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 15

Выполнил: Ушаков З. М.,
студент группы ИУ8-31

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

Цель работы

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

Условие задачи

1. На интервале [9, 12] задана унимодальная функция одного переменного $f(x) = x^2 * \sin(x)$. 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 $f(x) * \sin(5x)$.

Графики заданных функций

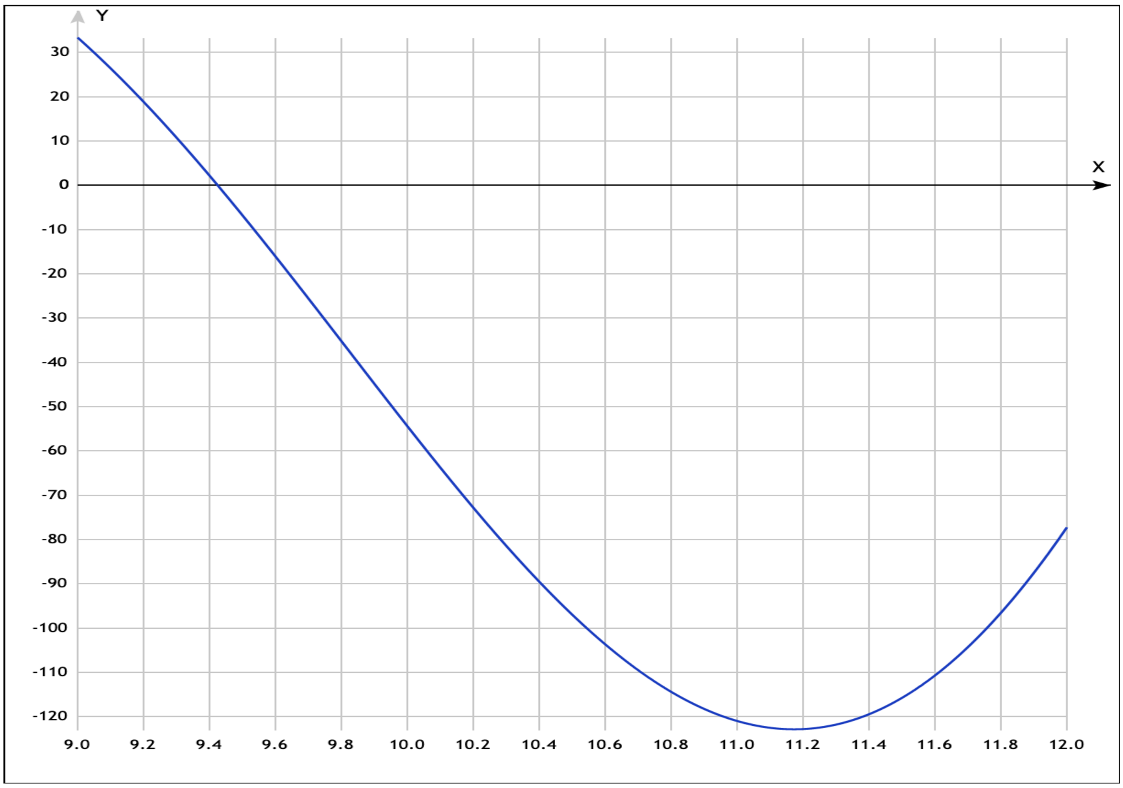

Рисунок 1 - $f(x) = x^2 * sin(x)$

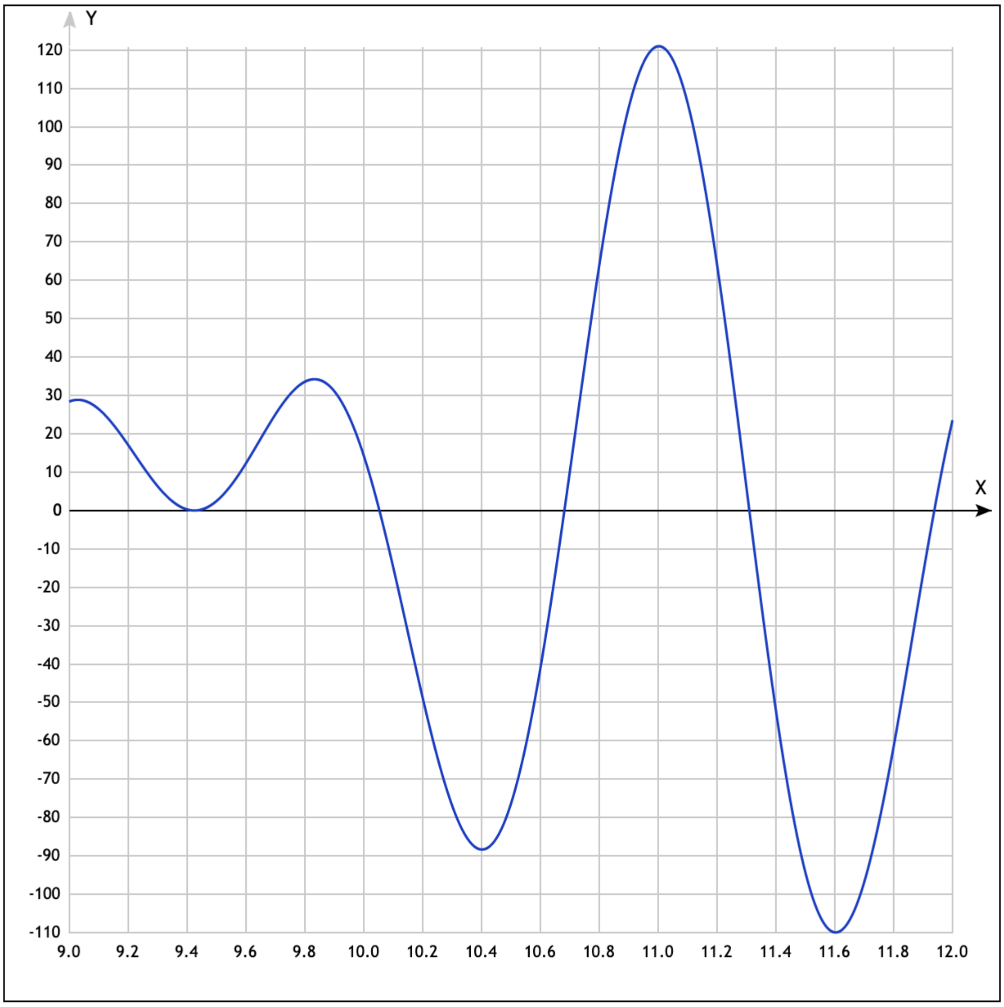

Рисунок 2 - $f(x) = x^2 * sin(x) * sin\,(5x)$

Результат алгоритма имитации отжига

Вариант 15

Функция y = x^2 * sin(x)

Интервал [9, 12]

N	T	x	f(x)
1	10000	9.50556	-7.29139
2	9500	9.48436	-5.35618
3	9025	9.89946	-44.791
4	8573.75	10.993	-120.846
5	8145.06	10.8817	-117.645
6	7737.81	11.0974	-122.515
7	7350.92	11.9278	-84.8062

