



**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 4**

**Выполнила: Бояркина Е.Р.,  
студент 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

1. На интервале  $[-2; 0]$  задана унимодальная функция одного переменного  $f(x) = \cos(x) * \text{th}(x)$ . 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

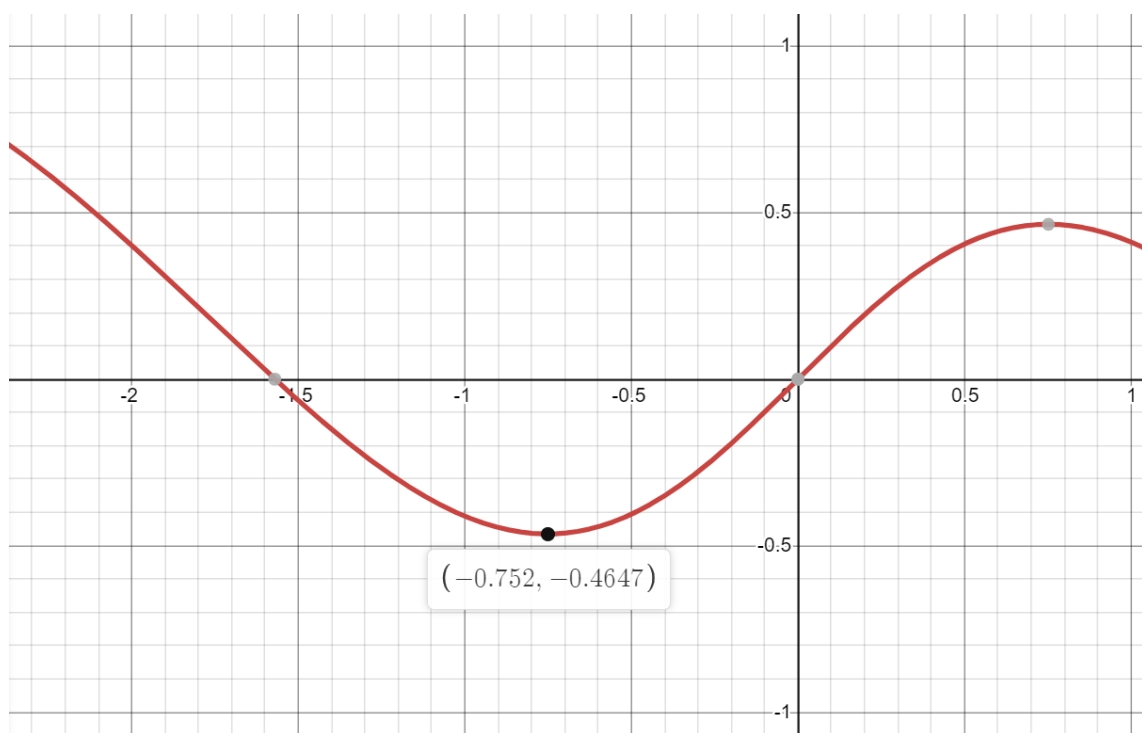


Рисунок 1 – График функции  $f(x) = \cos(x) * \text{th}(x)$  на отрезке  $[-2; 0]$

