

Benchmark Test Results

Comparing 2 methods of character mutation

Method 1

Chooses one bit randomly if random chance falls into the probability that the string should mutate, if `random.random() < probability_of_mutation`.

Method 2

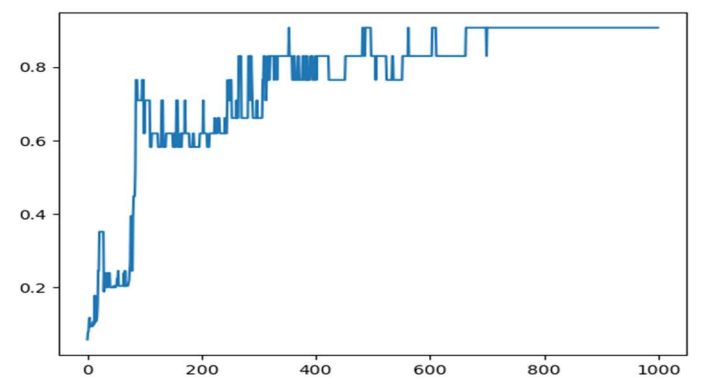
Bit by bit, perform mutation if random chance falls into the probability that the bit should mutate, if `random.random() < probability_of_mutation`.

All graphs are plotted as Fitness Measures (Y-axis [0, 1]) against Generations (X-axis [0, 1000]).

Several tests have shown that method 1 is capable of occasionally outperforming method 2, method 2 is far more consistent and is therefore more reliable.

De Jong

Method 1



Running Simple Genetic Algorithm on De Jong Sphere benchmark function

Tested population size: 40 Number of generations: 1000

Generation	Strongest Candidate	Fitness
0	[-1.9200, 1.9200, -1.9200, 2.2400,]	0.05855898060526562
1	[-1.6000, 1.6000, 2.5600, -0.3200,]	0.07827175954915469
2	[-1.9200, 1.9200, -1.9200, 0.6400,]	0.08020017964840241
3	[-0.3200, -1.9200, -1.9200, 0.6400,]	0.11255177381595535
4	[-0.3200, -1.9200, -1.9200, 0.0000,]	0.11799131583915423
5	[-1.6000, -1.6000, -1.9200, 0.9600,]	0.09321401938851603
6	[-1.2800, 1.9200, -1.9200, 0.0000,]	0.09988812529966437
7	[-1.2800, 2.2400, -1.6000, -0.3200,]	0.0969142502713599
9	[-1.2800, 0.6400, 2.2400, -1.6000,]	0.09411233248004816
10	[-0.6400, 0.6400, 2.2400, -1.6000,]	0.10641920653839604
12	[-2.5600, 0.6400, 0.6400, 1.2800,]	0.09988812529966438
13	[-1.2800, 0.9600, -0.6400, 1.2800,]	0.17831669044222537
14	[-1.2800, 0.9600, 0.9600, -2.2400,]	0.10527202290719216
15	[-0.6400, 1.6000, -1.6000, 1.6000,]	0.11001584228128854
17	[-0.6400, 0.9600, 0.9600, -2.2400,]	0.12091313600309538
18	[-1.2800, 0.9600, 0.6400, 1.6000,]	0.15314873805439846
19	[-1.2800, 0.9600, 0.6400, 0.3200,]	0.24557956777996065
21	[-0.6400, 0.9600, 0.6400, 0.3200,]	0.35171637591446264

For generation 21 to 28, the max fitness level was 0.3517.

29	[-0.6400, 0.9600, 0.6400, 1.6000,]	0.18865076969514044
----	------------------------------------	---------------------

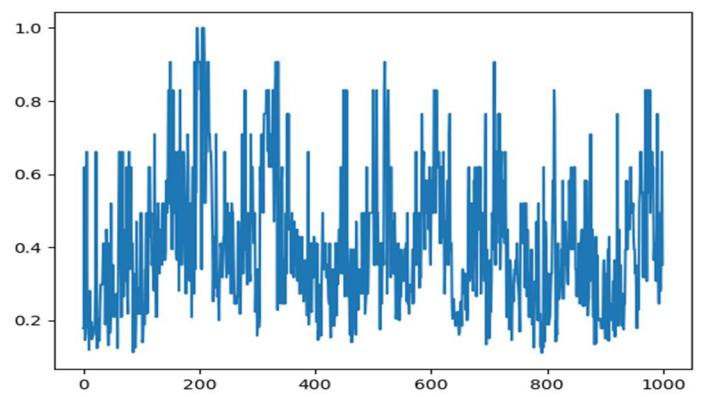
...

