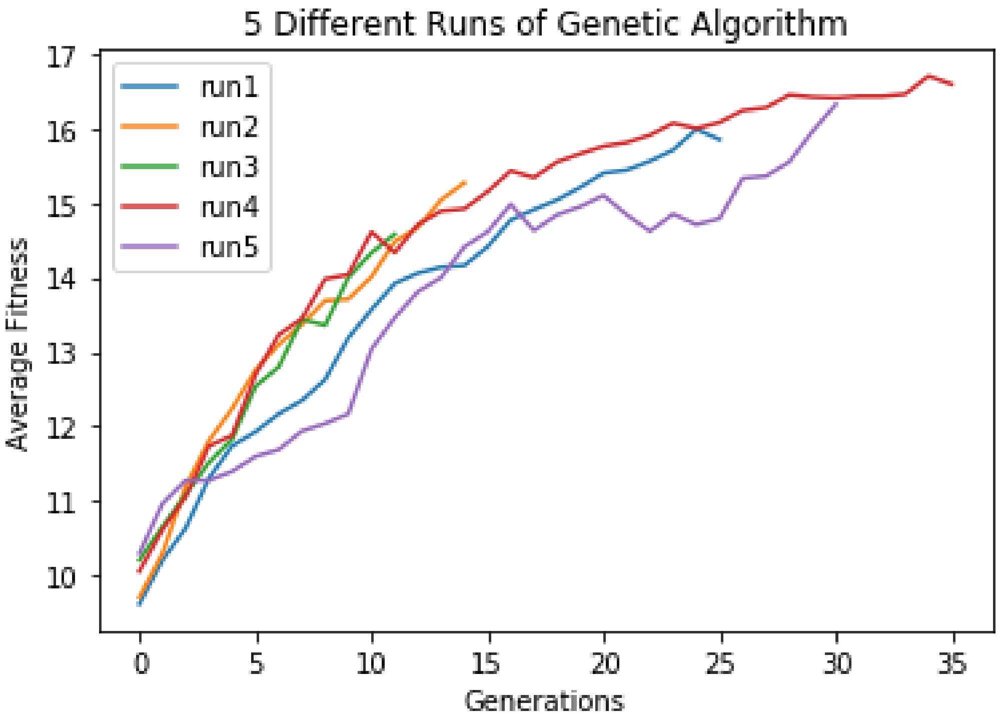
Code for Report

October 17, 2019

[2]:

```
import matplotlib.pyplot as plt
import numpy as np
%matplotlib inline




run1 = np.array([9.62,10.21,10.64,11.31,11.74,11.93,12.17,12.35,12.63,13.19,13.
 ↪57,13.92,14.06,14.14,14.16,14.41,14.77,14.91,15.04,15.21,15.4,15.44,15.56,15.
 ↪71,15.99,15.85])
generations1 = np.arange(0, len(run1), 1)


run2 = np.array([9.71,10.3,11.2,11.81,12.24,12.76,13.1,13.37,13.69,13.71,14.
 ↪01,14.48,14.67,15.04,15.27])
generations2 = np.arange(0, len(run2), 1)

run3 = np.array([10.21,10.66,11.09,11.52,11.83,12.54,12.8,13.44,13.36,14.0,14.
 ↪33,14.58])
generations3 = np.arange(0, len(run3), 1)

run4 = np.array([10.06,10.62,11.06,11.74,11.88,12.7,13.23,13.45,13.98,14.04,14.
 ↪61,14.33,14.71,14.89,14.92,15.15,15.43,15.34,15.55,15.66,15.76,15.81,15.
 ↪91,16.07,16.0,16.08,16.24,16.28,16.45,16.42,16.41,16.43,16.43,16.46,16.7,16.
 ↪59])
generations4 = np.arange(0, len(run4), 1)

run5 = np.array([10.3,10.97,11.28,11.28,11.4,11.6,11.69,11.94,12.04,12.17,13.
 ↪04,13.46,13.81,14.0,14.41,14.61,14.98,14.63,14.84,14.95,15.1,14.84,14.62,14.
 ↪85,14.71,14.79,15.33,15.36,15.55,15.96,16.32])
generations5 = np.arange(0, len(run5), 1)


plt.plot(generations1, run1, label="run1")
plt.plot(generations2, run2,label="run2")
plt.plot(generations3, run3,label="run3")
plt.plot(generations4, run4,label="run4")
plt.plot(generations5, run5,label="run5")
plt.xlabel("Generations")
plt.ylabel("Average Fitness")
plt.legend()
plt.title("5 Different Runs of Genetic Algorithm")
print("Inital Conditions \nPopulation = 100 \nCrossover rate = 80% \nMutation␣
 ↪Rate = 1%")
```