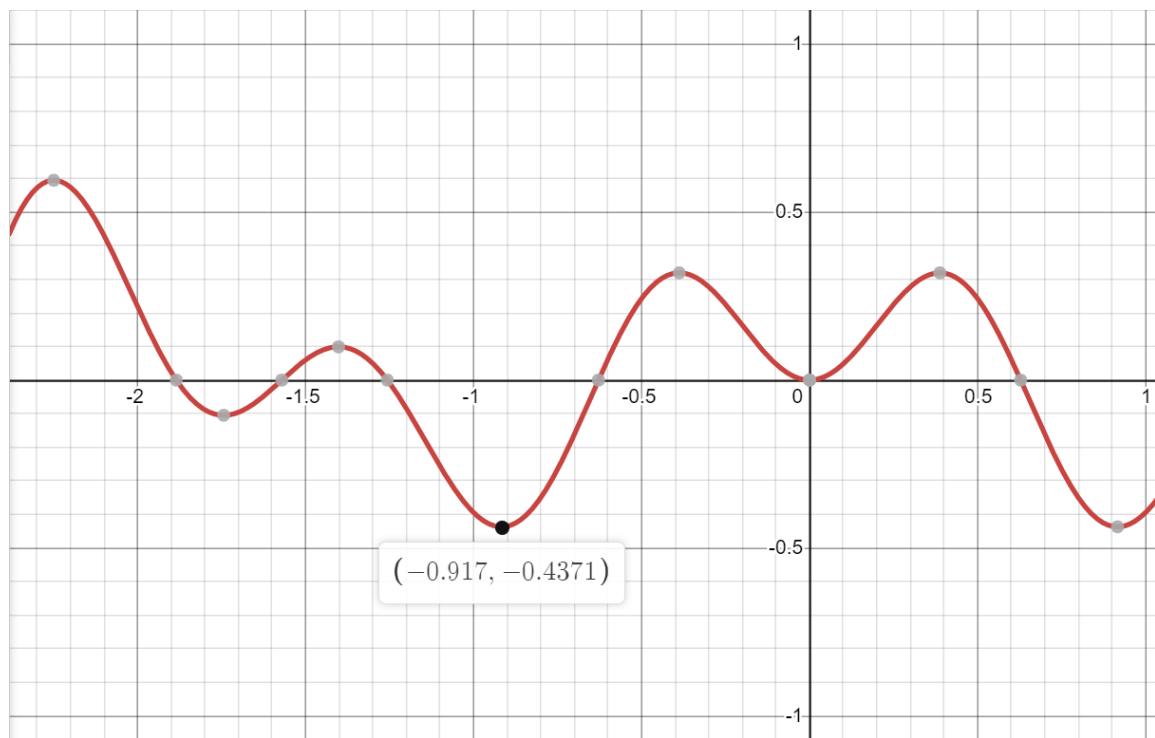


Рисунок 2 – График функции  $f(x) = \cos(x) \cdot \text{th}(x) \cdot \sin(5x)$  на отрезке  $[-2; 0]$

### Имитация отжига для заданных функций

Имитация отжига для  $y = \cos(x) \cdot \text{th}(x)$  на  $[-2; 0]$

N	T	x	f(x)
1	10000	-0.697226	-0.461972
2	9500	-0.192511	-0.186654
3	9025	-1.29344	-0.23549
4	8573.75	-1.3148	-0.21915
5	8145.06	-1.39555	-0.154193
6	7737.81	-0.60799	-0.445456
7	7350.92	-0.141545	-0.139201
8	6983.37	-0.729209	-0.46426
9	6634.2	-0.716453	-0.463575
10	6302.49	-0.686239	-0.460746
11	5987.37	-1.27891	-0.246379
12	5688	-0.851283	-0.455869
13	5403.6	-1.11399	-0.355281
14	5133.42	-1.67077	0.0929823
15	4876.75	-0.731162	-0.464338
16	4632.91	-1.03757	-0.394924
17	4401.27	-1.80206	0.217061
18	4181.2	-1.42631	-0.128274
19	3972.14	-1.51854	-0.0474503
20	3773.54	-0.398022	-0.348687
21	3584.86	-1.34507	-0.195362
22	3405.62	-1.65221	0.0755638
23	3235.34	-0.295297	-0.27458
24	3073.57	-1.80816	0.222831
25	2919.89	-1.15177	-0.33296
26	2773.9	-0.850551	-0.455998
27	2635.2	-0.282723	-0.264489