8	6983.37	10.9528	-119.854
9	6634.2	11.3397	-121.049
10	6302.49	10.7902	-113.982
11	5987.37	10.3758	-87.6321
12	5688	9.95459	-50.0793
13	5403.6	11.4948	-116.006
14	5133.42	10.2527	-77.4189
15	4876.75	10.667	-107.698
16	4632.91	9.82938	-38.0334
17	4401.27	9.41275	1.06591
18	4181.2	11.4644	-117.253
19	3972.14	9.43323	-0.751713
20	3773.54	11.308	-121.681
21	3584.86	9.14847	22.832
22	3405.62	10.8343	-115.859
23	3235.34	10.9211	-118.941
24	3073.57	11.878	-89.634
25	2919.89	11.6592	-107.088
26	2773.9	10.413	-90.5444
27	2635.2	10.0312	-57.3522
28	2503.44	10.9879	-120.729
29	2378.27	10.6162	-104.688
30	2259.36	9.82503	-37.6136
31	2146.39	10.9377	-119.433
32	2039.07	10.4147	-90.6719
33	1937.11	11.6698	-106.383
34	1840.26	11.7744	-98.6709
35	1748.25	11.5291	-114.446
36	1660.83	11.1069	-122.599
37	1577.79	9.62716	-18.6292
38	1498.9	11.5844	-111.599
39	1423.96	9.17665	20.681
40	1352.76	10.4468	-93.1119

41	1285.12	10.6578	-107.169
42	1220.87	9.20445	18.5159
43	1159.82	11.7476	-100.789
44	1101.83	10.5526	-100.609
45	1046.74	10.5837	-102.646
46	994.403	10.5837	-102.646
47	944.682	10.0596	-60.0119
48	897.448	10.6538	-106.937
49	852.576	11.4523	-117.712
50	809.947	11.1901	-122.857
51	769.45	10.5583	-100.987
52	730.977	11.5772	-111.995
53	694.428	11.9298	-84.597
54	659.707	9.93596	-48.2958
55	626.722	9.85458	-40.4658
56	595.386	9.53431	-9.93703
57	565.616	10.2153	-74.1635
58	537.335	10.5121	-97.8371
59	510.469	10.7973	-114.297
60	484.945	9.11003	25.6927
61	460.698	11.1055	-122.588
62	437.663	11.4836	-116.476
63	415.78	11.4836	-116.476
64	394.991	11.7932	-97.138
65	375.241	10.2877	-80.4132
66	356.479	10.3899	-88.7491
67	338.655	11.2657	-122.314
68	321.723	10.5852	-102.744
69	305.636	9.1146	25.3569
70	290.355	10.248	-77.0148
71	275.837	11.2578	-122.406
72	262.045	10.2596	-78.0135
73	248.943	9.33592	7.73479

74	236.496	9.78169	-33.4294
75	224.671	11.097	-122.51
76	213.437	10.2704	-78.9373
77	202.765	10.2704	-78.9373
78	192.627	11.6384	-108.417
79	182.996	11.0425	-121.802
80	173.846	11.6088	-110.209
81	165.154	11.361	-120.549
82	156.896	11.361	-120.549
83	149.051	11.361	-120.549
84	141.599	11.2653	-122.318
85	134.519	11.2283	-122.676
86	127.793	11.2283	-122.676
87	121.403	11.2283	-122.676
88	115.333	9.47773	-4.75392
89	109.566	11.066	-122.153
90	104.088	11.4984	-115.847
91	98.8836	11.4984	-115.847
92	93.9395	11.6482	-107.8
93	89.2425	11.2412	-122.572
94	84.7804	10.2923	-80.7948
95	80.5413	11.0235	-121.469
96	76.5143	11.0235	-121.469
97	72.6886	11.0235	-121.469
98	69.0541	11.0235	-121.469
99	65.6014	11.0235	-121.469
100	62.3214	11.0235	-121.469
101	59.2053	11.0235	-121.469
102	56.245	10.3031	-81.704
103	53.4328	11.1631	-122.87
104	50.7611	11.1631	-122.87
105	48.2231	11.0963	-122.503
106	45.8119	11.0963	-122.503