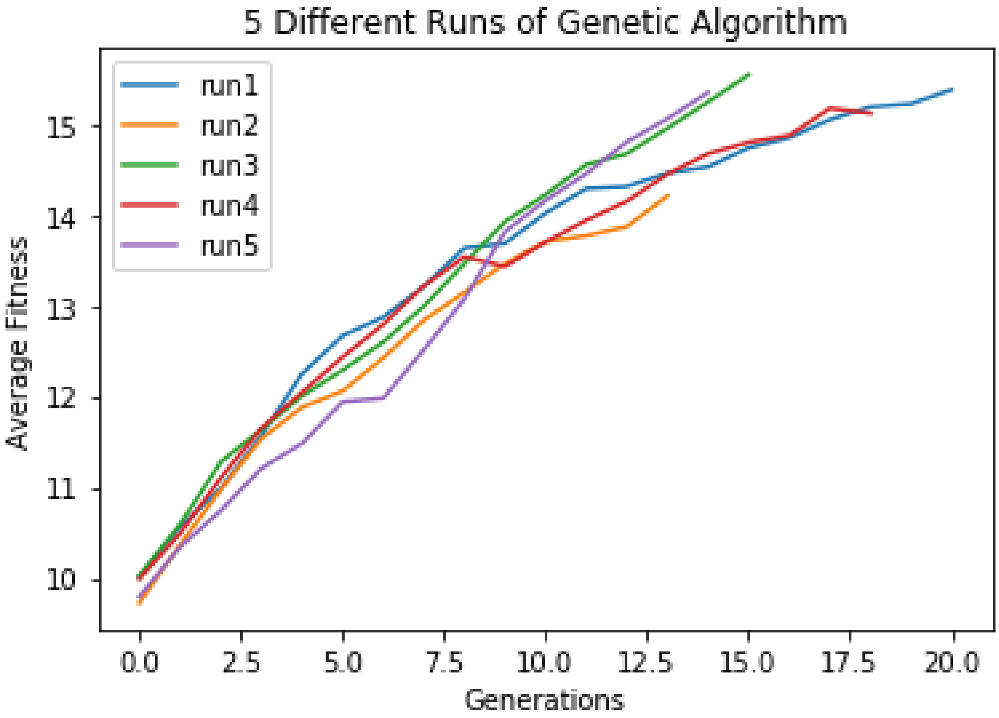
```
Inital Conditions
Population = 100
Crossover rate = 80%
Mutation Rate = 1%
```



[3]:
```
import matplotlib.pyplot as plt
import numpy as np
%matplotlib inline



run1 = np.array([10.015,10.565,10.985,11.57,12.25,12.67,12.875,13.215,13.635,13.
 ↪685,14.02,14.29,14.315,14.46,14.53,14.74,14.85,15.05,15.19,15.225,15.38])
generations1 = np.arange(0, len(run1), 1)
```

```python
run2 = np.array([9.725,10.365,10.97,11.535,11.88,12.06,12.425,12.84,13.15,13.465,13.705,13.77,13.87,14.21])
generations2 = np.arange(0, len(run2), 1)

run3 = np.array([10.025,10.59,11.275,11.635,12.005,12.29,12.6,12.995,13.465,13.925,14.225,14.555,14.675,14.955,15.245,15.545])
generations3 = np.arange(0, len(run3), 1)

run4 = np.array([9.99,10.49,11.1,11.65,12.045,12.435,12.795,13.22,13.54,13.435,13.7,13.94,14.15,14.45,14.675,14.8,14.875,15.175,15.12])
generations4 = np.arange(0, len(run4), 1)

run5 = np.array([9.795,10.34,10.74,11.21,11.48,11.94,11.98,12.515,13.075,13.815,14.16,14.455,14.805,15.06,15.35])
generations5 = np.arange(0, len(run5), 1)


plt.plot(generations1, run1, label="run1")
plt.plot(generations2, run2,label="run2")
plt.plot(generations3, run3,label="run3")
plt.plot(generations4, run4,label="run4")
plt.plot(generations5, run5,label="run5")
plt.xlabel("Generations")
plt.ylabel("Average Fitness")
plt.legend()
plt.title("5 Different Runs of Genetic Algorithm")
print("Inital Conditions \nPopulation = 200 \nCrossover rate = 80% \nMutation Rate = 1%")
```