28	2503.44	-0.670614	-0.458613	
29	2378.27	-0.339183	-0.308132	
30	2259.36	-1.49107	-0.0719552	
31	2146.39	-0.196295	-0.19009	
32	2039.07	-1.33457	-0.203692	
33	1937.11	-1.04892	-0.389526	
34	1840.26	-1.62499	0.0501219	
35	1748.25	-0.422376	-0.363871	
36	1660.83	-0.378185	-0.335612	
37	1577.79	-1.57219	0.00127993	
38	1498.9	-1.67736	0.099183	
39	1423.96	-1.75683	0.174261	
40	1352.76	-0.752953	-0.464734	
41	1285.12	-0.482559	-0.3971	
42	1220.87	-1.85791	0.269726	
43	1159.82	-1.76012	0.177381	
44	1101.83	-0.603473	-0.444221	
45	1046.74	-0.653462	-0.455744	
46	994.403	-1.70183	0.122253	
47	944.682	-1.48955	-0.0733067	
48	897.448	-0.0901517	-0.0895431	
49	852.576	-1.41575	-0.137237	
50	809.947	-1.00668	-0.408692	
51	769.45	-1.6801	0.101768	
52	730.977	-0.0011597	-0.0011597	
53	694.428	-0.462539	-0.386742	
54	659.707	-0.640828	-0.453276	
55	626.722	-0.111026	-0.109892	
56	595.386	-0.625813	-0.449952	
57	565.616	-0.351878	-0.317328	
58	537.335	-1.7351	0.153695	
59	510.469	-1.40135	-0.149353	
60	484.945	-0.881375	-0.449785	
61	460.698	-1.57128	0.000439951	
62	437.663	-1.55913	-0.0106781	
63	415.78	-1.08731	-0.369984	
64	394.991	-0.661885	-0.457222	
65	375.241	-0.989166	-0.415889	
66	356.479	-0.398877	-0.349236	
67	338.655	-1.84088	0.253703	
68	321.723	-0.661946	-0.457232	
69	305.636	-1.33195	-0.205761	
70	290.355	-0.368114	-0.328737	
71	275.837	-1.53404	-0.0334786	
72	262.045	-0.469497	-0.390422	
73	248.943	-0.0432752	-0.0432078	
74	236.496	-0.954314	-0.428845	
75	224.671	-0.842311	-0.457384	
76	213.437	-0.252632	-0.239539	
77	202.765	-1.2899	-0.238161	
78	192.627	-1.94256	0.348637	
79	182.996	-0.0143437	-0.0143412	
80	173.846	-1.96606	0.370249	
81	165.154	-1.7691	0.185875	
82	156.896	-0.916898	-0.440658	
83	149.051	-0.905362	-0.443849	
84	141.599	-0.731162	-0.464338	
85	134.519	-0.619221	-0.448359	
86	127.793	-1.17313	-0.319601	

87	121.403	-1.99738	0.398798	
88	115.333	-1.0875	-0.369887	
89	109.566	-1.41942	-0.134136	
90	104.088	-1.72222	0.141513	
91	98.8836	-1.51726	-0.0486011	
92	93.9395	-0.0220954	-0.0220864	
93	89.2425	-0.728355	-0.464223	
94	84.7804	-1.92059	0.3283	
95	80.5413	-1.77587	0.192288	
96	76.5143	-0.25367	-0.240417	
97	72.6886	-0.859401	-0.454379	
98	69.0541	-1.867	0.278267	
99	65.6014	-1.89972	0.308877	
100	62.3214	-1.36851	-0.176469	
101	59.2053	-0.56325	-0.431543	
102	56.245	-1.15134	-0.333222	
103	53.4328	-0.193854	-0.187875	
104	50.7611	-1.91882	0.326656	
105	48.2231	-0.828578	-0.459434	
106	45.8119	-0.458144	-0.384374	
107	43.5213	-0.611408	-0.446364	
108	41.3453	-0.389172	-0.342931	
109	39.278	-0.389172	-0.342931	
110	37.3141	-1.26792	-0.25449	
111	35.4484	-0.977508	-0.420428	
112	33.676	-1.155	-0.33097	
113	31.9922	-0.226142	-0.216703	
114	30.3926	-0.990997	-0.415158	
115	28.8729	-0.836207	-0.458336	
116	27.4293	-0.578509	-0.436709	
117	26.0578	-0.47615	-0.393861	
118	24.7549	-1.50401	-0.0604519	
119	23.5172	-1.44237	-0.114524	
120	22.3413	-0.528581	-0.418201	
121	21.2243	-1.41649	-0.136617	
122	20.1631	-0.96408	-0.425401	
123	19.1549	-1.71972	0.139147	
124	18.1972	-0.740318	-0.464611	
125	17.2873	-0.713706	-0.463388	
126	16.4229	-0.571001	-0.434221	
127	15.6018	-1.73498	0.153579	
128	14.8217	-1.20695	-0.297427	
129	14.0806	-0.505997	-0.408324	
130	13.3766	-0.0711692	-0.0708694	
131	12.7078	-0.234077	-0.223624	
132	12.0724	-1.62761	0.0525682	
133	11.4687	-0.949248	-0.430573	
134	10.8953	-1.35746	-0.185424	
135	10.3505	-0.121342	-0.119862	
136	9.83302	-0.162603	-0.159058	
137	9.34136	-0.755394	-0.464724	
138	8.8743	-1.30564	-0.2262	
139	8.43058	-0.996246	-0.413035	
140	8.00905	-1.91607	0.324105	
141	7.6086	-0.84518	-0.456915	
142	7.22817	-1.29606	-0.233503	
143	6.86676	-1.21677	-0.290756	
144	6.52342	-1.34812	-0.192925	
145	6.19725	-1.3831	-0.164512	