107	43.5213	11.0963	-122.503
108	41.3453	10.7797	-113.504
109	39.278	10.7797	-113.504
110	37.3141	10.7797	-113.504
111	35.4484	11.2692	-122.27
112	33.676	11.2692	-122.27
113	31.9922	11.2692	-122.27
114	30.3926	11.2692	-122.27
115	28.8729	11.2692	-122.27
116	27.4293	10.6589	-107.232
117	26.0578	10.7981	-114.335
118	24.7549	10.7981	-114.335
119	23.5172	10.7981	-114.335
120	22.3413	11.775	-98.6223
121	21.2243	11.1765	-122.875
122	20.1631	11.1765	-122.875
123	19.1549	11.1765	-122.875
124	18.1972	11.1765	-122.875
125	17.2873	11.1765	-122.875
126	16.4229	10.8249	-115.475
127	15.6018	10.8249	-115.475
128	14.8217	10.8249	-115.475
129	14.0806	11.8778	-89.6439
130	13.3766	11.8778	-89.6439
131	12.7078	11.8778	-89.6439
132	12.0724	11.122	-122.712
133	11.4687	11.122	-122.712
134	10.8953	11.122	-122.712
135	10.3505	11.122	-122.712
136	9.83302	11.122	-122.712
137	9.34136	11.1534	-122.852
138	8.8743	11.1689	-122.875
139	8.43058	11.1689	-122.875

140	8.00905	11.1689	-122.875
141	7.6086	11.1689	-122.875
142	7.22817	11.1689	-122.875
143	6.86676	11.3816	-120.008
144	6.52342	11.3511	-120.789
145	6.19725	11.3058	-121.72
146	5.88739	11.3058	-121.72
147	5.59302	11.3058	-121.72
148	5.31337	11.3058	-121.72
149	5.0477	11.3058	-121.72
150	4.79532	11.3058	-121.72
151	4.55555	11.3058	-121.72
152	4.32777	11.3058	-121.72
153	4.11138	11.3058	-121.72
154	3.90581	11.3058	-121.72
155	3.71052	11.3058	-121.72
156	3.525	11.3879	-119.83
157	3.34875	11.3879	-119.83
158	3.18131	11.3879	-119.83
159	3.02224	11.3879	-119.83
160	2.87113	11.3879	-119.83
161	2.72758	11.3879	-119.83
162	2.5912	11.2992	-121.832
163	2.46164	11.2992	-121.832
164	2.33856	11.2992	-121.832
165	2.22163	11.386	-119.885
166	2.11055	11.386	-119.885
167	2.00502	11.2976	-121.858
168	1.90477	11.2976	-121.858
169	1.80953	11.2976	-121.858
170	1.71905	11.2976	-121.858
171	1.6331	11.2976	-121.858
172	1.55145	11.2976	-121.858

