	30.851
	10	598.73694	-1.640	-6.737
	11	568.80009	-2.462	3.033
	12	540.36009	0.281	1.815
	13	513.34208	0.912	-2.466
	14	487.67498	-1.953	2.971
	15	463.29123	-4.200	-22.644
	16	440.12667	-1.697	-6.024
	17	418.12034	-4.674	31.621
	18	397.21432	0.954	-2.593
	19	377.35360	0.394	1.704
	20	358.48592	1.670	5.066
	21	340.56163	-4.115	-25.864
	22	323.53354	-2.483	1.831

	23	307.35687	-2.018	5.689
	24	291.98902	0.065	0.623
	25	277.38957	-0.693	1.070
	26	263.52009	-1.733	-5.269
	27	250.34409	-1.581	-6.857
	28	237.82689	-2.021	5.828
	29	225.93554	0.250	1.751
	30	214.63876	-3.848	-8.925
	31	203.90683	-0.210	-1.975
	32	193.71148	-3.990	-22.187
	33	184.02591	-4.124	-25.751
	34	174.82461	-2.327	8.951
	35	166.08338	-4.055	-25.314
	36	157.77921	0.365	1.784
	37	149.89025	-0.143	-1.423
	38	142.39574	-3.380	17.835
	39	135.27595	-3.053	-7.039
	40	128.51216	-2.944	-13.040
	41	122.08655	-0.012	-0.125
	42	115.98222	-0.575	-0.807
	43	110.18311	-3.007	-10.027
	44	104.67395	-0.424	-2.288
	45	99.44026	-1.046	3.944
	46	94.46824	-2.806	-14.468
	47	89.74483	-2.806	-14.468
	48	85.25759	-2.806	-14.468
	49	80.99471	1.157	-1.531
	50	76.94498	-2.830	-14.723
	51	73.09773	1.355	1.883
	52	69.44284	-2.766	-13.576
	53	65.97070	-4.910	19.258
	54	62.67216	-4.860	25.438
	55	59.53856	-1.672	-6.414
	56	56.56163	-1.192	1.613
	57	53.73355	-2.669	-9.525
	58	51.04687	0.353	1.807
	59	48.49453	0.353	1.807
	60	46.06980	0.353	1.807
	61	43.76631	-0.865	3.615
	62	41.57799	-4.276	-15.928
	63	39.49909	-0.024	-0.239
	64	37.52414	-0.024	-0.239
	65	35.64793	-1.145	2.611
	66	33.86554	0.151	1.292
	67	32.17226	-0.773	2.392
	68	30.56365	-3.749	2.312
	69	29.03546	-1.984	4.281
	70	27.58369	-1.970	3.692
	71	26.20451	0.994	-2.612
	72	24.89428	-1.585	-6.870
	73	23.64957	-0.797	2.755
	74	22.46709	-1.300	-1.192
	75	21.34373	0.525	0.947
	76	20.27655	0.258	1.772
	77	19.26272	-0.427	-2.273
	78	18.29958	1.769	3.532
	79	17.38460	-2.166	10.002
	80	16.51537	-1.208	1.252
	81	15.68961	1.556	5.033
	82	14.90513	1.556	5.033
	83	14.15987	1.133	-1.814
	84	13.45188	1.133	-1.814
	85	12.77928	-0.624	-0.062
	86	12.14032	-0.624	-0.062

	87	11.53330	-2.409	5.840
	88	10.95664	-2.105	8.710
	89	10.40880	-0.307	-2.442
	90	9.88836	-4.041	-24.832
	91	9.39395	-4.041	-24.832
	92	8.92425	-4.041	-24.832
	93	8.47804	-4.041	-24.832
	94	8.05413	-4.041	-24.832
	95	7.65143	-4.041	-24.832
	96	7.26886	-4.041	-24.832
	97	6.90541	-4.041	-24.832
	98	6.56014	-4.041	-24.832
	99	6.23214	-4.041	-24.832
	100	5.92053	-4.041	-24.832
	101	5.62450	-4.041	-24.832
	102	5.34328	-4.041	-24.832
	103	5.07611	-4.041	-24.832
	104	4.82231	-4.041	-24.832
	105	4.58119	-4.041	-24.832
	106	4.35213	-4.074	-25.731
	107	4.13453	-4.074	-25.731
	108	3.92780	-4.074	-25.731
	109	3.73141	-4.074	-25.731
	110	3.54484	-4.074	-25.731
	111	3.36760	-4.074	-25.731
	112	3.19922	-4.074	-25.731
	113	3.03926	-4.074	-25.731
	114	2.88729	-4.074	-25.731
	115	2.74293	-4.074	-25.731
	116	2.60578	-4.074	-25.731
	117	2.47549	-4.074	-25.731
	118	2.35172	-4.074	-25.731
	119	2.23413	-4.074	-25.731
	120	2.12243	-4.074	-25.731
	121	2.01631	-4.074	-25.731
	122	1.91549	-4.074	-25.731
	123	1.81972	-4.074	-25.731
	124	1.72873	-4.074	-25.731
	125	1.64229	-4.045	-24.992
	126	1.56018	-4.045	-24.992
	127	1.48217	-4.045	-24.992
	128	1.40806	-4.045	-24.992
	129	1.33766	-4.045	-24.992
	130	1.27078	-4.045	-24.992
	131	1.20724	-4.045	-24.992
	132	1.14687	-4.045	-24.992
	133	1.08953	-4.045	-24.992
	134	1.03505	-4.045	-24.992
	135	0.98330	-4.045	-24.992
	136	0.93414	-4.045	-24.992
	137	0.88743	-4.045	-24.992
	138	0.84306	-4.045	-24.992
	139	0.80091	-4.045	-24.992
	140	0.76086	-4.045	-24.992
	141	0.72282	-4.045	-24.992
	142	0.68668	-4.045	-24.992
	143	0.65234	-4.045	-24.992
	144	0.61973	-4.045	-24.992
	145	0.58874	-4.045	-24.992
	146	0.55930	-4.045	-24.992
	147	0.53134	-4.045	-24.992
	148	0.50477	-4.045	-24.992
	149	0.47953	-4.045	-24.992
	150	0.45555	-4.045	-24.992