146	5.88739	-0.441725	-0.375233	
147	5.59302	-0.123295	-0.121743	
148	5.31337	-1.4929	-0.070332	
149	5.0477	-1.98871	0.390926	
150	4.79532	-1.62725	0.0522268	
151	4.55555	-0.200323	-0.193733	
152	4.32777	-1.8001	0.215214	
153	4.11138	-1.70708	0.127209	
154	3.90581	-1.39116	-0.157847	
155	3.71052	-1.74541	0.163456	
156	3.525	-1.36583	-0.178654	
157	3.34875	-0.940641	-0.433418	
158	3.18131	-0.0159307	-0.0159273	
159	3.02224	-1.73797	0.156409	
160	2.87113	-1.53117	-0.0360711	
161	2.72758	-1.3607	-0.182811	
162	2.5912	-1.47826	-0.0832733	
163	2.46164	-1.6118	0.0378575	
164	2.33856	-1.32414	-0.211889	
165	2.22163	-1.51799	-0.0479436	
166	2.11055	-0.797815	-0.462824	
167	2.00502	-0.638142	-0.452712	
168	1.90477	-1.94659	0.352352	
169	1.80953	-0.368969	-0.329326	
170	1.71905	-1.24552	-0.270683	
171	1.6331	-0.246162	-0.234034	
172	1.55145	-1.25748	-0.262093	
173	1.47387	-1.0148	-0.405205	
174	1.40018	-0.894681	-0.446609	
175	1.33017	-0.894681	-0.446609	
176	1.26366	-1.41679	-0.136359	
177	1.20048	-1.41679	-0.136359	
178	1.14045	-0.313852	-0.289091	
179	1.08343	-0.510636	-0.410428	
180	1.02926	-0.350719	-0.316497	
181	0.977798	-1.30741	-0.224842	
182	0.928908	-1.30741	-0.224842	
183	0.882462	-0.460768	-0.385792	
184	0.838339	-1.90399	0.31286	
185	0.796422	-0.105838	-0.104855	
186	0.756601	-1.7036	0.123924	
187	0.718771	-0.690573	-0.461257	
188	0.682833	-0.625874	-0.449966	
189	0.648691	-0.816858	-0.460922	
190	0.616256	-1.0347	-0.396259	
191	0.585444	-1.13767	-0.341489	
192	0.556171	-1.17392	-0.319095	
193	0.528363	-0.319224	-0.293204	
194	0.501945	-0.319224	-0.293204	
195	0.476847	-0.319224	-0.293204	
196	0.453005	-0.35139	-0.316978	
197	0.430355	-0.384594	-0.339904	
198	0.408837	-0.384594	-0.339904	
199	0.388395	-0.384594	-0.339904	
200	0.368975	-0.211188	-0.20348	
201	0.350527	-1.60881	0.035081	
202	0.333	-0.627705	-0.450394	
203	0.31635	-0.627705	-0.450394	
204	0.300533	-0.627705	-0.450394	

205	0.285506	-0.627705	-0.450394	
206	0.271231	-1.18967	-0.308908	
207	0.257669	-1.18967	-0.308908	
208	0.244786	-1.18967	-0.308908	
209	0.232547	-1.18967	-0.308908	
210	0.220919	-0.318613	-0.292738	
211	0.209873	-0.0700705	-0.0697844	
212	0.19938	-0.835536	-0.458436	
213	0.189411	-0.835536	-0.458436	
214	0.17994	-0.835536	-0.458436	
215	0.170943	-0.381909	-0.338113	
216	0.162396	-0.381909	-0.338113	
217	0.154276	-0.381909	-0.338113	
218	0.146562	-0.391919	-0.34473	
219	0.139234	-0.175176	-0.170752	
220	0.132272	-0.824854	-0.459933	
221	0.125659	-0.456679	-0.383577	
222	0.119376	-0.456679	-0.383577	
223	0.113407	-0.456679	-0.383577	
224	0.107737	-0.456679	-0.383577	
225	0.10235	-1.35789	-0.185079	
226	0.0972324	-1.35789	-0.185079	
227	0.0923708	-0.452956	-0.381536	
228	0.0877523	-0.452956	-0.381536	
229	0.0833647	-0.452956	-0.381536	
230	0.0791964	-0.798975	-0.462727	
231	0.0752366	-0.798975	-0.462727	
232	0.0714748	-0.798975	-0.462727	
233	0.067901	-0.798975	-0.462727	
234	0.064506	-0.798975	-0.462727	
235	0.0612807	-0.798975	-0.462727	
236	0.0582167	-0.798975	-0.462727	
237	0.0553058	-0.798975	-0.462727	
238	0.0525405	-0.602741	-0.444018	
239	0.0499135	-0.789453	-0.463456	
240	0.0474178	-0.789453	-0.463456	
241	0.0450469	-0.789453	-0.463456	
242	0.0427946	-0.789453	-0.463456	
243	0.0406549	-0.789453	-0.463456	
244	0.0386221	-0.613361	-0.446874	
245	0.036691	-0.766747	-0.464535	
246	0.0348565	-0.766747	-0.464535	
247	0.0331136	-0.766747	-0.464535	
248	0.031458	-0.766747	-0.464535	
249	0.0298851	-0.766747	-0.464535	
250	0.0283908	-0.766747	-0.464535	
251	0.0269713	-0.72158	-0.463887	
252	0.0256227	-0.72158	-0.463887	
253	0.0243416	-0.72158	-0.463887	
254	0.0231245	-0.72158	-0.463887	
255	0.0219683	-0.72158	-0.463887	
256	0.0208699	-0.72158	-0.463887	
257	0.0198264	-0.72158	-0.463887	
258	0.018835	-0.72158	-0.463887	
259	0.0178933	-0.72158	-0.463887	
260	0.0169986	-0.72158	-0.463887	
261	0.0161487	-0.72158	-0.463887	
262	0.0153413	-0.72158	-0.463887	
263	0.0145742	-0.72158	-0.463887	