552	[-0.3200, 0.0000, 0.3200, 0.0000,]	0.8300132802124832
For generation 552 to 561, the max fitness level was 0.8300.		
562	[-0.3200, 0.0000, 0.0000, 0.0000,]	0.9071117561683597
563	[-0.3200, 0.0000, 0.3200, 0.0000,]	0.8300132802124832
For generation 563 to 603, the max fitness level was 0.8300.		
604	[-0.3200, 0.0000, 0.0000, 0.0000,]	0.9071117561683597
For generation 604 to 610, the max fitness level was 0.9071.		
611	[-0.3200, 0.0000, 0.3200, 0.0000,]	0.8300132802124832
For generation 611 to 662, the max fitness level was 0.8300.		
663	[-0.3200, 0.0000, 0.0000, 0.0000,]	0.9071117561683597
For generation 663 to 698, the max fitness level was 0.9071.		
699	[-0.3200, 0.0000, 0.3200, 0.0000,]	0.8300132802124832
700	[-0.3200, 0.0000, 0.0000, 0.0000,]	0.9071117561683597
For generation 700 to 998, the max fitness level was 0.9071.		
999	[-0.3200, 0.0000, 0.0000, 0.0000,]	0.9071117561683597

Highest fitness ached by:
('Gen: 353', [-0.3200000000000003, 0.0, 0.0, 0.0], 0.9071117561683597)

De Jong benchmark test complete

Method 2



Running Simple Genetic Algorithm on De Jong Sphere benchmark function

Tested population size: 40 Number of generations: 1000

Generation	Strongest Candidate	Fitness
0	[0.6400, -1.2800, 0.0000, 1.6000,]	0.17831669044222542
1	[1.2800, -0.6400, 0.0000, 1.6000,]	0.17831669044222542
2	[0.6400, -0.3200, 0.0000, 0.3200,]	0.6194251734390486
3	[0.6400, -1.2800, 1.9200, 0.3200,]	0.1462672595366253
4	[-0.6400, -0.3200, -0.9600, 1.9200,]	0.16339869281045755
5	[1.2800, 0.6400, 0.0000, 0.9600,]	0.25191455058444173
6	[0.0000, -0.6400, 0.0000, 0.3200,]	0.6613756613756615
7	[0.0000, -0.6400, 1.9200, 0.3200,]	0.19236688211757466
8	[0.0000, -1.9200, -0.6400, -0.6400,]	0.18163324614937523
9	[0.0000, -1.9200, -0.6400, -0.3200,]	0.19236688211757466
10	[0.0000, -2.5600, 0.6400, -0.6400,]	0.11943435887636158
11	[0.6400, 0.6400, -1.2800, 0.3200,]	0.2808988764044944
12	[0.0000, -1.9200, 0.6400, -0.9600,]	0.1661792076575379
13	[0.9600, -1.9200, 0.0000, -0.9600,]	0.15314873805439844
14	[0.0000, -1.9200, 0.0000, -0.6400,]	0.19623233908948196
15	[1.2800, -1.9200, 0.0000, -0.6400,]	0.14849132810643859
16	[0.6400, -1.9200, 0.0000, -0.6400,]	0.18163324614937523
17	[1.2800, -1.2800, 1.2800, -0.6400,]	0.15810776625347833
18	[1.2800, -1.2800, 1.2800, -0.3200,]	0.16617920765753783
19	[0.6400, -1.9200, -0.6400, -0.3200,]	0.17831669044222542
20	[-0.6400, 0.6400, -1.2800, 1.2800,]	0.19623233908948196
21	[-0.6400, -1.2800, 0.0000, -0.6400,]	0.28921795465062466

...