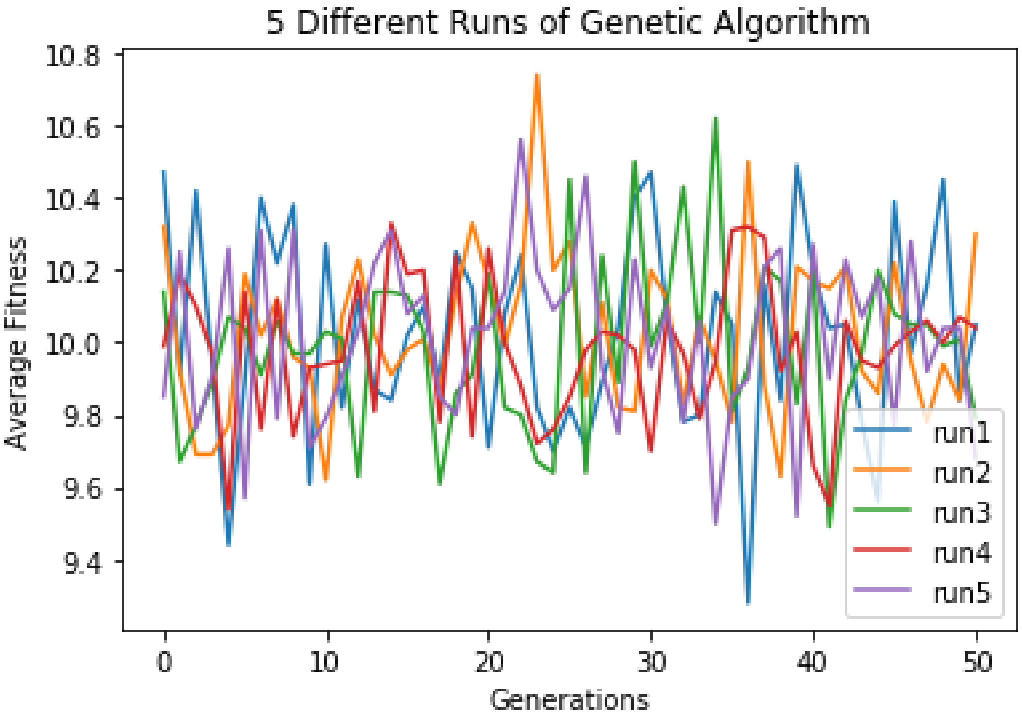
```
Inital Conditions
Population = 200
Crossover rate = 80%
Mutation Rate = 1%
```



