An Improved Grey Wolf Optimizer Based on Tracking and Seeking Modes to Solve Function Optimization Problems

Varsha Ramdas

Rajagiri School Of Engineering and Technology

Abstract

- Grey wolf optimizer (GWO) is a new meta-heuristic algorithm.
- It mimics the leadership hierarchy and hunting mechanism of grey wolves in nature.
- An improved grey wolf optimizer based on tracking and seeking mode
- Proposed to improve the diversity of the population and the ability of the algorithm to balance exploration and exploitation.

Metaheuristic Algorithm

- Metaheuristic algorithm finds the best solution out of all possible solutions of an optimization.
- Meta-heuristics have become remarkably common due to four factors:
  - Simplicity
  - Flexibility
  - Derivation-free mechanism
  - Local optima avoidance

Class division

- Meta-heuristics can be divided into two main classes:
  1. Single-solution-based: The search process starts with one candidate solution.
  2. Population-based: Perform the optimization using a set of solutions (population)

GWO Inspiration

- Grey wolves are considered as apex predators.

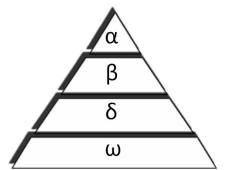

Figure 1: Hierarchy of grey wolves

- The leaders are a male and a female, called Alphas.
- The second level in the hierarchy is Beta.
- The Delta wolf provides food to the pack.
- The lowest ranking grey wolf is Omega which plays the role of scapegoat.

Mathematical Modelling

Phases of Grey Wolf Hunting

- Tracking, chasing, and approaching the prey
- Pursuing, encircling, and harassing the prey
- Attacking the prey

Figure 2: Hunting behaviour of Grey Wolves

Encircling the prey

- The formula for the grey wolf surrounding the prey can be described as follows.

$$D = |C.X_p(t) - X(t)| \tag{1}$$

$$X(t+1) = X_p(t) - A.D \tag{2}$$

- t - Algebraic number of the current iteration
- Xp(t)- Characterizes the position vector of the prey
- X(t)- Position vector of the grey wolf
- $\vec{A}$ and $\vec{C}$- Coefficient vectors

$$A = 2a.r1 - a \tag{3}$$

$$C = 2.r2 \tag{4}$$

Position vectors

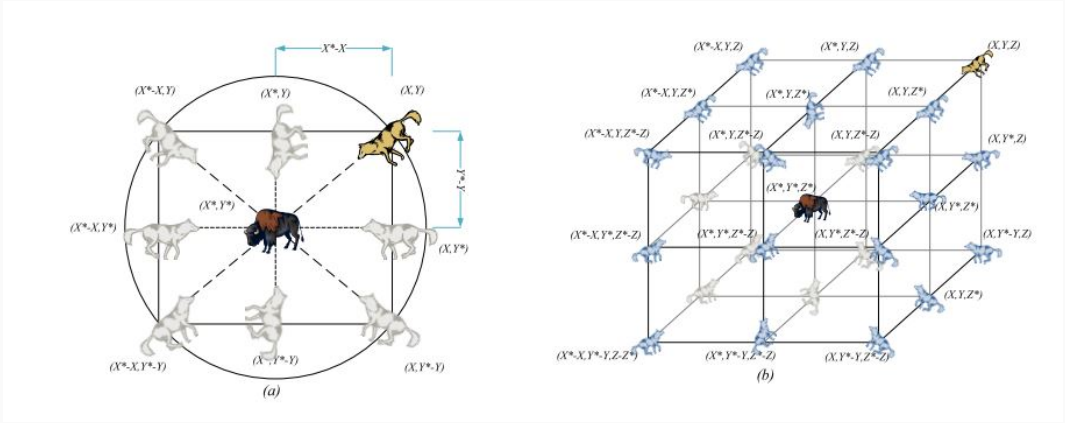

Figure 3: 2D and 3D position vectors and their possible next locations

Hunting

- In order to mathematically simulate the hunting we make the following assumptions:
  1. The alpha, beta, and delta have better knowledge about the potential location of prey.
  2. Thus, we update their positions according to the position of the best search agent.
- Alpha, beta, and delta estimate the position of the prey, and other wolves updates their positions randomly around the prey.

