

组号: 第 1 组



山东师范大学
SHANDONG NORMAL UNIVERSITY

信息科学与工程学院实验报告

《计算智能》

Computational Intelligence

姓名:	任福健
学号:	201911010533
班级:	计工本 1902
导师:	张庆科
时间:	2021 年 4 月 10 日

实验报告 （一）

基本要求：报告正文包含实验目的、实验内容、实验步骤、实验结果（图表）、实验总结五个部分。报告中若涉及算法，请在附录部分给出算法完整源码，报告撰写完毕后请提交 PDF 格式版本到网页版云班课。

一、实验目的

1. 掌握 Matlab 基本脚本程序设计
2. 熟悉 Matlab 典型数据分析方法
3. 熟悉 Matlab 基本绘图函数使用
4. 掌握智能优化算法的性能评估方法

二、实验内容

问题描述：某智能优化算法 A 和算法 B 在求解 Griewank 函数的最小数值时,各自独立重复运行了 30 次，每次算法运行迭代 1000 代，算法测试过程中得到的所有数据分别存放在矩阵 `dataA` 和矩阵 `dataB` 中。`dataA` 和 `dataB` 均为 30 行 1000 列的矩阵，矩阵的每行代表 1 次测试记录，30 行代表 30 次测试记录，每次测试记录里存放了算法循环迭代 1000 代的函数数值结果（由于算法在寻找函数的全局最小值问题，因此这 1000 个数值数据呈现由大到小排列的趋势）。

任务要求：请采用 matlab 模拟生成数据 `dataA` 和 `dataB`（或基于真实算法生成），然后基于这两组数据分析比较算法的性能。

1. 绘制基准测试 Griewank 函数在二维和三维空间的函数图形，分析该函数在不同维度下的特点；
2. 统计分析智能优化算法 A 和 B 在 30 次实验测试中获得的 30 个最优函数值的结果的均值，方差，最大，最小，中位数，数据可采用表格呈现；
3. 绘制算法测试结果的统计箱图，要求图中至少包括图形标题，横坐标标题，纵坐标标题等内容，图片保存为 png 格式；
4. 绘制算法 A 和算法 B 的平均收敛曲线（基于矩阵 `dataA` 和矩阵 `dataB` 内的数据），要求不同曲线使用不同的颜色、线型、和标记符号；
5. 基于算法 A 和算法 B 得到的 30 次最佳实验结果，分析两种算法的求解性能是否具有显著差异性；

提示：可以借助 matlab 里的 `ranksum` 函数或 `ttest` 函数，及矩阵 `dataA` 和矩阵 `dataB` 的最后一列数据进行分析

6. 结合上述实验数据和图表分析两个算法的综合性能差异，按照计算智能课程实验报告模板并撰写带有完整数据和图表的实验报告；
7. 实验报告后面附录里需要附上本次 matlab 脚本程序代码，并用 highlight 代码高亮软件转码后粘贴到 word。