[4]:
```
import matplotlib.pyplot as plt
import numpy as np
%matplotlib inline



run1 = np.array([10.47,9.91,10.42,9.94,9.44,9.89,10.4,10.22,10.38,9.61,10.27,9.
 ↪82,10.12,9.87,9.84,10.02,10.1,9.89,10.25,10.15,9.71,10.08,10.24,9.82,9.7,9.
 ↪82,9.71,9.89,10.04,10.4,10.47,10.07,9.78,9.8,10.14,10.05,9.28,10.16,9.84,10.
 ↪49,10.2,10.04,10.05,9.78,9.56,10.39,9.97,10.16,10.45,9.84,10.05])
generations1 = np.arange(0, len(run1), 1)


run2 = np.array([10.32,9.92,9.69,9.69,9.77,10.19,10.02,10.12,9.96,9.93,9.62,10.
 ↪07,10.23,10.02,9.91,9.98,10.01,9.87,10.13,10.33,10.17,9.99,10.16,10.74,10.
 ↪2,10.28,9.85,10.11,9.82,9.81,10.2,10.12,9.83,10.07,9.95,9.78,10.5,9.88,9.
 ↪63,10.21,10.17,10.15,10.21,9.92,9.86,10.22,9.95,9.78,9.94,9.84,10.3])
generations2 = np.arange(0, len(run2), 1)

run3 = np.array([10.14,9.67,9.77,9.89,10.07,10.04,9.91,10.06,9.97,9.97,10.03,10.
 ↪01,9.63,10.14,10.14,10.13,10.03,9.61,9.86,9.91,10.19,9.82,9.8,9.67,9.64,10.
 ↪45,9.64,10.24,9.89,10.5,9.99,10.11,10.43,10.0,10.62,9.81,9.93,10.21,10.17,9.
 ↪83,10.2,9.49,9.84,9.97,10.2,10.08,10.05,10.05,9.99,10.01,9.79])
generations3 = np.arange(0, len(run3), 1)
```

```
run4 = np.array([9.99,10.19,10.1,9.97,9.54,10.14,9.76,10.12,9.74,9.93,9.94,9.
 ↪95,10.17,9.81,10.33,10.19,10.2,9.78,10.24,9.74,10.26,10.0,9.88,9.72,9.76,9.
 ↪85,9.98,10.03,10.02,9.98,9.7,10.06,9.97,9.79,9.95,10.31,10.32,10.29,9.92,10.
 ↪03,9.66,9.55,10.06,9.95,9.93,9.99,10.03,10.06,10.0,10.07,10.04])
generations4 = np.arange(0, len(run4), 1)

run5 = np.array([9.85,10.25,9.76,9.91,10.26,9.57,10.31,9.79,10.31,9.71,9.79,9.
 ↪91,10.03,10.22,10.31,10.08,10.13,9.85,9.8,10.04,10.04,10.14,10.56,10.2,10.
 ↪09,10.15,10.46,9.93,9.75,10.23,9.93,10.1,9.78,10.06,9.5,9.85,9.9,10.21,10.
 ↪26,9.52,10.27,9.9,10.23,10.07,10.18,9.77,10.28,9.92,10.04,10.04,9.68])
generations5 = np.arange(0, len(run5), 1)


plt.plot(generations1, run1, label="run1")
plt.plot(generations2, run2,label="run2")
plt.plot(generations3, run3,label="run3")
plt.plot(generations4, run4,label="run4")
plt.plot(generations5, run5,label="run5")
plt.xlabel("Generations")
plt.ylabel("Average Fitness")
plt.legend()
plt.title("5 Different Runs of Genetic Algorithm")
print("Inital Conditions \nPopulation = 100 \nCrossover rate = 80% \nMutation␣
 ↪Rate = 50%")
```

```
Inital Conditions
Population = 100
Crossover rate = 80%
Mutation Rate = 50%
```