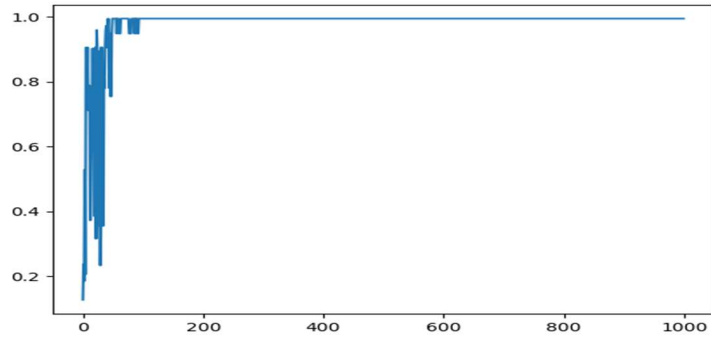
966	[0.0000, -0.6400, 0.3200, -1.2800,]	0.3174200101574403
967	[0.0000, -0.3200, 0.6400, -0.6400,]	0.5203996669442134
968	[1.2800, -0.6400, 0.0000, -0.3200,]	0.31742001015744026
969	[0.0000, -0.3200, 0.0000, -0.6400,]	0.6613756613756615
970	[0.0000, -0.3200, 0.0000, -0.3200,]	0.8300132802124832
971	[0.6400, -0.3200, 0.3200, -0.3200,]	0.5824790307548928
972	[1.2800, -0.6400, 0.3200, -0.3200,]	0.307427447122479
973	[0.0000, 0.0000, 0.3200, -0.3200,]	0.8300132802124832
974	[0.0000, -0.3200, 0.0000, -0.3200,]	0.8300132802124832
975	[0.0000, -0.6400, 0.0000, -0.6400,]	0.5496921723834656
976	[0.0000, -0.3200, 0.6400, -0.3200,]	0.6194251734390486
977	[0.0000, -0.3200, 0.0000, -0.3200,]	0.8300132802124832
978	[0.6400, -0.3200, 0.6400, -0.3200,]	0.4940711462450594
979	[0.0000, -0.3200, 0.3200, -0.6400,]	0.6194251734390486
981	[0.0000, -0.3200, 0.0000, -0.6400,]	0.6613756613756615
982	[0.0000, -0.3200, 0.3200, -0.6400,]	0.6194251734390486
984	[0.0000, -0.9600, 0.6400, -0.6400,]	0.3648569760653825
985	[0.0000, -0.3200, 0.9600, -0.6400,]	0.410913872452334
986	[0.0000, -1.2800, 0.0000, -0.3200,]	0.36485697606538225
988	[0.0000, -0.9600, 0.9600, -0.6400,]	0.3074274471224791
989	[0.6400, 0.6400, 0.0000, -0.6400,]	0.4486719310839916
990	[0.0000, -0.3200, 0.3200, -0.3200,]	0.7649938800489593
991	[1.2800, -0.3200, 0.0000, -0.6400,]	0.31742001015744026
993	[1.2800, -0.3200, 0.6400, -0.9600,]	0.24557956777996065
994	[0.3200, -0.3200, 0.6400, -0.6400,]	0.4940711462450594
995	[0.0000, -0.3200, 0.6400, -0.9600,]	0.410913872452334
996	[1.2800, -0.6400, 0.3200, -0.6400,]	0.28089887640449435
997	[0.0000, -0.6400, 0.0000, -0.9600,]	0.42896362388469467
998	[0.0000, -0.6400, 0.0000, -0.3200,]	0.6613756613756615
999	[0.6400, -0.3200, 0.6400, -0.9600,]	0.35171637591446264

Highest fitness ached by:
('Gen: 196', [0.0, 0.0, 0.0, 0.0], 1.0)

De Jong benchmark test complete

Himmelblau

Method 1



Running Simple Genetic Algorithm on Himmelblau benchmark function

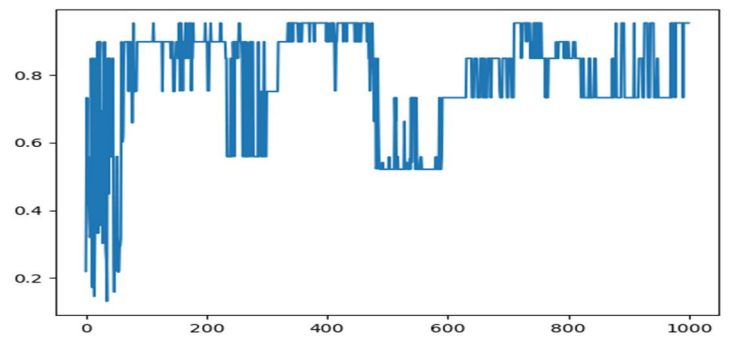
Tested population size: 100 Number of generations: 1000