三、实验步骤

1. 绘制基准测试 Griewank 函数在二维和三维空间的函数图形，分析该函数在不同维度下的特点

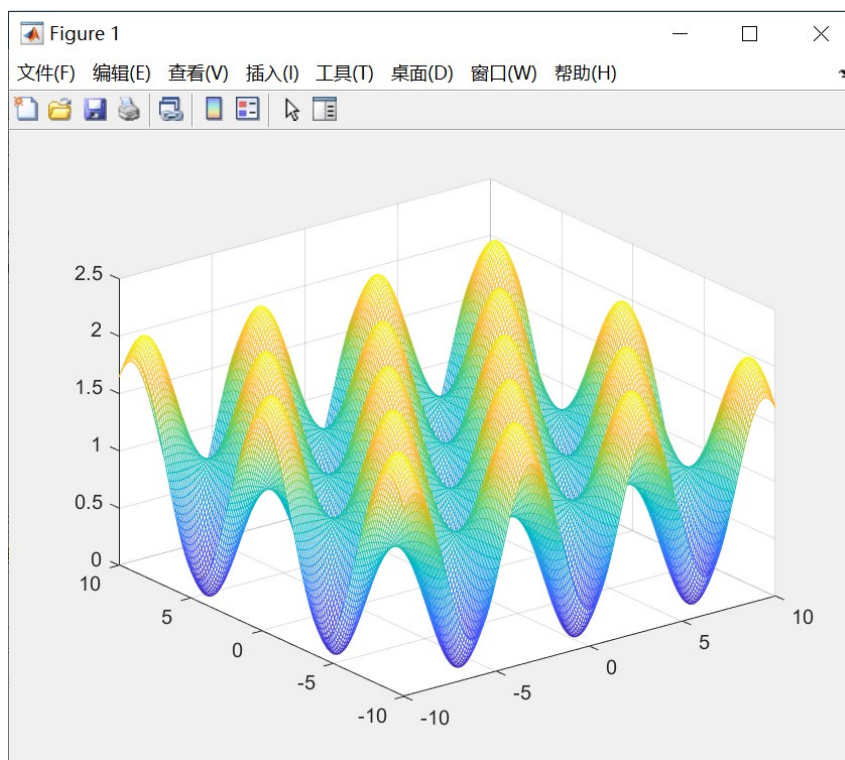


图1：三维空间中的 *Griewank* 函数图形

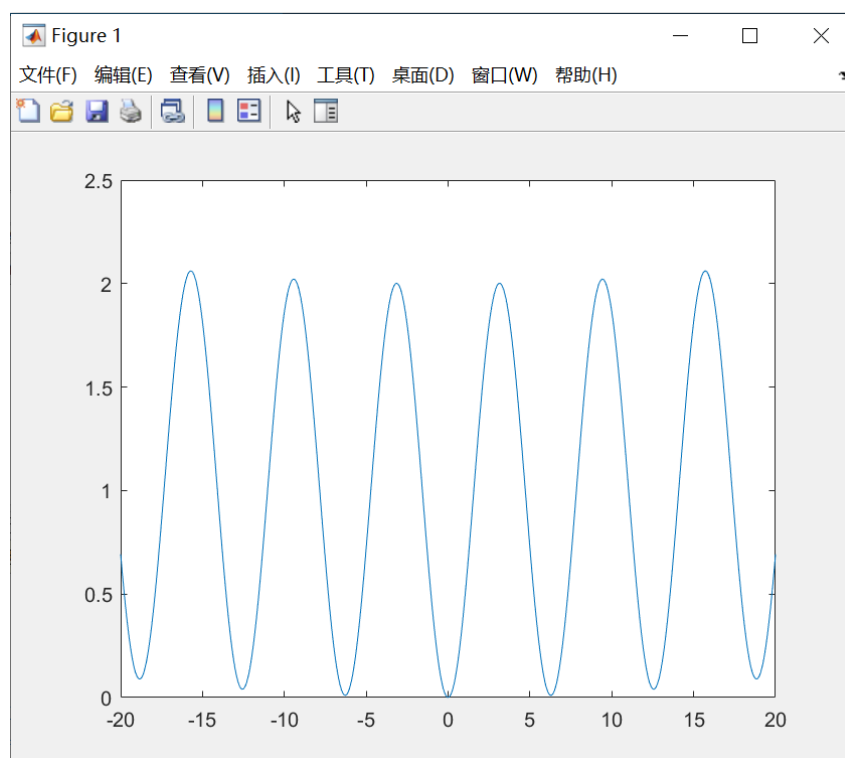
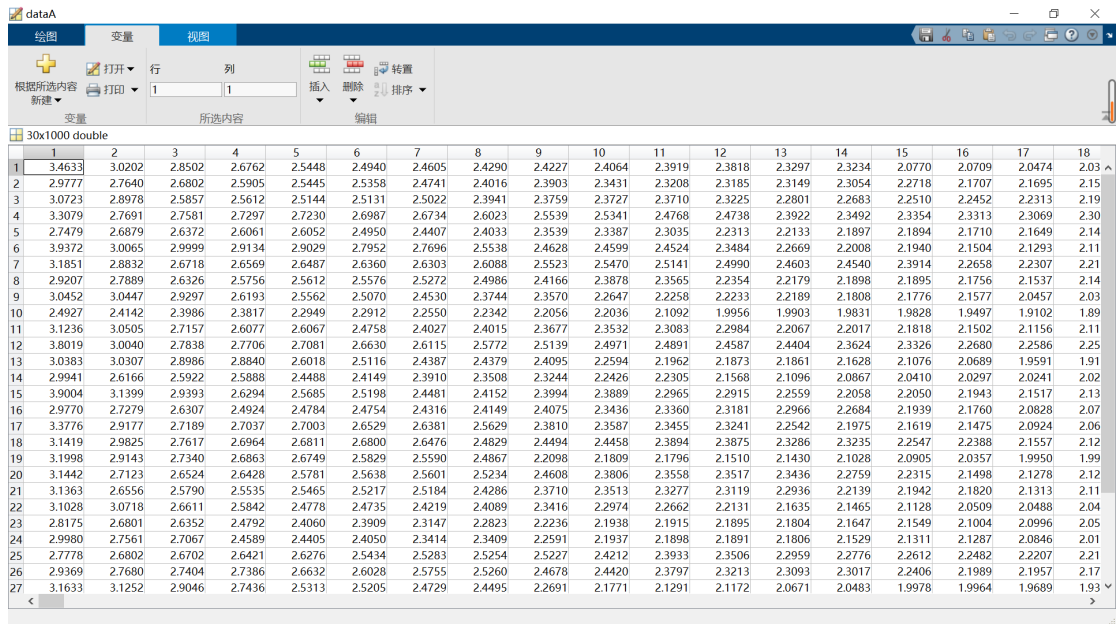


图2：二维空间中的 *Griewank* 函数图形

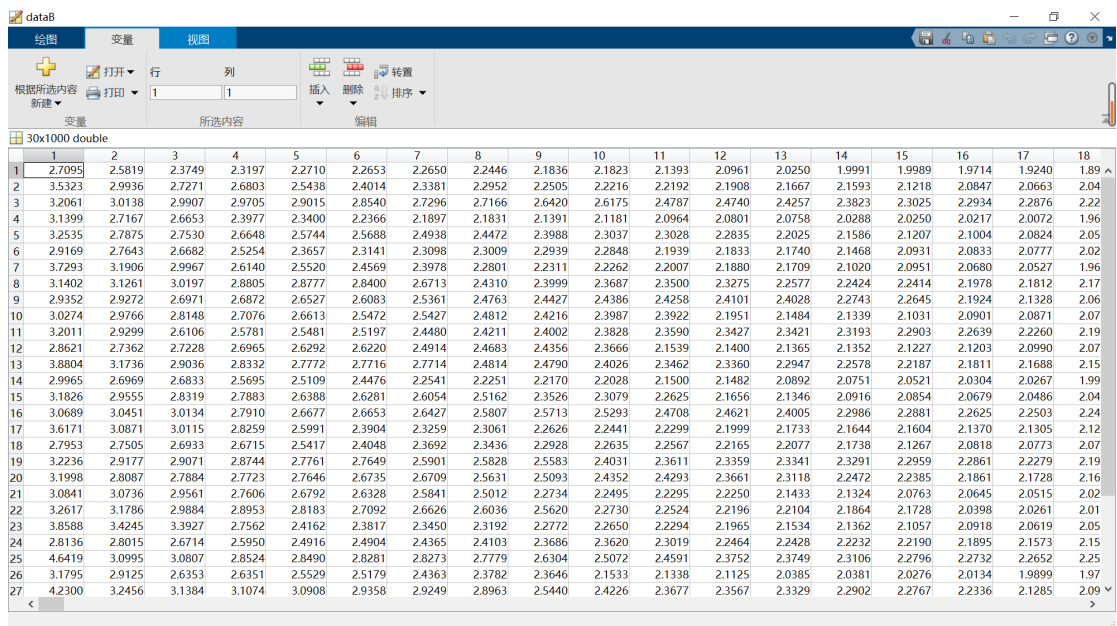
分析：该函数在二维和三维空间内的函数图像广泛的在较小的区域内具有多个峰值即极值。