264	0.0138455	-0.72158	-0.463887	
265	0.0131532	-0.72158	-0.463887	
266	0.0124956	-0.731468	-0.46435	
267	0.0118708	-0.731468	-0.46435	
268	0.0112772	-0.731468	-0.46435	
269	0.0107134	-0.731468	-0.46435	
270	0.0101777	-0.731468	-0.46435	

Result: Xmin = -0.731468, Fmin = -0.46435

Имитация отжига для  $y = \cos(x) * \tanh(x) * \sin(5x)$  на  $[-2; 0]$

N	T	x	f (x)	
1	10000	-1.01358	-0.380364	
2	9500	-1.61589	-0.0406028	
3	9025	-0.860073	-0.416235	
4	8573.75	-0.398144	0.318465	
5	8145.06	-1.76727	-0.102212	
6	7737.81	-0.763573	-0.290796	
7	7350.92	-1.83062	-0.0654844	
8	6983.37	-1.03372	-0.356137	
9	6634.2	-0.701254	-0.164902	
10	6302.49	-1.67608	-0.0847128	
11	5987.37	-1.81768	-0.0765162	
12	5688	-1.84295	-0.0532991	
13	5403.6	-0.691122	-0.142493	
14	5133.42	-1.20023	-0.0840267	
15	4876.75	-1.87774	-0.0103963	
16	4632.91	-1.78967	-0.0941749	
17	4401.27	-0.962249	-0.423978	
18	4181.2	-1.78698	-0.095421	
19	3972.14	-1.0228	-0.369719	
20	3773.54	-0.502152	0.239791	
21	3584.86	-1.8457	-0.0503647	
22	3405.62	-1.08982	-0.273053	
23	3235.34	-0.617267	0.0247349	
24	3073.57	-1.04752	-0.337614	
25	2919.89	-1.5269	0.0389685	
26	2773.9	-0.52028	0.21327	
27	2635.2	-1.833	-0.0632583	
28	2503.44	-1.63781	-0.0586307	
29	2378.27	-0.111209	0.0580981	
30	2259.36	-1.6784	-0.0860107	
31	2146.39	-1.34532	0.0837292	
32	2039.07	-0.490493	0.254987	
33	1937.11	-1.68426	-0.0891235	
34	1840.26	-0.014771	0.00108972	
35	1748.25	-0.246223	0.22071	
36	1660.83	-1.40361	0.0988697	
37	1577.79	-1.58947	-0.0171007	
38	1498.9	-1.669	-0.0805302	
39	1423.96	-1.56706	0.0034199	
40	1352.76	-1.15421	-0.162428	
41	1285.12	-1.08274	-0.28453	
42	1220.87	-1.87268	-0.0173999	
43	1159.82	-1.26688	0.0130724	
44	1101.83	-1.38224	0.0970742	