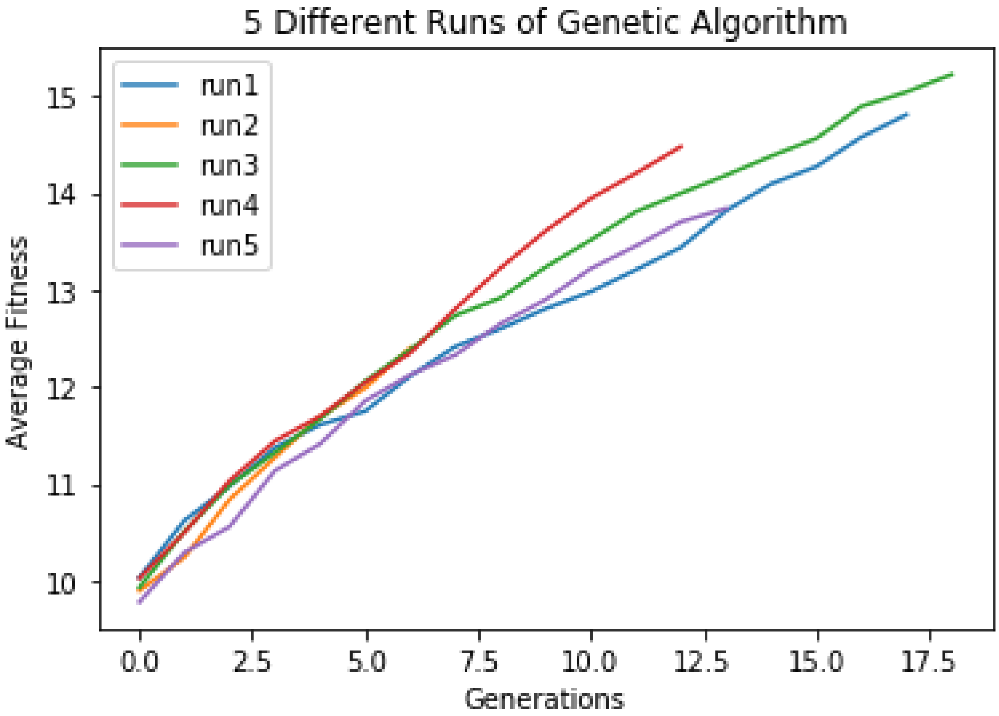


[5]:

```
import matplotlib.pyplot as plt
import numpy as np
%matplotlib inline



run1 = np.array([10.032,10.622,10.976,11.372,11.608,11.748,12.11,12.416,12.
 ↪598,12.806,12.98,13.206,13.436,13.814,14.092,14.27,14.574,14.806])
generations1 = np.arange(0, len(run1), 1)


run2 = np.array([9.898,10.242,10.838,11.272,11.69,11.986,12.402])
generations2 = np.arange(0, len(run2), 1)

run3 = np.array([9.932,10.51,10.984,11.322,11.664,12.056,12.384,12.736,12.
 ↪916,13.234,13.512,13.804,13.994,14.18,14.378,14.56,14.892,15.04,15.218])
generations3 = np.arange(0, len(run3), 1)

run4 = np.array([10.026,10.498,11.032,11.44,11.696,12.044,12.348,12.804,13.
 ↪224,13.606,13.94,14.2,14.476])
generations4 = np.arange(0, len(run4), 1)

run5 = np.array([9.78,10.292,10.558,11.136,11.41,11.854,12.124,12.328,12.652,12.
 ↪898,13.218,13.45,13.7,13.838])
```

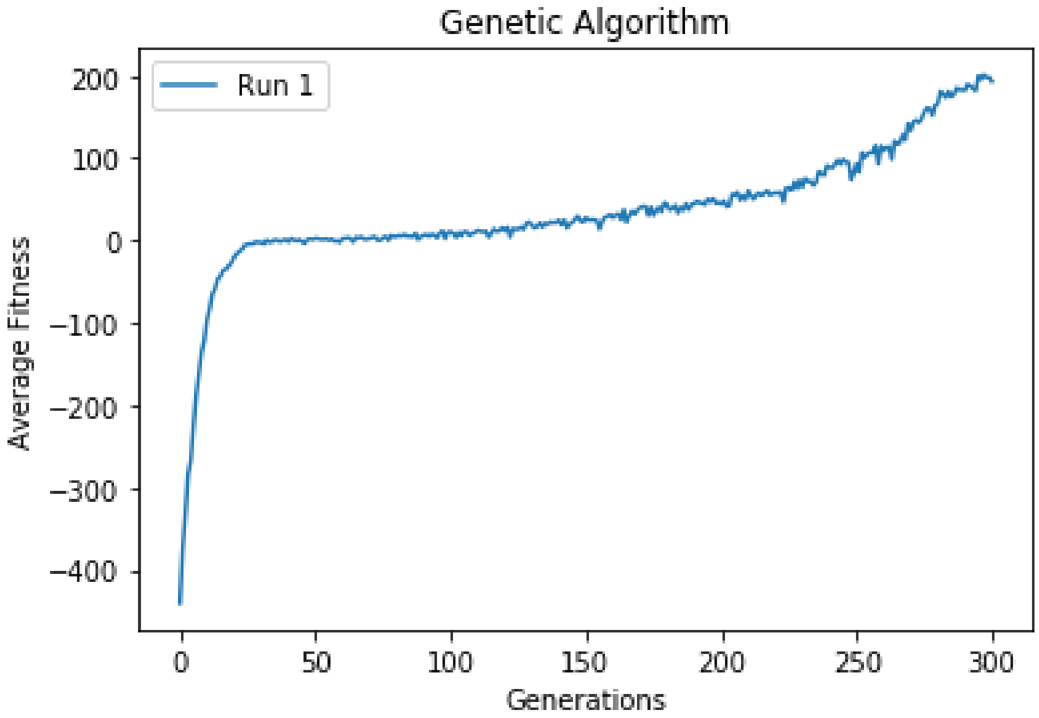
```
generations5 = np.arange(0, len(run5), 1)


plt.plot(generations1, run1, label="run1")
plt.plot(generations2, run2,label="run2")
plt.plot(generations3, run3,label="run3")
plt.plot(generations4, run4,label="run4")
plt.plot(generations5, run5,label="run5")
plt.xlabel("Generations")
plt.ylabel("Average Fitness")
plt.legend()
plt.title("5 Different Runs of Genetic Algorithm")
print("Inital Conditions \nPopulation = 500 \nCrossover rate = 80% \nMutation␣
 ↪Rate = 1%")
```

```
Inital Conditions
Population = 500
Crossover rate = 80%
Mutation Rate = 1%
```