2. 采用 matlab 模拟生成数据 dataA 和 dataB



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	3.4633	3.0202	2.8502	2.6762	2.5448	2.4940	2.4605	2.4290	2.4227	2.4064	2.3919	2.3818	2.3297	2.3234	2.0770	2.0709	2.0474	2.03
2	2.9777	2.7640	2.6802	2.5905	2.5445	2.5358	2.4741	2.4016	2.3903	2.3431	2.3208	2.3185	2.3149	2.3054	2.2718	2.1707	2.1695	2.15
3	3.0723	2.8978	2.5857	2.5612	2.5144	2.5131	2.5022	2.3941	2.3759	2.3727	2.3710	2.3225	2.2801	2.2683	2.2510	2.2452	2.2313	2.19
4	3.3079	2.7691	2.7581	2.7297	2.7230	2.6987	2.6734	2.6023	2.5539	2.5341	2.4768	2.4738	2.3922	2.3492	2.3354	2.3313	2.3069	2.30
5	2.7479	2.6879	2.6372	2.6061	2.6052	2.4950	2.4407	2.4033	2.3539	2.3387	2.3035	2.2313	2.2133	2.1897	2.1894	2.1710	2.1649	2.14
6	3.9372	3.0065	2.9999	2.9134	2.9029	2.7952	2.7696	2.5538	2.4628	2.4599	2.4524	2.3484	2.2669	2.2008	2.1940	2.1504	2.1293	2.11
7	3.1851	2.8832	2.6718	2.6569	2.6487	2.6360	2.6303	2.6088	2.5523	2.5470	2.5141	2.4990	2.4603	2.4540	2.3914	2.2658	2.2307	2.21
8	2.9207	2.7889	2.6326	2.5756	2.5612	2.5576	2.5272	2.4986	2.4166	2.3878	2.3565	2.2354	2.2179	2.1898	2.1895	2.1756	2.1537	2.14
9	3.0452	3.0447	2.9297	2.6193	2.5562	2.5070	2.4530	2.3744	2.3570	2.2647	2.2258	2.2233	2.2189	2.1808	2.1776	2.1577	2.0457	2.03
10	2.4927	2.4142	2.3986	2.3817	2.2949	2.2912	2.2550	2.2342	2.2056	2.2036	2.1092	1.9956	1.9903	1.9831	1.9828	1.9497	1.9102	1.89
11	3.1236	3.0505	2.7157	2.6077	2.6067	2.4758	2.4027	2.4015	2.3677	2.3532	2.3083	2.2984	2.2067	2.2017	2.1818	2.1502	2.1156	2.11
12	3.8019	3.0040	2.7838	2.7706	2.7081	2.6630	2.6115	2.5772	2.5139	2.4971	2.4891	2.4587	2.4404	2.3624	2.3326	2.2680	2.2586	2.25
13	3.0383	3.0307	2.8986	2.8840	2.6018	2.5116	2.4387	2.4379	2.4095	2.2594	2.1962	2.1873	2.1861	2.1628	2.1076	2.0689	1.9591	1.92
14	2.9941	2.6166	2.5922	2.5888	2.4488	2.4149	2.3910	2.3508	2.3244	2.2426	2.2305	2.1568	2.1096	2.0867	2.0410	2.0297	2.0241	2.02
15	3.9004	3.1399	2.9393	2.6294	2.5685	2.5198	2.4481	2.4152	2.3994	2.3889	2.2965	2.2915	2.2559	2.2058	2.2050	2.1943	2.1517	2.13
16	2.9770	2.7279	2.6307	2.4924	2.4784	2.4754	2.4316	2.4149	2.4075	2.3436	2.3360	2.3181	2.2966	2.2684	2.1939	2.1760	2.0828	2.07
17	3.3776	2.9177	2.7189	2.7037	2.7003	2.6529	2.6381	2.5629	2.3810	2.3587	2.3455	2.3241	2.2542	2.1975	2.1619	2.1475	2.0924	2.06
18	3.1419	2.9825	2.7617	2.6964	2.6811	2.6800	2.6476	2.4829	2.4094	2.4458	2.3894	2.3875	2.3286	2.3235	2.2547	2.2388	2.1557	2.12
19	3.1998	2.9143	2.7340	2.6863	2.6749	2.5829	2.5590	2.4867	2.2998	2.1809	2.1796	2.1510	2.1430	2.1028	2.0905	2.0237	1.9950	1.99
20	3.1442	2.7123	2.6524	2.6428	2.5781	2.5638	2.5601	2.5234	2.4608	2.3806	2.3558	2.3517	2.3436	2.2759	2.2315	2.1498	2.1278	2.12
21	3.1363	2.6556	2.5790	2.5535	2.5465	2.5217	2.5184	2.4286	2.3710	2.3513	2.3277	2.3119	2.2936	2.2139	2.1942	2.1820	2.1313	2.11
22	3.1028	3.0718	2.6611	2.5842	2.4778	2.4735	2.4219	2.4089	2.3416	2.2974	2.2662	2.2131	2.1635	2.1465	2.1128	2.0509	2.0488	2.04
23	2.8175	2.6801	2.6352	2.4792	2.4060	2.3909	2.3147	2.2823	2.2236	2.1938	2.1915	2.1895	2.1804	2.1647	2.1549	2.1004	2.0996	2.05
24	2.9980	2.7561	2.7067	2.4589	2.4405	2.4050	2.3414	2.3409	2.2591	2.1937	2.1898	2.1891	2.1806	2.1529	2.1311	2.1287	2.0846	2.01
25	2.7778	2.6802	2.6702	2.6421	2.6276	2.5434	2.5254	2.5227	2.4212	2.3933	2.3506	2.2959	2.2776	2.2612	2.2482	2.2207	2.2207	2.21
26	2.9369	2.7680	2.7404	2.7386	2.6632	2.6028	2.5755	2.5260	2.4678	2.4420	2.3797	2.3213	2.3093	2.3017	2.2406	2.1989	2.1957	2.17
27	3.1633	3.1252	2.9046	2.7436	2.5313	2.5205	2.4729	2.4495	2.2691	2.1771	2.1291	2.1172	2.0671	2.0483	1.9978	1.9964	1.9689	1.93