45	1046.74	-1.42576	0.0963493	
46	994.403	-1.46391	0.0824792	
47	944.682	-0.946684	-0.431337	
48	897.448	-1.86004	-0.0337616	
49	852.576	-0.446669	0.298052	
50	809.947	-0.0327158	0.00532305	
51	769.45	-1.66881	-0.0804177	
52	730.977	-1.1207	-0.220896	
53	694.428	-1.37315	0.0950042	
54	659.707	-1.5341	0.0328625	
55	626.722	-1.92132	0.059492	
56	595.386	-1.29283	0.0424605	
57	565.616	-0.45674	0.290188	
58	537.335	-0.610187	0.0403816	
59	510.469	-1.92621	0.0683004	
60	484.945	-1.45152	0.0882119	
61	460.698	-1.45262	0.087749	
62	437.663	-0.0709861	0.024566	
63	415.78	-1.47508	0.0763946	
64	394.991	-1.78161	-0.0976808	
65	375.241	-0.919095	-0.437022	
66	356.479	-1.40623	0.0987994	
67	338.655	-1.19279	-0.0963124	
68	321.723	-1.43016	0.0953431	
69	305.636	-0.0653096	0.0208754	
70	290.355	-1.65142	-0.0688231	
71	275.837	-1.22697	-0.0419387	
72	262.045	-0.647298	-0.0430725	
73	248.943	-0.338328	0.305262	
74	236.496	-1.85607	-0.0385667	
75	224.671	-1.768	-0.102033	
76	213.437	-0.582232	0.100016	
77	202.765	-0.14832	0.0983652	
78	192.627	-1.95862	0.130843	
79	182.996	-1.0734	-0.299305	
80	173.846	-1.54082	0.0270368	
81	165.154	-1.70641	-0.0985743	
82	156.896	-0.108829	0.055786	
83	149.051	-1.25706	0.000551495	
84	141.599	-0.757958	-0.280563	
85	134.519	-1.24931	-0.00982109	
86	127.793	-0.444411	0.299645	
87	121.403	-1.60869	-0.0343419	
88	115.333	-1.3842	0.0974161	
89	109.566	-0.864467	-0.419337	
90	104.088	-0.0631123	0.019522	
91	98.8836	-1.58293	-0.0111345	
92	93.9395	-0.132817	0.0806651	
93	89.2425	-1.73302	-0.104493	
94	84.7804	-1.19443	-0.0935757	
95	80.5413	-1.81622	-0.0776508	
96	76.5143	-1.15824	-0.155399	
97	72.6886	-0.214118	0.180839	
98	69.0541	-0.0650655	0.0207229	
99	65.6014	-1.19163	-0.098244	
100	62.3214	-0.0770287	0.0287963	
101	59.2053	-1.16868	-0.137279	
102	56.245	-1.86883	-0.0225481	
103	53.4328	-1.54631	0.0221891	

104	50.7611	-1.62694	-0.049909	
105	48.2231	-0.514054	0.222762	
106	45.8119	-0.419935	0.312881	
107	43.5213	-1.5584	0.0113251	
108	41.3453	-1.08499	-0.280894	
109	39.278	-1.60729	-0.0331065	
110	37.3141	-1.64061	-0.0608046	
111	35.4484	-1.03507	-0.354399	
112	33.676	-0.584796	0.0947179	
113	31.9922	-1.6223	-0.0460499	
114	30.3926	-1.88684	0.00279298	
115	28.8729	-0.202338	0.165772	
116	27.4293	-0.0643941	0.0203063	
117	26.0578	-0.805445	-0.357841	
118	24.7549	-0.85989	-0.4161	
119	23.5172	-1.17368	-0.128655	
120	22.3413	-0.641743	-0.0304153	
121	21.2243	-0.109561	0.0564934	
122	20.1631	-1.421	0.097254	
123	19.1549	-0.374096	0.318004	
124	18.1972	-1.48723	0.0688755	
125	17.2873	-0.799097	-0.348815	
126	16.4229	-1.93731	0.0889784	
127	15.6018	-0.428419	0.30913	
128	14.8217	-1.80682	-0.0843763	
129	14.0806	-1.28513	0.0343276	
130	13.3766	-0.0377209	0.00706384	
131	12.7078	-1.72192	-0.102792	
132	12.0724	-1.24387	-0.0173375	
133	11.4687	-0.0783105	0.0297328	
134	10.8953	-0.174505	0.130313	
135	10.3505	-1.2761	0.024142	
136	9.83302	-0.347972	0.310039	
137	9.34136	-0.913053	-0.436973	
138	8.8743	-0.876247	-0.42643	
139	8.43058	-0.531938	0.194466	
140	8.00905	-0.796106	-0.344414	
141	7.6086	-1.14914	-0.171284	
142	7.22817	-1.32267	0.0690672	
143	6.86676	-1.77227	-0.100877	
144	6.52342	-1.97711	0.169123	
145	6.19725	-1.22251	-0.048703	
146	5.88739	-1.18137	-0.115512	
147	5.59302	-0.926054	-0.436497	
148	5.31337	-0.926054	-0.436497	
149	5.0477	-1.5941	-0.0213135	
150	4.79532	-0.982086	-0.410487	
151	4.55555	-0.982086	-0.410487	
152	4.32777	-0.775109	-0.310956	
153	4.11138	-1.90686	0.0344847	
154	3.90581	-0.9335	-0.435251	
155	3.71052	-0.746483	-0.258862	
156	3.525	-1.6657	-0.0784738	
157	3.34875	-0.100772	0.0482438	
158	3.18131	-0.972198	-0.417772	
159	3.02224	-0.672323	-0.100148	
160	2.87113	-1.40886	0.0986668	
161	2.72758	-1.37242	0.094803	
162	2.5912	-0.336009	0.303973	