[9]:
```
run1 = np.array([-440.0213333333336,
-374.7351999999999,
-332.1616,
-283.9146666666666,
-267.2343999999999,
-225.07573333333335,
-188.14480000000003,
-162.61386666666655,
-135.21173333333329,
-121.57813333333331,
-96.0962666666666,
-82.2642666666666,
-64.9445333333333,
-58.1087999999999,
-46.3885333333332,
-43.1178666666664,
-35.9386666666666,
-34.949066666666674,
-30.775466666666663,
-27.335199999999997,
-20.0509333333334,
-16.149066666666673,
-13.178666666666661,
-10.759999999999994,
-5.525600000000005,
-4.636000000000001,
-3.8850666666666682,
-4.28,
-1.5552,
-2.479466666666666,
-2.146399999999999,
-4.261600000000003,
0.5479999999999999,
-2.1933333333333316,
-1.4720000000000006,
1.0677333333333334,
0.09813333333333354,
-1.571466666666667,
0.5367999999999998,
1.2432,
-1.6298666666666661,
2.543733333333333,
1.0336,
-0.5685333333333336,
1.1589333333333334,
-1.1874666666666667,
-3.551466666666666,
1.2813333333333334,
2.1330666666666676,
1.2288000000000006,
2.772000000000007,
2.480000000000001,
1.3248000000000006,
1.440533333333333,
2.1346666666666674,
0.4938666666666671,
1.9658666666666673,
-0.7247999999999996,
-0.36293333333333466,
-2.1618666666666684,
2.333866666666666,
2.4949333333333317,
3.537866666666668,
2.3218666666666663,
-1.7112,
3.8895999999999997,
2.7255999999999987,
4.216,
1.631466666666666,
1.460799999999999,
2.695200000000001,
3.591466666666666,
4.473866666666665,
0.5229333333333335,
0.14400000000000057,
3.31866666666667,
4.696,
0.10666666666666685,
6.2376,
4.482933333333333,
5.128000000000002,
6.154666666666667,
5.364,
7.260799999999998,
5.070399999999996,
5.467466666666668,
4.547199999999999,
8.763733333333338,
1.1906666666666672,
4.661066666666668,
6.330933333333335,
7.2336,
4.960799999999998,
7.610133333333332,
6.47786666666667,
3.0178666666666643,
8.878399999999996,
11.818933333333337,
2.1880000000000006,
10.093333333333337,
10.643199999999998,
8.465600000000002,
3.9261333333333317,
9.969866666666668,
5.4064000000000005,
10.849333333333336,
10.397066666666673,
5.961333333333334,
8.580533333333333,
10.000800000000003,
12.410666666666662,
11.141866666666667,
10.224266666666665,
11.677599999999996,
4.2344,
8.652000000000008,
14.218666666666667,
11.244533333333333,
11.785066666666667,
15.591199999999995,
11.790399999999996,
15.8632,
4.381866666666667,
14.040799999999999,
13.722133333333328,
13.844266666666663,
15.870666666666663,
13.535733333333335,
20.830933333333334,
22.237600000000004,
18.473333333333333,
15.502133333333331,
17.146400000000003,
21.608800000000013,
16.336000000000006,
20.845599999999997,
19.776533333333337,
22.613333333333323,
21.521866666666654,
21.999199999999995,
24.689333333333337,
18.942933333333333,
24.926133333333333,
15.345333333333341,
20.340799999999994,
22.061866666666663,
25.85013333333333,
29.558399999999985,
23.779999999999994,
22.340533333333337,
27.80106666666666,
24.91466666666665,
25.104533333333343,
25.264266666666664,
24.216800000000003,
14.080266666666663,
23.242666666666672,
28.252266666666667,
29.3256,
31.602933333333326,