$$D_\alpha = |C1.X_\alpha - X|, D_\beta = |C1.X_\beta - X|, D_\delta = |C1.X_\delta - X| \tag{5}$$

$$X_1 = X_\alpha - A_1.(D_\alpha), X_2 = X_\beta - A_2.(D_\beta), X_3 = X_\delta - A_3.(D_\delta) \tag{6}$$

$$X(t+1) = \frac{X_1 + X_2 + X_3}{3} \tag{7}$$

Position Updation in GWO

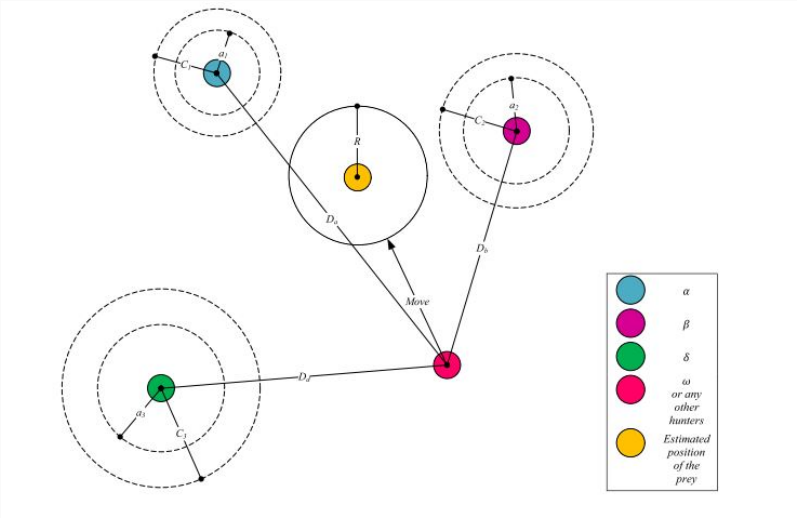

Figure 4: Position updation in GWO

Exploration of prey

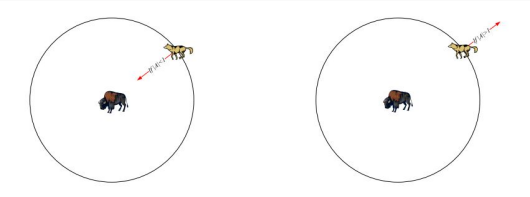

Figure 5: Exploitation vs Exploration of prey

- They diverge from each other to search for prey and converge to attack prey.
- A with random values greater than 1 or less than -1 to oblige the search agent to diverge from the prey.
- C vector contains random values in [0, 2].

Flowchart based on Tracking and Seeking mode

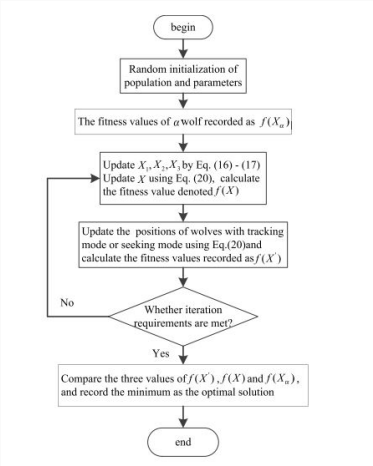

Figure 6: Flowchart of GWO based on tracking mode and seeking mode

Algorithm

Tracking mode Algorithm

1. Calculate the speed of the i^{th} Wolf in each dimension.
2. Update the location of the Grey Wolf

Seeking mode Algorithm

1. The memory pool has N copies.
2. Change the dimension of each wolf in the memory pool.
3. Calculate the fitness values of all the found candidate solutions.
4. The position of the optimal candidate is taken as the position of the current individual.

Tracking and Seeking mode Algorithm

1. Initialize the algorithm control parameters: population size , the maximum iteration number, memory pool (SMP), number of changes
2. update the positions of α, β and δ wolf. Calculate the fitness value of all wolves,
3.
4. Update the position X
5. If f
6. Meet the conditions for stopping the algorithm and output the optimal solution, otherwise return to stop.

Pseudo Code of GWO

Initialize the grey wolf population X_i *(i = 1, 2, ..., n)*
Initialize a, A, and C
Calculate the fitness of each search agent
X_α*=the best search agent*
X_β*=the second best search agent*
X_δ*=the third best search agent*
while *(t < Max number of iterations)*
    for *each search agent*
        Update the position of the current search agent by equation (3.7)
    end for
    Update a, A, and C
    Calculate the fitness of all search agents
    Update X_α, X_β, *and* X_δ
    t=t+1
end while
return X_α

Figure 7: Pseudo code of GWO

Simulation and Performance Anlaysis