173	1.47387	11.2976	-121.858
174	1.40018	11.2976	-121.858
175	1.33017	11.2976	-121.858
176	1.26366	11.2976	-121.858
177	1.20048	11.2976	-121.858
178	1.14045	11.2976	-121.858
179	1.08343	11.0674	-122.172
180	1.02926	11.0674	-122.172
181	0.977798	11.0674	-122.172
182	0.928908	11.0674	-122.172
183	0.882462	11.0674	-122.172
184	0.838339	11.0674	-122.172
185	0.796422	11.0674	-122.172
186	0.756601	11.0674	-122.172
187	0.718771	11.0674	-122.172
188	0.682833	11.0674	-122.172
189	0.648691	11.0674	-122.172
190	0.616256	11.0674	-122.172
191	0.585444	11.0674	-122.172
192	0.556171	11.0674	-122.172
193	0.528363	11.0674	-122.172
194	0.501945	11.0674	-122.172
195	0.476847	11.0674	-122.172
196	0.453005	11.0674	-122.172
197	0.430355	11.0674	-122.172
198	0.408837	11.0674	-122.172
199	0.388395	11.0674	-122.172
200	0.368975	11.0674	-122.172
201	0.350527	11.0674	-122.172
202	0.333	11.0674	-122.172
203	0.31635	11.0674	-122.172
204	0.300533	11.0674	-122.172
205	0.285506	11.0674	-122.172

```
|       206 |  0.271231 |   11.0674 |-122.172
|       207 |  0.257669 |   11.0674 |-122.172
|       208 |  0.244786 |   11.0674 |-122.172
|       209 |  0.232547 |   11.0674 |-122.172
|       210 |  0.220919 |   11.0674 |-122.172
|       211 |  0.209873 |   11.0674 |-122.172
|       212 |   0.19938 |   11.0674 |-122.172
|       213 |  0.189411 |   11.0674 |-122.172
|       214 |   0.17994 |   11.0674 |-122.172
|       215 |  0.170943 |   11.0674 |-122.172
|       216 |  0.162396 |   11.0674 |-122.172
|       217 |  0.154276 |   11.0674 |-122.172
|       218 |  0.146562 |   11.0674 |-122.172
|       219 |  0.139234 |   11.0674 |-122.172
|       220 |  0.132272 |   11.0674 |-122.172
|       221 |  0.125659 |   11.0674 |-122.172
|       222 |  0.119376 |   11.0674 |-122.172
|       223 |  0.113407 |   11.0674 |-122.172
|       224 |  0.107737 |   11.0674 |-122.172
|       225 |   0.10235 |   11.0674 |-122.172
---------------------------------------------
Result: Xmin = 11.0674 | Fmin = -122.172


Функция y = x^2 * sin(x) * sin(5 * x)
Интервал [9, 12]
---------------------------------------------
|         N |         T |         x |    f(x)
---------------------------------------------
|         1 |     10000 |   11.7355 |-86.2717
|         2 |      9500 |   9.74602 |29.9719
|         3 |      9025 |   10.3195 |-80.7153
|         4 |   8573.75 |   9.81776 |34.0817
```

5	8145.06	9.5879	10.8708
6	7737.81	10.865	92.9567
7	7350.92	11.9081	-12.9387
8	6983.37	10.2355	-60.0398
9	6634.2	9.09525	26.6903
10	6302.49	11.1088	103.492
11	5987.37	11.2288	48.313
12	5688	10.6074	-37.6463
13	5403.6	9.92181	28.647
14	5133.42	9.53517	5.25235
15	4876.75	10.8911	102.242
16	4632.91	10.535	-66.4694
17	4401.27	11.4071	-55.7932
18	4181.2	10.6484	-17.5085
19	3972.14	11.0402	118.748
20	3773.54	10.1195	-21.3834
21	3584.86	11.3219	-7.39371
22	3405.62	11.2556	32.757
23	3235.34	11.8503	-39.1492
24	3073.57	9.15323	21.964
25	2919.89	11.1455	89.8829
26	2773.9	10.5764	-51.2007
27	2635.2	11.9281	-4.20944
28	2503.44	10.5664	-55.2205
29	2378.27	10.3111	-79.1379
30	2259.36	11.0812	111.296
31	2146.39	10.7668	46.7432
32	2039.07	11.3647	-32.7015
33	1937.11	9.39126	0.492975
34	1840.26	11.6448	-107.421
35	1748.25	11.6448	-107.421
36	1660.83	10.4104	-88.2508
37	1577.79	11.6506	-106.691

38	1498.9	11.739	-85.092
39	1423.96	9.1661	20.6702
40	1352.76	11.0178	120.615
41	1285.12	10.8683	94.2233
42	1220.87	9.122	24.7707
43	1159.82	9.61868	14.7009
44	1101.83	10.6434	-20.1002
45	1046.74	11.942	1.64184
46	994.403	9.76714	31.7079
47	944.682	9.63015	16.1827
48	897.448	9.97736	19.3175
49	852.576	11.3765	-39.3896
50	809.947	9.07248	27.8865
51	769.45	10.6522	-15.539
52	730.977	11.2526	34.4865
53	694.428	11.4143	-59.4258
54	659.707	11.4143	-59.4258
55	626.722	11.1657	81.0152
56	595.386	11.7634	-76.31
57	565.616	10.8653	93.0699
58	537.335	11.8859	-22.9014
59	510.469	9.90474	30.6039
60	484.945	10.8261	76.4546
61	460.698	10.533	-67.1026
62	437.663	11.3807	-41.6889
63	415.78	11.4059	-55.1644
64	394.991	9.93277	27.1625
65	375.241	11.1115	102.598
66	356.479	9.93248	27.2036
67	338.655	9.02904	28.8482
68	321.723	10.9409	115.078
69	305.636	10.4804	-80.6931
70	290.355	10.4804	-80.6931