27.570933333333343,
30.789600000000014,
32.48133333333333,
25.166933333333347,
37.38880000000001,
22.316266666666664,
29.75679999999999,
35.78346666666667,
31.824533333333335,
36.6928,
39.80160000000001,
41.60213333333333,
41.2904,
30.527199999999993,
41.08240000000001,
30.398400000000002,
36.29493333333334,
40.65813333333333,
34.93199999999999,
45.698133333333345,
43.573066666666676,
37.05093333333331,
38.933600000000006,
42.815200000000026,
33.31173333333332,
40.515999999999984,
36.885066666666695,
44.67360000000001,
37.68719999999999,
43.83013333333332,
44.60346666666668,
47.468533333333305,
46.338666666666676,
45.51573333333332,
43.96293333333334,
50.569066666666664,
47.06533333333333,
51.22293333333335,
45.34479999999999,
46.06213333333333,
44.6776,
48.20026666666668,
41.34586666666668,
42.29280000000003,
56.10079999999999,
55.234933333333316,
58.575466666666664,
48.96453333333336,
54.44746666666666,
49.76933333333331,
60.76080000000002,
54.79546666666664,
50.24320000000001,
55.446400000000004,
57.62826666666669,
54.58773333333332,
57.69199999999999,
60.70053333333335,
55.419733333333326,
56.761066666666665,
58.09733333333331,
57.57493333333332,
58.63573333333336,
46.77786666666664,
63.6832,
64.08000000000003,
61.07520000000004,
71.51759999999997,
63.121066666666664,
73.36800000000004,
64.7328,
75.32400000000004,
71.47093333333333,
72.03920000000001,
66.93253333333334,
70.84720000000004,
83.8624,
80.9186666666667,
79.31093333333332,
91.08453333333335,
89.30666666666669,
87.97093333333333,
92.76533333333333,
97.83599999999997,
92.7336,
98.83173333333336,
95.47653333333334,
94.73440000000004,
74.08666666666664,
82.29813333333333,
93.57146666666668,
83.43066666666667,
106.56240000000007,
100.97013333333334,
104.85840000000002,
107.18346666666663,
106.49253333333333,
115.22693333333332,
93.36186666666667,
114.73653333333336,
108.99999999999999,
112.9589333333333,
113.37840000000001,
98.67226666666672,
120.37173333333327,
116.75066666666667,
118.3850666666666,
128.00373333333337,
122.15493333333332,
141.8738666666667,
133.49439999999996,
144.2026666666666,
145.95386666666667,
143.27893333333338,
147.81093333333337,
155.80106666666657,
160.7616,
160.27040000000002,
151.8584,
163.67813333333334,
166.65973333333335,
181.16453333333325,
177.55546666666675,
173.57653333333334,
181.11493333333334,
174.47039999999998,
177.69839999999996,
185.39999999999995,
182.46480000000005,
183.35386666666676,
182.15973333333326,
189.54613333333336,
187.57066666666674,
184.89253333333343,
181.90906666666666,
200.13200000000003,
195.92159999999998,
201.35946666666683,
197.9040000000001,
198.14800000000005,
193.25599999999997,
])
generations1 = np.arange(0, len(run1), 1)

plt.plot(generations1, run1,label="Run 1")
plt.xlabel("Generations")
plt.ylabel("Average Fitness")
plt.legend()
plt.title("Genetic Algorithm")
print("Inital Conditions of population = 150 \nCrossover rate = 80% \nMutation Rate = 0.05%")
```

```
Inital Conditions of population = 150
Crossover rate = 80%
Mutation Rate = 0.05%
```