图3: dataA 部分数据表格



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	2.7095	2.5819	2.3749	2.3197	2.2710	2.2653	2.2650	2.2446	2.1836	2.1823	2.1393	2.0961	2.0250	1.9991	1.9989	1.9714	1.9240	1.89
2	3.5323	2.9936	2.7271	2.6803	2.5438	2.4014	2.3381	2.2952	2.2505	2.2216	2.2192	2.1908	2.1667	2.1593	2.1218	2.0847	2.0663	2.04
3	3.2061	3.0138	2.9907	2.9705	2.9015	2.8540	2.7296	2.7166	2.6420	2.6175	2.4787	2.4740	2.4257	2.3823	2.3025	2.2934	2.2876	2.22
4	3.1399	2.7167	2.6653	2.3977	2.3400	2.2366	2.1897	2.1831	2.1391	2.1181	2.0964	2.0801	2.0758	2.0288	2.0250	2.0217	2.0072	1.96
5	3.2535	2.7875	2.7530	2.6648	2.5744	2.5688	2.4938	2.4472	2.3988	2.3037	2.3028	2.2835	2.2025	2.1586	2.1207	2.1004	2.0824	2.05
6	2.9169	2.7643	2.6682	2.5254	2.3657	2.3141	2.3098	2.3009	2.2939	2.2848	2.1939	2.1833	2.1740	2.1468	2.0931	2.0833	2.0777	2.02
7	3.7293	3.1906	2.9967	2.6140	2.5520	2.4569	2.3978	2.2801	2.2311	2.2262	2.2007	2.1880	2.1709	2.1020	2.0951	2.0680	2.0527	1.96
8	3.1402	3.1261	3.0197	2.8805	2.8777	2.8400	2.6713	2.4310	2.3999	2.3687	2.3500	2.3275	2.2577	2.2424	2.2414	2.1978	2.1812	2.17
9	2.9352	2.9272	2.6971	2.6872	2.6527	2.6083	2.5361	2.4763	2.4427	2.4386	2.4258	2.4101	2.4028	2.2743	2.2645	2.1924	2.1328	2.06
10	3.0274	2.9266	2.8148	2.7076	2.6613	2.5472	2.5427	2.4812	2.4216	2.3987	2.3922	2.1951	2.1484	2.1339	2.1031	2.0901	2.0871	2.07
11	3.2011	2.9299	2.6106	2.5781	2.5481	2.5197	2.4480	2.4211	2.4002	2.3828	2.3590	2.3427	2.3421	2.3193	2.2903	2.2639	2.2260	2.19
12	2.8621	2.7362	2.7228	2.6965	2.6292	2.6220	2.4914	2.4683	2.4356	2.3666	2.1539	2.1400	2.1365	2.1352	2.1227	2.1203	2.0990	2.07
13	3.8804	3.1736	2.9036	2.8332	2.7772	2.7716	2.7714	2.4814	2.4790	2.4026	2.3462	2.3360	2.2947	2.2578	2.2187	2.1811	2.1688	2.15
14	2.9965	2.6969	2.6833	2.5695	2.5109	2.4476	2.2541	2.2251	2.2170	2.2028	2.1500	2.1482	2.0892	2.0751	2.0521	2.0304	2.0267	1.99
15	3.1826	2.9555	2.8319	2.7883	2.6388	2.6281	2.6054	2.5162	2.3526	2.3079	2.2625	2.1656	2.1346	2.0916	2.0854	2.0679	2.0486	2.04
16	3.0689	3.0451	3.0134	2.7910	2.6677	2.6653	2.6427	2.5807	2.5713	2.5293	2.4708	2.4621	2.4005	2.2986	2.2881	2.2625	2.2503	2.24
17	3.6171	3.0871	3.0115	2.8259	2.5991	2.3904	2.3259	2.3061	2.2626	2.2441	2.2299	2.1999	2.1733	2.1644	2.1604	2.1370	2.1305	2.12
18	2.7953	2.7505	2.6933	2.6715	2.5417	2.4048	2.3692	2.3436	2.2928	2.2635	2.2567	2.2165	2.2077	2.1738	2.1267	2.0818	2.0773	2.07
19	3.2236	2.9177	2.9071	2.8744	2.7761	2.7649	2.5901	2.5828	2.5583	2.4031	2.3611	2.3359	2.3341	2.3291	2.2959	2.2861	2.2279	2.19
20	3.1998	2.8087	2.7884	2.7723	2.7646	2.6735	2.6709	2.5631	2.5093	2.4352	2.4293	2.3661	2.3118	2.2472	2.2385	2.1861	2.1728	2.16
21	3.0841	3.0736	2.9561	2.7606	2.6792	2.6328	2.5841	2.5012	2.2734	2.2495	2.2295	2.2250	2.1433	2.1324	2.0763	2.0645	2.0515	2.02
22	3.2617	3.1786	2.9884	2.8953	2.8183	2.7092	2.6626	2.6036	2.5620	2.2730	2.2524	2.2196	2.2104	2.1864	2.1362	2.1057	2.0918	2.01
23	3.8588	3.4245	3.3927	2.7562	2.4162	2.3817	2.3450	2.3192	2.2772	2.2650	2.2294	2.1965	2.1534	2.1362	2.1057	2.0776	2.0732	2.25
24	2.8136	2.8015	2.6714	2.5950	2.4916	2.4904	2.4365	2.4103	2.3686	2.3620	2.3019	2.2464	2.2428	2.2232	2.2190	2.1895	2.1573	2.15
25	4.6419	3.0995	3.0807	2.8524	2.8490	2.8281	2.8273	2.7779	2.6304	2.5072	2.4591	2.3752	2.3749	2.3106	2.2796	2.2732	2.2652	2.25
26	3.1795	2.9125	2.6353	2.6351	2.5529	2.5179	2.4363	2.3782	2.3646	2.1533	2.1338	2.1125	2.0385	2.0381	2.0276	2.0134	1.9899	1.97
27	4.2300	3.2456	3.1384	3.1074	3.0908	2.9358	2.9249	2.8963	2.5440	2.4226	2.3677	2.3567	2.3329	2.2902	2.2767	2.2336	2.1285	2.09

