

```

1  #rosenbrock_test.py
2  from simple_genetic_algorithm import SGA
3
4  ALNUM = ["0", "1"]
5  VAR_STRING_LEN = 16
6  NUMBER_OF_VARIABLES = 2
7  VARIABLE_LEN = int(VAR_STRING_LEN / NUMBER_OF_VARIABLES)
8  PROBABILITY_OF_MUTATION = 0.01
9  POP_SIZE = 100
10 NUM_GENERATIONS = 1000
11 DOMAIN_MIN = -2
12 DOMAIN_MAX = 2
13
14
15 def rosenbrock(*xs):
16     """
17     The function is defined on n-dimensional space.
18     The function can be defined on any input domain but it is usually evaluated on
19     x_i element of [-5, 10] for i = 1, ..., n.
20
21     Takes in n dimensional coordinates and returns output of:
22         f(x,y) = sum[b(x_i + 1 - (x_i)^2)^2 + (a - x_i)^2]]
23     for i = 1, ..., n; and the parameters a and b are constants set to a = 1 and b =
24     100..
25
26     The function has one global minimum f(x^*) = 0 at x^* = (1, ..., 1).
27
28     Args:
29         xs (List[num]): n dimensional coordinates for Euclidean (n + 1)-space
30     Returns:
31         (float): f(x,y) = value closer to 0 indicates a coordinate closer to know
32         global minimum
33     """
34
35     x = xs[0]
36     y = xs[1]
37
38     return (1-x)**2+100*(y-x**2)**2
39
40
41 print("Running Simple Genetic Algorithm on Rosenbrock benchmark function")
42
43 SGA(rosenbrock, POP_SIZE, ALNUM, VAR_STRING_LEN, VARIABLE_LEN, DOMAIN_MIN, DOMAIN_MAX,
44     NUM_GENERATIONS,
45     PROBABILITY_OF_MUTATION)
46
47 print("\nRosenbrock benchmark test complete\n")
48

```