Generation	Strongest Candidate	Fitness
0	[3.4219, -0.9844,]	0.12685092257472316
1	[3.4219, -2.1562,]	0.23576390181437498
2	[-2.4375, 3.0000,]	0.18418870798658832
3	[3.0469, 1.7344,]	0.5281303795684011
4	[-2.4375, 3.0938,]	0.2049805991176092
5	[-2.7656, 3.0938,]	0.9057085681453921
7	[-2.7656, 3.0469,]	0.7546138259377453
8	[-2.7656, 3.0938,]	0.9057085681453921
9	[-2.9062, 3.0938,]	0.7123313727807917
10	[3.5625, -1.9688,]	0.7879095273411465
12	[3.5625, -2.1562,]	0.3724650854831554
14	[3.5625, -2.0625,]	0.5542531418616059
15	[3.5625, -2.0156,]	0.6684277361309007
16	[3.6094, -1.7812,]	0.9024394166606019
17	[3.5625, -2.0156,]	0.6684277361309007
19	[3.6094, -1.7812,]	0.9024394166606019
20	[3.6094, -2.1562,]	0.38445216363670154
21	[3.6094, -1.9219,]	0.9066419898073936
22	[3.6094, -2.2031,]	0.31530059654322834
23	[3.6094, -1.8281,]	0.9595585892857286
24	[3.6094, -1.7812,]	0.9024394166606019
25	[3.6094, -2.0156,]	0.6887587173430882
26	[3.5625, -1.4531,]	0.35525716342591057

...

57	[-2.8125, 3.1406,]	0.9948443921995289
For generation 57 to 59, the max fitness level was 0.9948.		
60	[-2.7656, 3.1406,]	0.9487880473133589
For generation 60 to 62, the max fitness level was 0.9488.		
63	[-2.8125, 3.1406,]	0.9948443921995289
For generation 63 to 76, the max fitness level was 0.9948.		
77	[-2.7656, 3.1406,]	0.9487880473133589
For generation 77 to 80, the max fitness level was 0.9488.		
81	[-2.8125, 3.1406,]	0.9948443921995289
For generation 81 to 85, the max fitness level was 0.9948.		
86	[-2.7656, 3.1406,]	0.9487880473133589
87	[-2.8125, 3.1406,]	0.9948443921995289
89	[-2.7656, 3.1406,]	0.9487880473133589
90	[-2.8125, 3.1406,]	0.9948443921995289
91	[-2.7656, 3.1406,]	0.9487880473133589
93	[-2.8125, 3.1406,]	0.9948443921995289
For generation 93 to 998, the max fitness level was 0.9948.		
999	[-2.8125, 3.1406,]	0.9948443921995289
Highest fitness achieved by: ('Gen: 42', [-2.8125, 3.140625], 0.9948443921995289)		
Himmelblau benchmark test complete		

Method 2



Running Simple Genetic Algorithm on Himmelblau benchmark function

Tested population size: 100 Number of generations: 1000

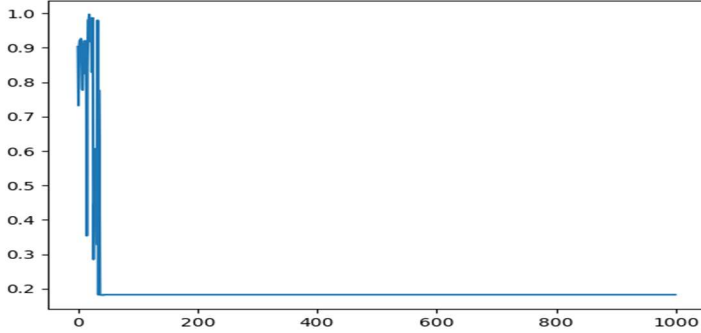
Generation	Strongest Candidate	Fitness
0	[3.1875, 2.2500,]	0.21972996442665219
1	[-3.6562, -3.1875,]	0.521534039504397
2	[-3.7500, -3.1875,]	0.7333848099282685
3	[-3.7500, -3.0938,]	0.41835122118981877
4	[-3.7500, -3.1406,]	0.5580541659818725
5	[-2.7656, 2.9531,]	0.44446231772630235
6	[3.0000, 2.2969,]	0.3682727405489496
7	[3.1406, 2.2031,]	0.3198419604143811
8	[3.2344, 1.9219,]	0.3426966317380834
9	[-3.7500, -3.3281,]	0.8495425659298472
10	[3.0000, 2.2031,]	0.5649189289312718
11	[-3.0469, 2.8594,]	0.17223202916858715
12	[-2.8125, 2.9531,]	0.452451961679104
13	[-3.7500, -3.3281,]	0.8495425659298472
14	[2.5312, 2.2031,]	0.14580124120040613
15	[3.7031, -1.5469,]	0.3147154801968908
16	[3.6562, -2.1094,]	0.43676270558425573
17	[-3.7500, -3.3281,]	0.8495425659298472
18	[3.4688, -2.1094,]	0.3329600109200428
19	[-3.7500, -3.3281,]	0.8495425659298472
20	[-3.7500, -3.2344,]	0.8985072610951793
21	[3.4688, -2.1094,]	0.3329600109200428
22	[-3.7500, -3.2344,]	0.8985072610951793