71	275.837	9.52316	4.20745
72	262.045	9.36272	1.66004
73	248.943	11.956	7.33837
74	236.496	10.269	-69.491
75	224.671	10.269	-69.491
76	213.437	10.3099	-78.9016
77	202.765	9.94468	25.3471
78	192.627	9.76094	31.2384
79	182.996	10.6485	-17.4586
80	173.846	9.0039	28.5177
81	165.154	9.1774	19.4843
82	156.896	10.4619	-83.8697
83	149.051	9.63231	16.4638
84	141.599	9.67682	22.2356
85	134.519	9.67682	22.2356
86	127.793	10.4272	-87.5472
87	121.403	11.4861	-89.8294
88	115.333	11.4861	-89.8294
89	109.566	10.4035	-88.3394
90	104.088	9.892	31.7843
91	98.8836	10.3784	-87.7066
92	93.9395	10.3784	-87.7066
93	89.2425	10.3784	-87.7066
94	84.7804	10.347	-84.8871
95	80.5413	10.347	-84.8871
96	76.5143	11.5755	-108.823
97	72.6886	11.4224	-63.4209
98	69.0541	10.509	-74.1054
99	65.6014	10.509	-74.1054
100	62.3214	10.3607	-86.383
101	59.2053	10.3607	-86.383
102	56.245	10.3607	-86.383
103	53.4328	9.90925	30.129

104	50.7611	9.03093	28.8457
105	48.2231	10.6503	-16.5588
106	45.8119	10.6503	-16.5588
107	43.5213	9.98597	17.4793
108	41.3453	9.66139	20.2568
109	39.278	10.5164	-72.1063
110	37.3141	11.5542	-106.406
111	35.4484	11.5542	-106.406
112	33.676	11.5542	-106.406
113	31.9922	11.5542	-106.406
114	30.3926	11.5542	-106.406
115	28.8729	11.5542	-106.406
116	27.4293	11.5542	-106.406
117	26.0578	11.5408	-104.184
118	24.7549	11.5408	-104.184
119	23.5172	11.5408	-104.184
120	22.3413	11.5408	-104.184
121	21.2243	11.5408	-104.184
122	20.1631	11.5408	-104.184
123	19.1549	11.5408	-104.184
124	18.1972	11.5408	-104.184
125	17.2873	11.5408	-104.184
126	16.4229	11.5408	-104.184
127	15.6018	11.5408	-104.184
128	14.8217	11.5408	-104.184
129	14.0806	11.5408	-104.184
130	13.3766	11.5408	-104.184
131	12.7078	11.5408	-104.184
132	12.0724	11.5408	-104.184
133	11.4687	11.5408	-104.184
134	10.8953	11.5408	-104.184
135	10.3505	11.5408	-104.184
136	9.83302	11.5408	-104.184

