```
import random
import math
import copy
import sys


def fitness_rastrigin(position):
  fitness_value = 0.0
  for i in range(len(position)):
    xi = position[i]
    fitness_value += (xi * xi) - (10 * math.cos(2 * math.pi * xi)) + 10
  return fitness_value


def fitness_sphere(position):
  fitness_value = 0.0
  for i in range(len(position)):
    xi = position[i]
    fitness_value += (xi*xi);
  return fitness_value;




class wolf:
  def __init__(self, fitness, dim, minx, maxx, seed):
    self.rnd = random.Random(seed)
    self.position = [0.0 for i in range(dim)]

    for i in range(dim):
      self.position[i] = ((maxx - minx) * self.rnd.random() + minx)

    self.fitness = fitness(self.position) # curr fitness




def gwo(fitness, max_iter, n, dim, minx, maxx):
  rnd = random.Random(0)


  population = [ wolf(fitness, dim, minx, maxx, i) for i in range(n)]


  population = sorted(population, key = lambda temp: temp.fitness)


  alpha_wolf, beta_wolf, gamma_wolf = copy.copy(population[: 3])



  Iter = 0
  while Iter < max_iter:

    if Iter % 10 == 0 and Iter > 1:
      print("Iter = " + str(Iter) + " best fitness = %.3f" % alpha_wolf.fitness)


    a = 2*(1 - Iter/max_iter)


    for i in range(n):
      A1, A2, A3 = a * (2 * rnd.random() - 1), a * (
      2 * rnd.random() - 1), a * (2 * rnd.random() - 1)
```

```
        C1, C2, C3 = 2 * rnd.random(), 2*rnd.random(), 2*rnd.random()

        X1 = [0.0 for i in range(dim)]
        X2 = [0.0 for i in range(dim)]
        X3 = [0.0 for i in range(dim)]
        Xnew = [0.0 for i in range(dim)]
        for j in range(dim):
          X1[j] = alpha_wolf.position[j] - A1 * abs(
          C1 * alpha_wolf.position[j] - population[i].position[j])
          X2[j] = beta_wolf.position[j] - A2 * abs(
          C2 * beta_wolf.position[j] - population[i].position[j])
          X3[j] = gamma_wolf.position[j] - A3 * abs(
          C3 * gamma_wolf.position[j] - population[i].position[j])
          Xnew[j]+= X1[j] + X2[j] + X3[j]

        for j in range(dim):
          Xnew[j]/=3.0


        fnew = fitness(Xnew)


        if fnew < population[i].fitness:
          population[i].position = Xnew
          population[i].fitness = fnew


      population = sorted(population, key = lambda temp: temp.fitness)


      alpha_wolf, beta_wolf, gamma_wolf = copy.copy(population[: 3])

      Iter+= 1

    return alpha_wolf.position




print("\nBegin grey wolf optimization on rastrigin function\n")
dim = 3
fitness = fitness_rastrigin


print("Goal is to minimize Rastrigin's function in " + str(dim) + " variables")
print("Function has known min = 0.0 at (", end="")
for i in range(dim-1):
  print("0, ", end="")
print("0)")

num_particles = 50
max_iter = 100

print("Setting num_particles = " + str(num_particles))
print("Setting max_iter = " + str(max_iter))
print("\nStarting GWO algorithm\n")



best_position = gwo(fitness, max_iter, num_particles, dim, -10.0, 10.0)

print("\nGWO completed\n")
print("\nBest solution found:")
print(["%.6f"%best_position[k] for k in range(dim)])
err = fitness(best_position)
```

```
err = fitness(best_position)
print("fitness of best solution = %.6f" % err)

print("\nEnd GWO for rastrigin\n")


print()
print()


print("\nBegin grey wolf optimization on sphere function\n")
dim = 3
fitness = fitness_sphere


print("Goal is to minimize sphere function in " + str(dim) + " variables")
print("Function has known min = 0.0 at (", end="")
for i in range(dim-1):
  print("0, ", end="")
print("0)")

num_particles = 50
max_iter = 100

print("Setting num_particles = " + str(num_particles))
print("Setting max_iter = " + str(max_iter))
print("\nStarting GWO algorithm\n")



best_position = gwo(fitness, max_iter, num_particles, dim, -10.0, 10.0)

print("\nGWO completed\n")
print("\nBest solution found:")
print(["%.6f"%best_position[k] for k in range(dim)])
err = fitness(best_position)
print("fitness of best solution = %.6f" % err)

print("\nEnd GWO for sphere\n")
```

```
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
```