...

For generation 936 to 941, the max fitness level was 0.9541.		
942	[-3.7500, -3.1875,]	0.7333848099282685
For generation 942 to 946, the max fitness level was 0.7334.		
947	[-3.7500, -3.3281,]	0.8495425659298472
949	[-3.7500, -3.1875,]	0.7333848099282685
For generation 949 to 969, the max fitness level was 0.7334.		
970	[-3.7500, -3.2812,]	0.9540826957320063
971	[-3.7500, -3.1875,]	0.7333848099282685
972	[-3.7500, -3.2812,]	0.9540826957320063
For generation 972 to 976, the max fitness level was 0.9541.		
977	[-3.7500, -3.1875,]	0.7333848099282685
978	[-3.7500, -3.2812,]	0.9540826957320063
For generation 978 to 989, the max fitness level was 0.9541.		
990	[-3.7500, -3.1875,]	0.7333848099282685
991	[-3.7500, -3.2812,]	0.9540826957320063
For generation 991 to 998, the max fitness level was 0.9541.		
999	[-3.7500, -3.2812,]	0.9540826957320063
Highest fitness achieved by: ('Gen: 79', [-3.75, -3.28125], 0.9540826957320063)		
Himmelblau benchmark test complete		

Rosenbrock

Method 1



```
Running Simple Genetic Algorithm on Rosenbrock benchmark function
Tested population size: 100 Number of generations: 1000

Generation Strongest Candidate Fitness
=====
0 [1.2031, 1.4219, ] 0.9033636766002906
1 [0.9531, 0.9688, ] 0.7321505844648348
2 [0.6719, 0.4531, ] 0.9025612245710402
3 [1.2500, 1.5781, ] 0.9200359389038635
4 [1.2500, 1.5469, ] 0.9200359389038635
5 [0.7188, 0.5156, ] 0.9266149414114279
6 [0.7500, 0.5781, ] 0.9200359389038635
7 [0.7500, 0.5156, ] 0.7798933739527799
8 [0.9219, 0.7969, ] 0.7771361403442459
9 [0.9844, 0.9375, ] 0.9095603828204
10 [0.7969, 0.5938, ] 0.8254253515527755
11 [0.7500, 0.5781, ] 0.9200359389038635
12 [0.7500, 0.5469, ] 0.9200359389038635
14 [0.9375, 0.9219, ] 0.8413701021927797
15 [0.7344, 0.6719, ] 0.35360758557227123
16 [0.9375, 0.9219, ] 0.8413701021927797
17 [0.9531, 0.9219, ] 0.9801733757561382
19 [0.9531, 0.9062, ] 0.9973271002925899
20 [0.8906, 0.7656, ] 0.9190567751421268
21 [0.9531, 0.9219, ] 0.9801733757561382
22 [0.8906, 0.7969, ] 0.9868706773126782
23 [0.5469, 0.2969, ] 0.8293214471223307
24 [0.8750, 0.7969, ] 0.8982456140350877
25 [0.8906, 0.7969, ] 0.9868706773126782
26 [0.9531, 0.7500, ] 0.2846771771491518
27 [0.9531, 0.7969, ] 0.44503103266984945
29 [0.9531, 0.7812, ] 0.38166311708082545
30 [0.9531, 0.8281, ] 0.6070304195138754
32 [0.9531, 0.7656, ] 0.328729236951222
33 [0.9531, 0.9219, ] 0.9801733757561382
34 [-1.1094, 1.2500, ] 0.18226017435828668
35 [0.9219, 0.7969, ] 0.7771361403442459
36 [0.9219, 0.7656, ] 0.5829041895321386
37 [-1.1094, 1.2500, ] 0.18226017435828668
38 [-1.1250, 1.2656, ] 0.1813031161473088

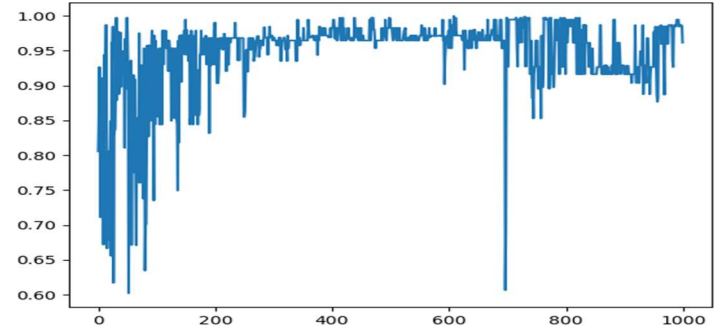
For generation 38 to 42, the max fitness level was 0.1813.

43 [-1.1094, 1.2500, ] 0.18226017435828668
44 [-1.1250, 1.2656, ] 0.1813031161473088
45 [-1.1094, 1.2500, ] 0.18226017435828668

For generation 45 to 998, the max fitness level was 0.1823.

999 [-1.1094, 1.2500, ] 0.18226017435828668
=====
Highest fitness acheived by:
('Gen: 19', [0.953125, 0.90625], 0.9973271002925899)
=====
Rosenbrock benchmark test complete
```