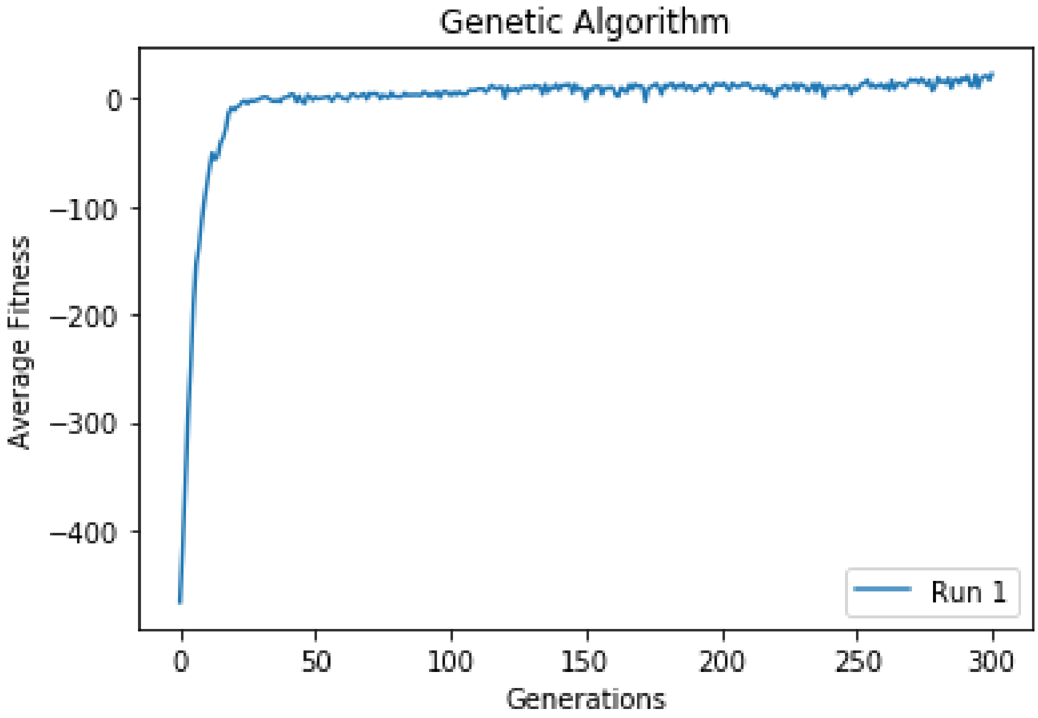

Genetic Algorithm
Run 1
200
100
0
−100
−200
−300
−400
Average Fitness
0
50
100
150
200
250
300
Generations

[10]:

```
run1 = np.array([-466.62, -416.85360000000014, -356.6548, -293.89599999999996,␣
 ↪-247.80039999999997, -190.90039999999993, -153.39639999999997, -138.
 ↪05880000000005, -114.56639999999999, -95.21119999999998, -79.28960000000002,␣
 ↪-63.186800000000005, -50.107599999999984, -56.5312, -53.151199999999998, -41.
 ↪0492, -36.21240000000001, -27.290799999999994, -13.236799999999995, -8.
 ↪212800000000003, -10.7228, -7.690399999999998, -6.4556, -3.2463999999999995,␣
 ↪-2.0380000000000003, -4.808000000000001, -1.7059999999999997, -2.
 ↪3779999999999997, -1.6752, -0.7931999999999999, 0.7384, 1.4791999999999996,␣
 ↪1.4024, -0.4412000000000004, -1.8368000000000004, -2.4388, -1.
 ↪3512000000000006, -2.834000000000002, 0.7527999999999999, 1.
 ↪1327999999999998, 1.7187999999999999, 4.2764000000000015, 3.6639999999999993,␣
 ↪-2.7447999999999984, 2.5052000000000008, -2.4324000000000003, -4.39, 3.
 ↪3191999999999986, 1.7796, -1.3419999999999996, 1.494, -0.11719999999999964,␣
 ↪0.5104000000000003, 0.9708000000000001, 1.4484000000000001, 2.
 ↪0143999999999993, -2.289999999999999, 0.14599999999999977, 3.
 ↪649999999999999, 2.8196, 0.08480000000000022, 0.2063999999999993, -1.
 ↪4020000000000001, 3.6572000000000005, 4.3792, 2.088, 1.4736000000000001, 3.
 ↪5280000000000014, 4.460000000000002, -0.5548000000000002, 4.812, 4.
 ↪145600000000001, 4.8344, 4.159200000000001, -0.38960000000000006, 3.
 ↪349599999999997, 2.0312000000000006, 0.5495999999999993, 5.6108, 2.0196, 2.
 ↪2484000000000037, 1.5603999999999996, 2.1475999999999997, 5.1588, 2.7228, 3.
 ↪765199999999999, 3.0324, 3.7064, 3.3212, 2.9520000000000017, 6.
 ↪119200000000002, 5.170000000000001, 2.2976000000000014, 4.716000000000001, 2.
 ↪853999999999999, 6.4784000000000015, 1.2703999999999989, 5.976, 6.
 ↪5723999999999965, 5.403199999999999, 3.1964000000000006, 5.879599999999999,␣
 ↪3.7016000000000013, 6.3064000000000004, 4.6412, 4.018, 4.8532, 8.
 ↪000000000000002, 8.021599999999998, 7.4144000000000005, 9.057200000000002, 9.
 ↪1656, 9.106399999999997, 5.978399999999998, 8.629200000000003, 12.
 ↪134000000000007, 10.467200000000005, 9.047599999999997, 10.723600000000001,␣
```

```
generations1 = np.arange(0, len(run1), 1)

plt.plot(generations1, run1,label="Run 1")
plt.xlabel("Generations")
plt.ylabel("Average Fitness")
plt.legend()
plt.title("Genetic Algorithm")
print("Inital Conditions of population = 100 \nCrossover rate = 100% \nMutation␣
 ↪Rate = 0.05%")
```

```
Inital Conditions of population = 100
Crossover rate = 100%
Mutation Rate = 0.05%
```



[ ]:

```

```