163	2.46164	-1.44963	0.0889879	
164	2.33856	-0.813318	-0.368423	
165	2.22163	-1.47423	0.0768892	
166	2.11055	-1.33769	0.0793352	
167	2.00502	-1.43907	0.0928109	
168	1.90477	-1.3162	0.063989	
169	1.80953	-0.542314	0.176658	
170	1.71905	-0.801111	-0.351726	
171	1.6331	-1.29014	0.0396778	
172	1.55145	-1.82067	-0.0741274	
173	1.47387	-1.83532	-0.0610309	
174	1.40018	-1.27262	0.0200422	
175	1.33017	-0.478347	0.269192	
176	1.26366	-0.407666	0.316738	
177	1.20048	-1.30454	0.0538659	
178	1.14045	-0.603168	0.0557047	
179	1.08343	-1.59581	-0.0228587	
180	1.02926	-0.868435	-0.421926	
181	0.977798	-0.868435	-0.421926	
182	0.928908	-0.361095	0.314976	
183	0.882462	-1.50529	0.056156	
184	0.838339	-1.60442	-0.0305684	
185	0.796422	-0.82046	-0.377416	
186	0.756601	-0.82046	-0.377416	
187	0.718771	-0.82046	-0.377416	
188	0.682833	-1.00479	-0.389776	
189	0.648691	-1.00479	-0.389776	
190	0.616256	-1.00479	-0.389776	
191	0.585444	-1.00479	-0.389776	
192	0.556171	-0.638264	-0.0225044	
193	0.528363	-1.82952	-0.0664913	
194	0.501945	-0.488174	0.25783	
195	0.476847	-0.254402	0.230357	
196	0.453005	-1.22684	-0.0421226	
197	0.430355	-1.78771	-0.0950888	
198	0.408837	-1.78771	-0.0950888	
199	0.388395	-1.14585	-0.177051	
200	0.368975	-1.20627	-0.0742193	
201	0.350527	-1.20627	-0.0742193	
202	0.333	-1.77398	-0.100362	
203	0.31635	-1.73528	-0.104698	
204	0.300533	-1.07498	-0.296826	
205	0.285506	-1.97589	0.166551	
206	0.271231	-0.362072	0.315269	
207	0.257669	-1.43962	0.0926337	
208	0.244786	-0.0945463	0.0427293	
209	0.232547	-0.107059	0.054091	
210	0.220919	-1.83642	-0.0599557	
211	0.209873	-0.626362	0.00440317	
212	0.19938	-0.474685	0.273141	
213	0.189411	-1.96997	0.154161	
214	0.17994	-1.88769	0.00407214	
215	0.170943	-1.25614	-0.00065136	
216	0.162396	-0.939299	-0.433796	
217	0.154276	-0.939299	-0.433796	
218	0.146562	-0.939299	-0.433796	
219	0.139234	-0.939299	-0.433796	
220	0.132272	-0.939299	-0.433796	
221	0.125659	-0.939299	-0.433796	