151	0.43278	-4.045	-24.992
152	0.41114	-4.045	-24.992
153	0.39058	-4.045	-24.992
154	0.37105	-4.045	-24.992
155	0.35250	-4.045	-24.992
156	0.33487	-4.045	-24.992
157	0.31813	-4.045	-24.992
158	0.30222	-4.045	-24.992
159	0.28711	-4.045	-24.992
160	0.27276	-4.045	-24.992
161	0.25912	-4.045	-24.992
162	0.24616	-4.045	-24.992
163	0.23386	-4.045	-24.992
164	0.22216	-4.045	-24.992
165	0.21105	-4.045	-24.992
166	0.20050	-4.045	-24.992
167	0.19048	-4.045	-24.992
168	0.18095	-4.045	-24.992
169	0.17191	-4.045	-24.992
170	0.16331	-4.045	-24.992
171	0.15514	-4.045	-24.992
172	0.14739	-4.045	-24.992
173	0.14002	-4.045	-24.992
174	0.13302	-4.045	-24.992
175	0.12637	-4.045	-24.992
176	0.12005	-4.045	-24.992
177	0.11405	-4.045	-24.992
178	0.10834	-4.045	-24.992
179	0.10293	-4.045	-24.992

+-----+-----+-----+-----+

Ссылка на репозиторий с выполненной работой: <https://github.com/shreddered/lab-03>

4. Выводы

Применимость алгоритма имитации отжига не зависит от того, является ли функция унимодальной или мультимодальной. Однако, по сравнению с методом случайного поиска, данный алгоритм является более эффективным.

Приложение А. 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();
}
```

Файл source/algorithms/package.d

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

Файл source/algorithms/annealing_searcher.d

```
module algorithms.annealing_searcher;

import algorithms.searcher;

import std.math : exp;
import std.random : uniform;
import std.stdio : writefln, writeln;

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

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

    public override void search() {
        // function for evaluating probability of transition
        alias probability = (delta, t) => exp(-delta/t);

        // table header
        writeln("+-----+-----+-----+-----+");
        writefln("|%5c|%10c|%7c|%7s|", 'N', 'T', 'x', "f(x)");
        writeln("+-----+-----+-----+-----+");

        // initial values
        const double t_min = 0.1;
        double x_min = uniform(_a, _b);
        size_t counter = 0;

        for (double t_max = 1000.0; t_max > t_min; t_max *= 0.95) {
            double x_i = uniform(_a, _b);
            double diff = func(x_i) - func(x_min);
            if (diff <= 0.0) {
                x_min = x_i;
            }
            else {
                auto temp = uniform!(0.0, 1.0);
                if (temp < probability(diff, t_max)) {
                    x_min = x_i;
                }
            }
            writefln("|%5d|%10,4.5f|%7,3.3f|%7,3.3f|", counter++, t_max,
x_min, func(x_min));
        }

        // table footer
        writeln("+-----+-----+-----+-----+");
    }
}
```

Файл source/app.d

```
import algorithms;

import std.math : exp, sin;
import std.stdio : writeln;

alias fun1 = (x) => (1 - x) * (1 - x) + exp(x);

alias fun2 = (x) => fun1(x) * sin(5 * x);

int main() {
    writeln("f(x) = (1 - x)^2 + exp(x)");
    ISearcher searcher = new AnnealingSearcher!fun1;
    searcher.setInterval(-5, 2);
    searcher.search();

    writeln('\n', "f1(x) = f(x) * sin(5 * x)");
    searcher = new AnnealingSearcher!fun2;
    searcher.setInterval(-5, 2);
    searcher.search();
    return 0;
}
```

Приложение Б. Контрольный вопрос

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

В алгоритме имитации отжига новое решение всегда заменяет предыдущее, если исследуемая функция меньше в этой точке, иначе переход к новому решению осуществляется с некоторой вероятностью, зависящей от параметра, который принято называть температурой.

Вероятность упомянутого перехода экспоненциально зависит от величины частного увеличения функции и температуры, умноженного на -1 . Таким образом, при уменьшении температуры вероятность осуществления перехода уменьшается.

Метод имитации отжига применяется во множестве оптимизационных задач:

- Работа с финансами
- Компьютерная графика
- Комбинаторные задачи
- Обучение нейронных сетей