137	9.34136	11.5408	-104.184
138	8.8743	11.5408	-104.184
139	8.43058	11.5408	-104.184
140	8.00905	11.5408	-104.184
141	7.6086	11.5408	-104.184
142	7.22817	11.5408	-104.184
143	6.86676	11.5408	-104.184
144	6.52342	11.5408	-104.184
145	6.19725	11.5408	-104.184
146	5.88739	11.5408	-104.184
147	5.59302	11.5408	-104.184
148	5.31337	11.5408	-104.184
149	5.0477	11.5408	-104.184
150	4.79532	11.5408	-104.184
151	4.55555	11.5408	-104.184
152	4.32777	11.5408	-104.184
153	4.11138	11.5408	-104.184
154	3.90581	11.5408	-104.184
155	3.71052	11.5408	-104.184
156	3.525	11.5408	-104.184
157	3.34875	11.5408	-104.184
158	3.18131	11.5408	-104.184
159	3.02224	11.5408	-104.184
160	2.87113	11.6271	-109.103
161	2.72758	11.6271	-109.103
162	2.5912	11.6271	-109.103
163	2.46164	11.6271	-109.103
164	2.33856	11.6271	-109.103
165	2.22163	11.6271	-109.103
166	2.11055	11.6271	-109.103
167	2.00502	11.6271	-109.103
168	1.90477	11.6271	-109.103
169	1.80953	11.6271	-109.103

170	1.71905	11.6271	-109.103
171	1.6331	11.6271	-109.103
172	1.55145	11.6271	-109.103
173	1.47387	11.6271	-109.103
174	1.40018	11.6271	-109.103
175	1.33017	11.6271	-109.103
176	1.26366	11.6271	-109.103
177	1.20048	11.6271	-109.103
178	1.14045	11.6271	-109.103
179	1.08343	11.6271	-109.103
180	1.02926	11.6271	-109.103
181	0.977798	11.6271	-109.103
182	0.928908	11.6271	-109.103
183	0.882462	11.6271	-109.103
184	0.838339	11.6271	-109.103
185	0.796422	11.6271	-109.103
186	0.756601	11.6271	-109.103
187	0.718771	11.6271	-109.103
188	0.682833	11.6271	-109.103
189	0.648691	11.6271	-109.103
190	0.616256	11.6271	-109.103
191	0.585444	11.6271	-109.103
192	0.556171	11.6271	-109.103
193	0.528363	11.6271	-109.103
194	0.501945	11.6271	-109.103
195	0.476847	11.6271	-109.103
196	0.453005	11.6271	-109.103
197	0.430355	11.6271	-109.103
198	0.408837	11.6271	-109.103
199	0.388395	11.6271	-109.103
200	0.368975	11.6271	-109.103
201	0.350527	11.6271	-109.103
202	0.333	11.6271	-109.103

```
|        203 |   0.31635 |    11.6271 |-109.103
|        204 |  0.300533 |    11.6271 |-109.103
|        205 |  0.285506 |    11.6271 |-109.103
|        206 |  0.271231 |    11.6271 |-109.103
|        207 |  0.257669 |    11.6271 |-109.103
|        208 |  0.244786 |    11.6271 |-109.103
|        209 |  0.232547 |    11.6271 |-109.103
|        210 |  0.220919 |    11.6271 |-109.103
|        211 |  0.209873 |    11.6271 |-109.103
|        212 |   0.19938 |    11.6271 |-109.103
|        213 |  0.189411 |    11.5859 |-109.508
|        214 |   0.17994 |    11.5859 |-109.508
|        215 |  0.170943 |    11.5859 |-109.508
|        216 |  0.162396 |    11.5859 |-109.508
|        217 |  0.154276 |    11.5859 |-109.508
|        218 |  0.146562 |    11.5859 |-109.508
|        219 |  0.139234 |    11.5859 |-109.508
|        220 |  0.132272 |    11.5859 |-109.508
|        221 |  0.125659 |    11.5859 |-109.508
|        222 |  0.119376 |    11.5859 |-109.508
|        223 |  0.113407 |    11.5859 |-109.508
|        224 |  0.107737 |    11.5859 |-109.508
|        225 |   0.10235 |    11.5859 |-109.508
---------------------------------------------
Result: Xmin = 11.5859 | Fmin = -109.508
```

Выводы

Метод имитации отжига, как и метод случайного поиска, не зависит от унимодальности или мультимодальности функции. Преимущество это метода над методом случайного поиска в его большей эффективности.