222	0.119376	-0.939299	-0.433796	
223	0.113407	-0.939299	-0.433796	
224	0.107737	-0.939299	-0.433796	
225	0.10235	-0.939299	-0.433796	
226	0.0972324	-0.939299	-0.433796	
227	0.0923708	-0.939299	-0.433796	
228	0.0877523	-0.939299	-0.433796	
229	0.0833647	-0.939299	-0.433796	
230	0.0791964	-0.939299	-0.433796	
231	0.0752366	-0.939299	-0.433796	
232	0.0714748	-0.939299	-0.433796	
233	0.067901	-0.939299	-0.433796	
234	0.064506	-0.939299	-0.433796	
235	0.0612807	-0.939299	-0.433796	
236	0.0582167	-0.939299	-0.433796	
237	0.0553058	-0.970367	-0.419	
238	0.0525405	-0.970367	-0.419	
239	0.0499135	-0.970367	-0.419	
240	0.0474178	-0.970367	-0.419	
241	0.0450469	-0.970367	-0.419	
242	0.0427946	-0.970367	-0.419	
243	0.0406549	-0.970367	-0.419	
244	0.0386221	-0.970367	-0.419	
245	0.036691	-0.970367	-0.419	
246	0.0348565	-0.970367	-0.419	
247	0.0331136	-0.970367	-0.419	
248	0.031458	-0.970367	-0.419	
249	0.0298851	-0.970367	-0.419	
250	0.0283908	-0.970367	-0.419	
251	0.0269713	-0.970367	-0.419	
252	0.0256227	-0.970367	-0.419	
253	0.0243416	-0.970367	-0.419	
254	0.0231245	-0.970367	-0.419	
255	0.0219683	-0.970367	-0.419	
256	0.0208699	-0.970367	-0.419	
257	0.0198264	-0.932707	-0.435417	
258	0.018835	-0.932707	-0.435417	
259	0.0178933	-0.932707	-0.435417	
260	0.0169986	-0.932707	-0.435417	
261	0.0161487	-0.932707	-0.435417	
262	0.0153413	-0.932707	-0.435417	
263	0.0145742	-0.932707	-0.435417	
264	0.0138455	-0.932707	-0.435417	
265	0.0131532	-0.932707	-0.435417	
266	0.0124956	-0.932707	-0.435417	
267	0.0118708	-0.932707	-0.435417	
268	0.0112772	-0.932707	-0.435417	
269	0.0107134	-0.932707	-0.435417	
270	0.0101777	-0.932707	-0.435417	

-----  
Result: Xmin = -0.932707, Fmin = -0.435417

## Выводы

Из полученных таблиц и графиков видно, что алгоритм имитации отжига является весьма эффективным алгоритмом случайного поиска глобального минимума. Применимость метода не зависит от того, является ли функция унимодальной или мультимодальной.

## Приложение. Исходный код программы

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

using std::cout;

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

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 simulatedAnnealingMethod(const double lower, const double upper,
                             Function func) {
    printTableHead();

    const double T_min = .01;
    double T_i = 10000.;
    double x_i = randomInRange(lower, upper);
    int i = 0;
    while (T_i > T_min) {
        ++i;
        double x_new = randomInRange(lower, upper);
        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 function_variant_4(const double x) {
    return cos(x) * tanh(x);
}
```

```

}

double function_variant_4_with_sin(const double x) {
    return function_variant_4(x) * sin(5*x);
}

const double LOWER = -2.;
const double UPPER = 0.;

int main() {
    cout << "Variant 4.\nFunction 1: cos(x)*th(x), interval: ["
        << LOWER << " " << UPPER << "]\n";
    srand(time(nullptr));
    auto result_1 = simulatedAnnealingMethod(LOWER, UPPER, function_variant_4);
    cout << "Result: Xmin = " << result_1.first
        << ", Fmin = " << result_1.second << '\n';

    cout << "\nFunction 2: cos(x)*th(x)*sin(5*x), interval: ["
        << LOWER << " " << UPPER << "]\n";
    auto result_2 = simulatedAnnealingMethod(LOWER, UPPER,
function_variant_4_with_sin);
    cout << "Result: Xmin = " << result_2.first
        << ", Fmin = " << result_2.second << '\n';

    return 0;
}

```

## **Ответ на контрольный вопрос**

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

Основная идея алгоритма случайного заключается в том, что итеративно выбирается точка, в которой может быть минимум и сравнивается значение функции в нём со значением в предыдущей точке. Если значение меньше, то выполняется переход в новую точку. Если же оно больше, то осуществляется переход в новую точку с некоторой вероятностью, которая зависит от значения параметра-температуры (чем меньше температура, тем меньше вероятность перехода). За счёт этого можно выходить из локальных минимумов.

В настоящее время метод имитации отжига применяется для решения многих оптимизационных задач – финансовых, компьютерной графики, комбинаторных, в телекоммуникационных сетях и многих других. Зачастую метод отжига используют для обучения нейронных сетей.