Grey wolf optimization

```
import numpy as np

# Objective function
def objective_function(x):
    return x ** 2  # Example: f(x) = x^2

# Grey Wolf Optimizer (GWO)
def grey_wolf_optimizer(num_wolves, num_iterations, search_space):
    # Initialize positions of wolves randomly within the search space
    wolves = np.random.uniform(search_space[0], search_space[1], num_wolves)
    fitness = np.array([objective_function(x) for x in wolves])

    # Identify initial alpha, beta, and delta
    sorted_indices = np.argsort(fitness)
    alpha = wolves[sorted_indices[0]]  # Best wolf (alpha)
    beta = wolves[sorted_indices[1]]   # Second-best wolf (beta)
    delta = wolves[sorted_indices[2]]  # Third-best wolf (delta)

    # Main loop
    for iteration in range(num_iterations):
        # Coefficients for position update
        a = 2 - 2 * (iteration / num_iterations)  # Decreases linearly from 2 to 0

        for i in range(num_wolves):
            # Calculate distances and update positions based on alpha, beta, and delta
            r1, r2 = np.random.rand(), np.random.rand()
            A1 = 2 * a * r1 - a
            C1 = 2 * r2
            D_alpha = abs(C1 * alpha - wolves[i])
            X1 = alpha - A1 * D_alpha

            r1, r2 = np.random.rand(), np.random.rand()
            A2 = 2 * a * r1 - a
            C2 = 2 * r2
            D_beta = abs(C2 * beta - wolves[i])
            X2 = beta - A2 * D_beta

            r1, r2 = np.random.rand(), np.random.rand()
            A3 = 2 * a * r1 - a
            C3 = 2 * r2
            D_delta = abs(C3 * delta - wolves[i])
            X3 = delta - A3 * D_delta

            # Update wolf position
            wolves[i] = (X1 + X2 + X3) / 3

        # Re-evaluate fitness and reassign alpha, beta, delta
        fitness = np.array([objective_function(x) for x in wolves])
        sorted_indices = np.argsort(fitness)
        alpha = wolves[sorted_indices[0]]
        beta = wolves[sorted_indices[1]]
        delta = wolves[sorted_indices[2]]

        # Print progress
        print(f"Iteration {iteration + 1}: Best fitness =
{fitness[sorted_indices[0]]}")

    # Return the best solution
    return alpha, objective_function(alpha)

# Parameters
num_wolves = 5
num_iterations = 10
search_space = [-10, 10]

# Run GWO
best_position, best_fitness = grey_wolf_optimizer(num_wolves,
num_iterations, search_space)
print(f"\nBest Position: {best_position}")
print(f"Best Fitness: {best_fitness}")
```

Output:

```
Iteration 1: Best fitness = 0.01378208183853719
Iteration 2: Best fitness = 0.024041598710359807
Iteration 3: Best fitness = 0.21586441846415377
Iteration 4: Best fitness = 0.005337081602530873
Iteration 5: Best fitness = 0.008144580288255723
Iteration 6: Best fitness = 0.012572112428281266
Iteration 7: Best fitness = 0.013078741511827101
Iteration 8: Best fitness = 0.0106563547678389
Iteration 9: Best fitness = 0.010225590482327171
Iteration 10: Best fitness = 0.009993460292752011

Best Position: 0.09996729611603992
Best Fitness: 0.009993460292752011
```