Method 2



```
Running Simple Genetic Algorithm on Rosenbrock benchmark function
Tested population size: 100 Number of generations: 1000

Generation Strongest Candidate Fitness
=====
0 [0.5781, 0.3594, ] 0.8056634380357375
1 [0.7188, 0.5469, ] 0.8541535654355582
2 [0.7188, 0.5156, ] 0.9266149414114279
3 [0.6094, 0.3906, ] 0.8404864689282009
4 [1.1250, 1.2031, ] 0.7111111111111111
5 [0.6250, 0.3438, ] 0.7351040918880115
6 [1.2969, 1.6719, ] 0.9106188394120442

For generation 6 to 8, the max fitness level was 0.9106.

9 [0.3750, 0.1719, ] 0.6719160104986877
10 [1.0938, 1.2188, ] 0.9440745049032495
11 [0.8750, 0.7969, ] 0.8982456140350877
12 [0.8906, 0.7656, ] 0.9190567751421268
13 [0.8906, 0.7969, ] 0.9868706773126782
15 [0.4219, 0.2188, ] 0.6664623251239198
16 [0.5312, 0.2969, ] 0.8056821639430923

For generation 16 to 18, the max fitness level was 0.8057.

19 [0.3125, 0.0938, ] 0.6783422349190577
20 [0.5000, 0.2344, ] 0.7846743295019157

...

969 [0.8750, 0.7656, ] 0.9846153846153847
971 [0.9375, 0.8594, ] 0.9596438821531073
972 [0.8750, 0.7656, ] 0.9846153846153847
974 [0.9375, 0.8594, ] 0.9596438821531073
975 [0.9062, 0.8281, ] 0.9867167533019916

For generation 975 to 979, the max fitness level was 0.9867.

980 [0.8750, 0.7656, ] 0.9846153846153847
981 [0.9062, 0.8281, ] 0.9867167533019916
982 [0.9062, 0.7969, ] 0.9359845468859904
983 [0.7188, 0.5156, ] 0.9266149414114279
984 [0.8750, 0.7656, ] 0.9846153846153847
986 [0.8906, 0.7969, ] 0.9868706773126782
987 [0.8750, 0.7656, ] 0.9846153846153847
989 [0.9375, 0.8750, ] 0.9945972196928307

For generation 989 to 992, the max fitness level was 0.9946.

993 [0.8750, 0.7656, ] 0.9846153846153847

For generation 993 to 995, the max fitness level was 0.9846.

996 [0.8906, 0.7969, ] 0.9868706773126782
998 [0.8750, 0.7656, ] 0.9846153846153847
999 [0.8750, 0.7500, ] 0.9615023474178404
=====
Highest fitness acheived by:
('Gen: 608', [0.984375, 0.96875], 0.9997499614454374)
=====
Rosenbrock benchmark test complete
```