图4: dataB 部分数据表格

3. 统计分析智能优化算法 A 和 B 在 30 次实验测试中获得的 30 个最优函数值的结果的均值，方差，最大，最小，中位数，数据可采用表格呈现；

名称 ^	值
dataA	30x1000 double
dataB	30x1000 double
maxA	-2.6944
maxB	-2.6201
meanA	-3.3211
meanB	-3.2061
midA	-3.2374
midB	-3.2076
minA	-4.0509
minB	-4.0204
sumA	-99.6338
sumB	-96.1824
varA	0.1196
varB	0.1314

图5：均值、方差、最大、最小、中位数表格

分析：算法 B 的 30 次最优函数值的均值、最大、最小、中位数值比算法 A 更优，但算法 A 的 30 次最优函数值的方差更小一点，更稳定。

4. 绘制算法测试结果的统计箱图，要求图中至少包括图形标题，横坐标标题，纵坐标标题等内容，图片保存为 png 格式

Ps:因数据过多，仅做后十次迭代数据的统计箱图。

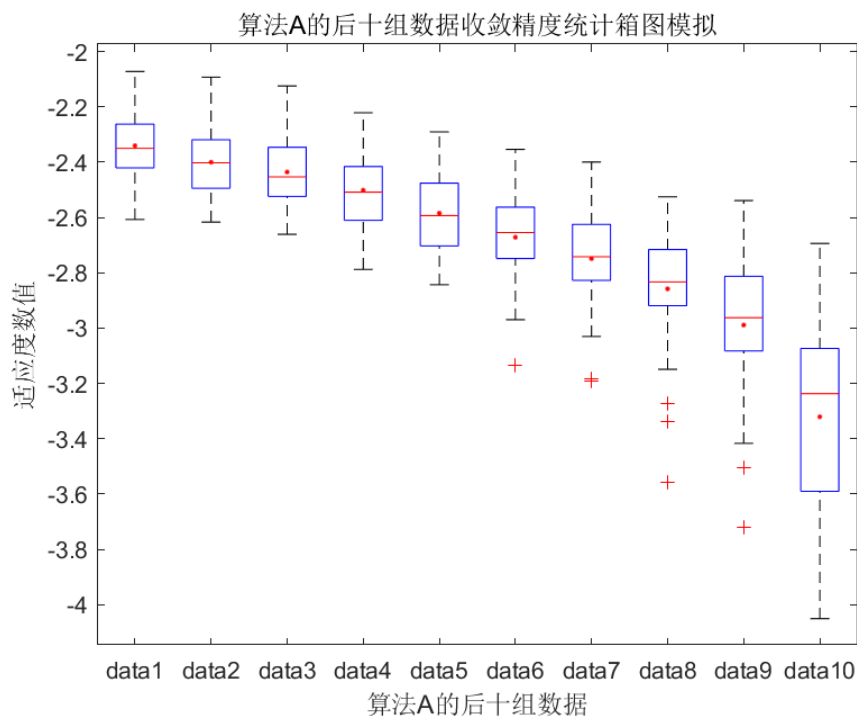


图6：算法 A 后十组数据收敛精度统计箱图模拟

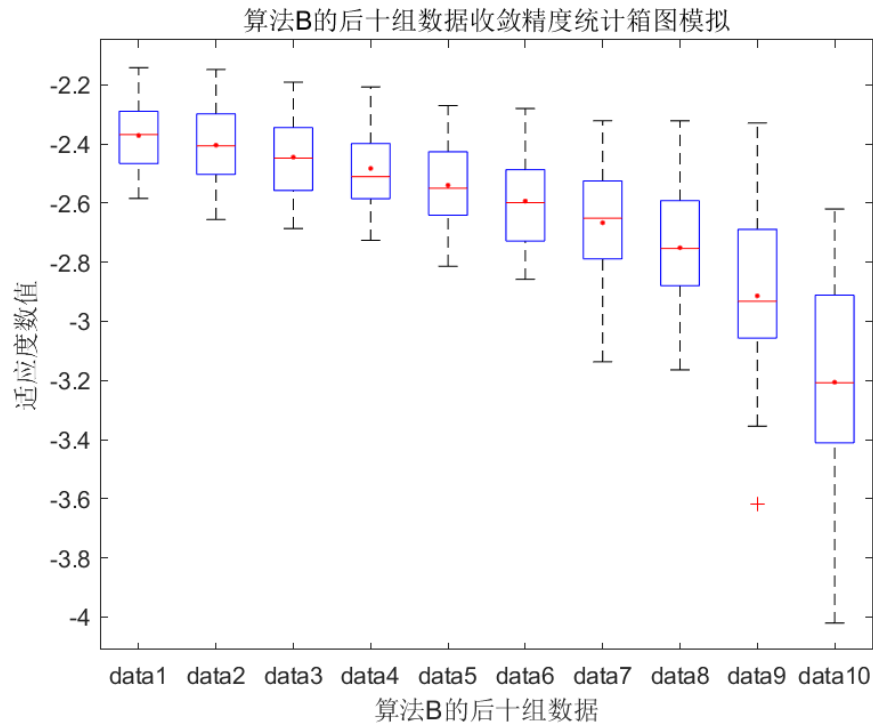


图7：算法B 后十组数据收敛精度统计箱图模拟

5.绘制算法 A 和算法 B 的平均收敛曲线（基于矩阵 dataA 和矩阵 dataB 内的数据），要求不同曲线使用不同的颜色、线型、和标记符号

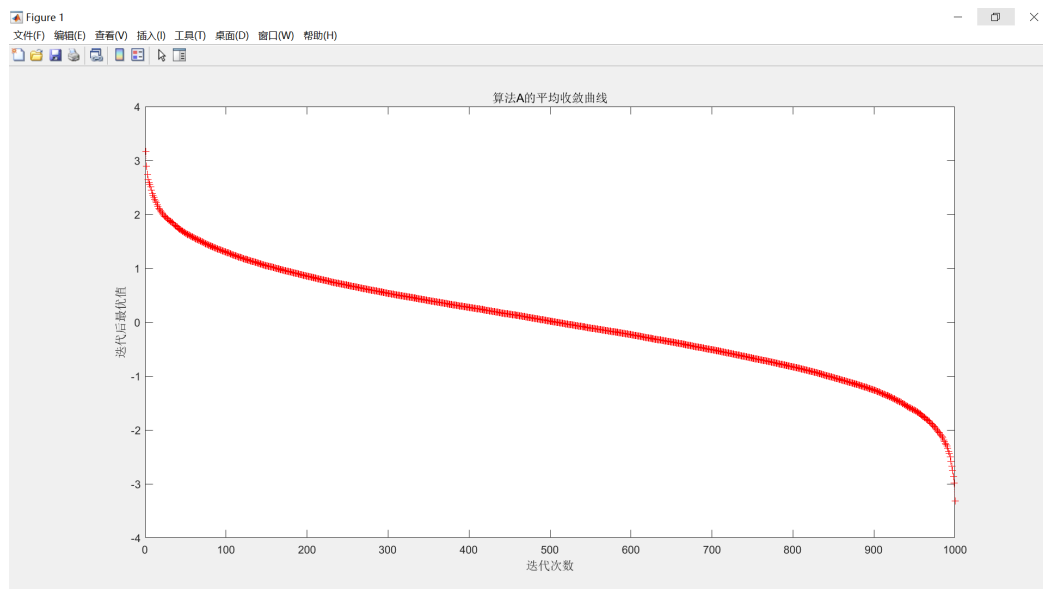


图8：算法A 的平均收敛曲线

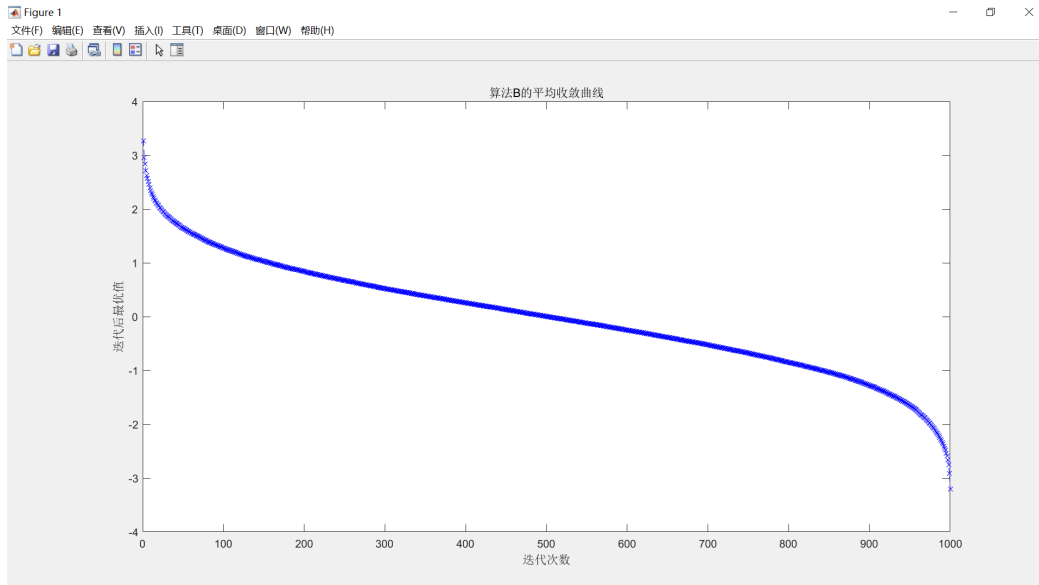


图9：算法B的平均收敛曲线

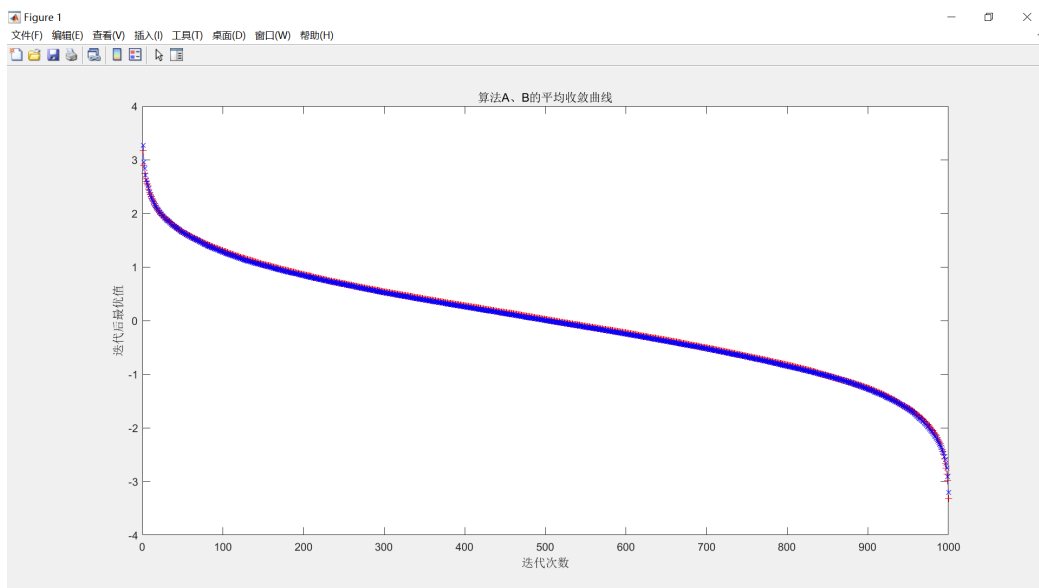


图10：算法A、B的平均收敛曲线对比图

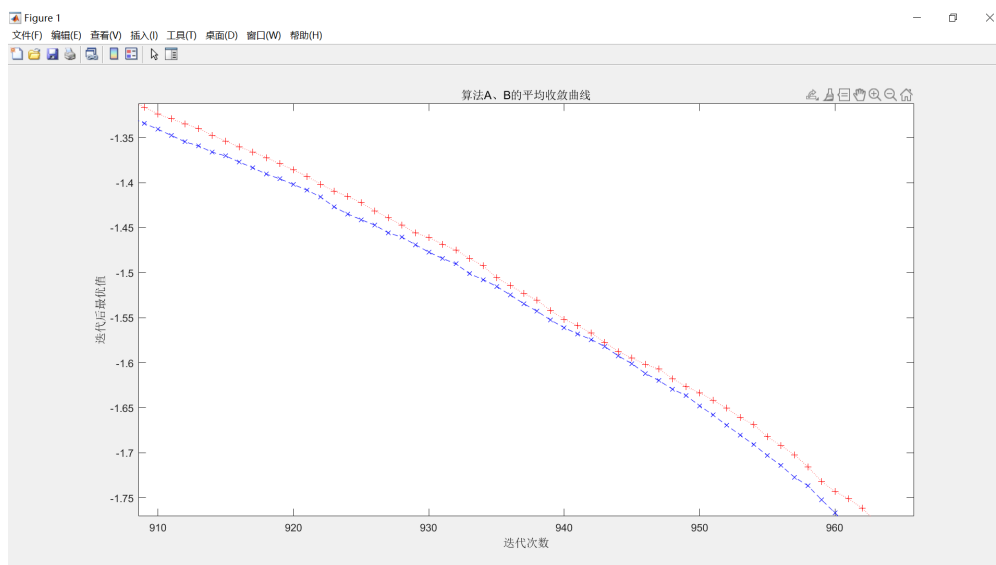


图11：算法A、B的平均收敛曲线部分细节放大图

分析：算法A、B的平均收敛曲线高度吻合，前期和后期的斜率较高，符合生成数据时正态分布的情况。