Приложение 1. Исходный код программы

Файл simulated_annealing.hpp:

```
#ifndef TSISA_LAB03_SIMULATED_ANNEALING_HPP
#define TSISA_LAB03_SIMULATED_ANNEALING_HPP
```

```cpp
#include <iostream>
#include <map>
#include <cmath>
#include <random>
#include <iomanip>
#include <string>

using std::cout;
using std::endl;

auto f(const double x) noexcept -> double {
    return pow(x, 2) * sin(x);
}

auto f_m(const double x) noexcept -> double {
    return pow(x, 2) * sin(x) * sin(5 * x);
}

auto P(const double delta_f, const double t_i) noexcept -> double {
    return std::exp((-delta_f)/t_i);
}

auto random(double a, double b) -> double {
    if (a > b) throw std::invalid_argument("Invalid segment");
    std::random_device rd;
    std::mt19937_64 rng(rd());
    std::uniform_real_distribution<double> rand(a, b);
    return rand(rng);
}

auto simulated_annealing(const double a, const double b, const double
t_min, double t_max, double func(double))
                                                noexcept ->
std::map<double,std::pair<double, double>> {
    std::map<double, std::pair<double, double>> result;
    auto x_min = random(a, b);
    while (t_max > t_min) {
        auto x_i = random(a, b);
        auto difference = func(x_i) - func(x_min);
        auto probability = 0.0;
        if (difference <= 0) {
            x_min = x_i;
            probability = 1;
        }
        else {
            auto temp = random(0, 1);
            probability = P(difference, t_max);
            if (temp < probability) {
                x_min = x_i;
            }
```

```
        }
        result[t_max] = {x_min, func(x_min)};
        t_max *= 0.95;
    }

    return result;
}

void print(const std::map<double, std::pair<double, double>>& result)
noexcept {
    size_t counter = 0;

    cout << std::string(45, '-') << endl;
    cout << "|" << std::setw(10) << "N" << " |" << std::setw(10) <<
"T" << " |" << std::setw(10)
    << "x" << " |" << std::setw(8) << "f(x)" << endl;
    cout << std::string(45, '-') << endl;
    for(auto item = result.rbegin(); item != result.rend(); item++) {
        counter++;
        cout << "|" << std::setw(10) << counter << " |" <<
std::setw(10) << item->first
        << " |" << std::setw(10) << item->second.first << " |" <<
item->second.second << endl;
    }
    cout << std::string(45, '-') << endl;
    cout << "Result: Xmin = " << result.begin()->second.first << " |
Fmin = " << result.begin() ->second.second << endl;
}

#endif //TSISA_LAB03_SIMULATED_ANNEALING_HPP
```

Файл main.cpp:

```
#include "simulated_annealing.hpp"
int main() {


    cout << "Вариант 15" << endl;
    cout << "Функция y = x^2 * sin(x)" << endl;
    cout << "Интервал [9, 12]" << endl;
    auto result = simulated_annealing(9, 12, 0.1, 10000, f);
    print(result);


    cout << endl << endl;
    cout << "Функция y = x^2 * sin(x) * sin(5 * x)" << endl;
    cout << "Интервал [9, 12]" << endl;
    auto result_m = simulated_annealing(9, 12, 0.1, 10000, f_m);
    print(result_m);
    return 0;
}
```

Приложение 2. Контрольный вопрос

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

В методе имитации отжига новое решение замещает текущее всегда, если оно лучше, и с некоторой вероятностью, зависящей от параметра, который принято называть температурой, если целевая функция на нем хуже.

Вероятность перехода к худшему решению обратно пропорциональна величине увеличения значения функционала на новом решении и прямо пропорциональна температуре.

Для сходимости метода на каждой последующей итерации вероятность перехода к худшему решению уменьшается. Это обеспечивается уменьшением температуры.

Метод имитации отжига применяется во множестве оптимизационных задач:

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