

```

1
2
3 import random
4 import math
5 import copy
6 import sys
7
8
9 def fitness_rastrigin(position):
10     fitness_value = 0.0
11     for i in range(len(position)):
12         xi = position[i]
13         fitness_value += (xi * xi) - (10 * math.cos(2 * math.pi * xi)) + 10
14     return fitness_value
15
16
17 def fitness_sphere(position):
18     fitness_value = 0.0
19     for i in range(len(position)):
20         xi = position[i]
21         fitness_value += (xi*xi);
22     return fitness_value;
23
24
25
26
27 class wolf:
28     def __init__(self, fitness, dim, minx, maxx, seed):
29         self.rnd = random.Random(seed)
30         self.position = [0.0 for i in range(dim)]
31
32         for i in range(dim):
33             self.position[i] = ((maxx - minx) * self.rnd.random() + minx)
34
35         self.fitness = fitness(self.position) # curr fitness
36
37
38
39
40 def gwo(fitness, max_iter, n, dim, minx, maxx):
41     rnd = random.Random(0)
42
43
44     population = [ wolf(fitness, dim, minx, maxx, i) for i in range(n)]
45
46
47     population = sorted(population, key = lambda temp: temp.fitness)
48
49
50     alpha_wolf, beta_wolf, gamma_wolf = copy.copy(population[: 3])
51
52
53
54     Iter = 0
55     while Iter < max_iter:
56
57         if Iter % 10 == 0 and Iter > 1:
58             print("Iter = " + str(Iter) + " best fitness = %.3f" % alpha_wolf.fitness)
59
60
61         a = 2*(1 - Iter/max_iter)
62
63
64         for i in range(n):
65             A1, A2, A3 = a * (2 * rnd.random() - 1), a * (
66                 2 * rnd.random() - 1), a * (2 * rnd.random() - 1)

```

```

67 C1, C2, C3 = 2 * rnd.random(), 2*rnd.random(), 2*rnd.random()
68
69 X1 = [0.0 for i in range(dim)]
70 X2 = [0.0 for i in range(dim)]
71 X3 = [0.0 for i in range(dim)]
72 Xnew = [0.0 for i in range(dim)]
73 for j in range(dim):
74     X1[j] = alpha_wolf.position[j] - A1 * abs(
75     C1 * alpha_wolf.position[j] - population[i].position[j])
76     X2[j] = beta_wolf.position[j] - A2 * abs(
77     C2 * beta_wolf.position[j] - population[i].position[j])
78     X3[j] = gamma_wolf.position[j] - A3 * abs(
79     C3 * gamma_wolf.position[j] - population[i].position[j])
80     Xnew[j] += X1[j] + X2[j] + X3[j]
81
82 for j in range(dim):
83     Xnew[j] /= 3.0
84
85
86 fnew = fitness(Xnew)
87
88
89 if fnew < population[i].fitness:
90     population[i].position = Xnew
91     population[i].fitness = fnew
92
93
94 population = sorted(population, key = lambda temp: temp.fitness)
95
96
97 alpha_wolf, beta_wolf, gamma_wolf = copy.copy(population[: 3])
98
99 Iter += 1
100
101 return alpha_wolf.position
102
103
104
105
106
107
108 print("\nBegin grey wolf optimization on rastrigin function\n")
109 dim = 3
110 fitness = fitness_rastrigin
111
112
113 print("Goal is to minimize Rastrigin's function in " + str(dim) + " variables")
114 print("Function has known min = 0.0 at (", end="")
115 for i in range(dim-1):
116     print("0, ", end="")
117 print("0)")
118
119 num_particles = 50
120 max_iter = 100
121
122 print("Setting num_particles = " + str(num_particles))
123 print("Setting max_iter = " + str(max_iter))
124 print("\nStarting GWO algorithm\n")
125
126
127
128 best_position = gwo(fitness, max_iter, num_particles, dim, -10.0, 10.0)
129
130 print("\nGWO completed\n")
131 print("\nBest solution found:")
132 print(["%.6f"%best_position[k] for k in range(dim)])
133 ann = fitness(best_position)

```

```

133 err = fitness(best_position)
134 print("fitness of best solution = %.6f" % err)
135
136 print("\nEnd GWO for rastrigin\n")
137
138
139 print()
140 print()
141
142
143
144 print("\nBegin grey wolf optimization on sphere function\n")
145 dim = 3
146 fitness = fitness_sphere
147
148
149 print("Goal is to minimize sphere function in " + str(dim) + " variables")
150 print("Function has known min = 0.0 at (", end="")
151 for i in range(dim-1):
152     print("0, ", end="")
153 print("0)")
154
155 num_particles = 50
156 max_iter = 100
157
158 print("Setting num_particles = " + str(num_particles))
159 print("Setting max_iter = " + str(max_iter))
160 print("\nStarting GWO algorithm\n")
161
162
163
164 best_position = gwo(fitness, max_iter, num_particles, dim, -10.0, 10.0)
165
166 print("\nGWO completed\n")
167 print("\nBest solution found:")
168 print(["%.6f"%best_position[k] for k in range(dim)])
169 err = fitness(best_position)
170 print("fitness of best solution = %.6f" % err)
171
172 print("\nEnd GWO for sphere\n")
173

```



Begin grey wolf optimization on rastrigin function

Goal is to minimize Rastrigin's function in 3 variables
 Function has known min = 0.0 at (0, 0, 0)
 Setting num_particles = 50
 Setting max_iter = 100

Starting GWO algorithm

Iter = 10 best fitness = 6.636
 Iter = 20 best fitness = 1.047
 Iter = 30 best fitness = 1.012
 Iter = 40 best fitness = 1.010
 Iter = 50 best fitness = 1.008
 Iter = 60 best fitness = 1.008
 Iter = 70 best fitness = 1.008
 Iter = 80 best fitness = 1.006
 Iter = 90 best fitness = 1.005

GWO completed

Best solution found:
 ['-0.004395', '0.995042', '0.005800']
 fitness of best solution = 1.005465

End GWO for rastrigin

Begin grey wolf optimization on sphere function

Goal is to minimize sphere function in 3 variables
Function has known min = 0.0 at (0, 0, 0)
Setting num_particles = 50
Setting max_iter = 100

Starting GWO algorithm

Iter = 10 best fitness = 0.000
Iter = 20 best fitness = 0.000
Iter = 30 best fitness = 0.000
Iter = 40 best fitness = 0.000
Iter = 50 best fitness = 0.000
Iter = 60 best fitness = 0.000
Iter = 70 best fitness = 0.000
Iter = 80 best fitness = 0.000
Iter = 90 best fitness = 0.000

GWO completed

Best solution found:
['0.000000', '0.000000', '-0.000000']
fitness of best solution = 0.000000