6. 基于算法A和算法B得到的30次最佳实验结果，分析两种算法的求解性能是否具有显著差异性；

提示：可以借助matlab里的ranksum函数或ttest函数，及矩阵dataA和矩阵dataB的最后一列数据进行分析

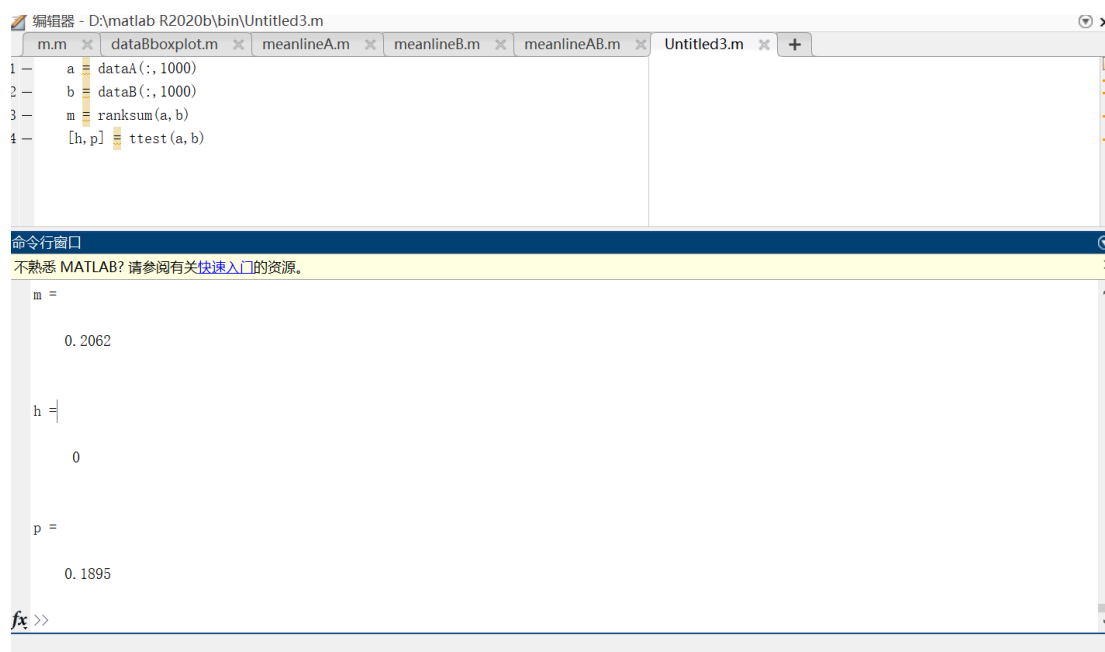


图12：算法A、B的显著差异性判断

分析：因 $m > 0.05$, $h = 0$, 可以证明算法A、B无显著差异性

四、实验结果与分析

实验结果:

综合上述实验过程及分析, 得到以下结论:

算法 A、B 均值、方差均无明显差别, 且平均收敛曲线迭代吻合度极高。

算法 A、B 无显著差异性。

原因分析:

算法 A、B 的实验数据均为 matlab 的 randn() 函数自动生成的符合正态分布的矩阵, 故得此结果。

五、实验总结

通过此实验, 我粗略的了解了 matlab 的使用方式。学会了自行上网查找问题的解决方法。同时, 我明白学习不能是一蹴而就的, 要有耐心的不断前行。

Just do it !

附录: 程序源码(借助 Highlight 软件导入带有行号的代码, 添加必要注释信息)

1. 绘制 Griewank 函数图像

三维空间

```
01 x=[ -10:0.1 : 10 ];
02 y =[ -10 :0.1 : 10 ];
03 [X,Y] = meshgrid(x,y);
04 z=1+(1/4000.0)*X.^2+(1/4000.0)*Y.^2;
05 z=z-cos(X).*(cos(1/2.0*sqrt(2)).*Y);
06 mesh(X,Y,z)
```

二维空间

```
01 x=[-20:0.1:20];
02 y=1+(1/4000.0)*x.^2-cos(x);
03 plot(x,y)
```

2. 模拟数据, 制作每行从大到小排列的30*1000 的矩阵dataA 和矩阵dataB

```
01 dataA = randn(30,1000)
```

```
02 dataB = randn(30,1000)
03 dataA = sort(dataA,2,'descend')
04 dataB = sort(dataA,2,'descend')
```

3. 统计分析智能优化算法 A 和 B 在 30 次实验测试中获得的 30 个最优函数值的结果的均值，方差，最大，最小，中位数，数据可采用表格呈现；

```
01 %1.求30个最优值的平均值
02 meanA = mean(dataA(:,1000))
03 meanB = mean(dataB(:,1000))
04 %2.求30个最优值的方差
05 varA = var(dataA(:,1000))
06 varB = var(dataB(:,1000))
07 %3.求30个最优值的最大值
08 maxA = max(dataA(:,1000))
09 maxB = max(dataB(:,1000))
10 %4.求30个最优值的最小值
11 minA = min(dataA(:,1000))
12 minB = min(dataB(:,1000))
13 %5.求30个最优值的中位数
14 midA = median(dataA(:,1000))
15 midB = median(dataB(:,1000))
```

4. 绘制算法测试结果的统计箱图，要求图中至少包括图形标题，横坐标标题，纵坐标标题等内容，图片保存为 png 格式(后十组数据)

算法 A

```
01 %1.绘制算法A的后十组数据收敛精度统计箱图
02 data_A = dataA(:,991:1000);
03 boxplot(data_A);
04 hold on
05 %2.绘制算法A的后十组数据的平均数
06 means = mean(data_A);
07 for i = 1:10
08 plot(i,means(i),'r.')
09 end
10 %3. 显示图形标题，横坐标标题，纵坐标标题等内容
11 title('算法A的后十组数据收敛精度统计箱图模拟');
```

```
12 xlabel('算法A的后十组数据');
13 ylabel('适应度数值');
14 xticklabels({'data1','data2','data3','data4','data5','data6',
15             'data7','data8','data9','data10'});
16 %4.保存图片为png格式
17 saveas(gcf,'dataA-box-plot','png');
```

算法 B

```
01 %1.绘制算法B的后十组数据收敛精度统计箱图
02 data_B = dataB(:,991:1000);
03 boxplot(data_B);
04 hold on
05 %2.绘制算法B的后十组数据的平均数
06 means = mean(data_B);
07 for i = 1:10
08 plot(i,means(i),'r.')
09 end
10 %3. 显示图形标题，横坐标标题，纵坐标标题等内容
11 title('算法B的后十组数据收敛精度统计箱图模拟');
12 xlabel('算法B的后十组数据');
13 ylabel('适应度数值');
14 xticklabels({'data1','data2','data3','data4','data5','data6',
15             'data7','data8','data9','data10'});
16 %4.保存图片为png格式
17 saveas(gcf,'dataB-box-plot','png');
```

5. 绘制算法 A、B 的平均收敛曲线

算法 A

```
01 %绘制算法A的平均收敛曲线
02 meansA = mean(dataA);
03 x=[1:1: 1000 ];
04 plot(x,meansA,'r:+')
05 %显示图形标题，横坐标标题，纵坐标标题等内容
06 title('算法A的平均收敛曲线');
07 xlabel('迭代次数');
08 ylabel('迭代后最优值');
```

算法 B

```
01 %绘制算法B的平均收敛曲线
02 meansB = mean(dataB);
03 x=[1:1: 1000 ];
04 plot(x,meansB,'b--x')
05 %显示图形标题，横坐标标题，纵坐标标题等内容
06 title('算法B的平均收敛曲线');
07 xlabel('迭代次数');
08 ylabel('迭代后最优值');
```

算法 A、B

```
01 %绘制算法A、B的平均收敛曲线
02 meansA = mean(dataA);
03 meansB = mean(dataB);
04 x=[1:1: 1000 ];
05 %绘制算法A的平均收敛曲线
06 plot(x,meansA,'r:+')
07 hold on
08 %绘制算法B的平均收敛曲线
09 plot(x,meansB,'b--x')
10 %显示图形标题，横坐标标题，纵坐标标题等内容
11 title('算法A、B的平均收敛曲线');
12 xlabel('迭代次数');
13 ylabel('迭代后最优值');
```

6. 显著差异性分析

```
01 a = dataA(:,1000)
02 b = dataB(:,1000)
03 m = ranksum(a,b)
04 [h,p] = ttest(a,b)
```