Metaheuristic Optimization: Grey Wolf Optimizer (GWO)

Adaptive and Cooperative Algorithms (ECE 457A)

ECE, MME, and MSCI Departments,
University of Waterloo, ON, Canada

Course Instructor: Benyamin Ghojogh
Fall 2023

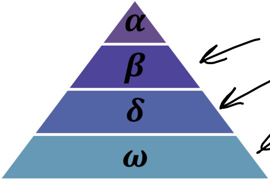
Introduction

- Grey Wolf Optimizer (GWO) was proposed by Seyedali Mirjalili *et al.* in 2014 [1].
- This method is highly cited and recognized.
- It is a **nature-inspired swarm** metaheuristic optimization algorithm.
- It is inspired by **grey wolves**, also called **Canis lupus**.
- It uses two main aspects of grey wolves' lives:
  - Hierarchy of grey wolves: α, β, δ, and ω wolves
  - Hunting strategy:
    - Searching for the prey
    - Encircling the prey
    - Attacking the prey

Credit of image: `https://www.nwf.org/Educational-Resources/Wildlife-Guide/Mammals/Gray-Wolf`

Hierarchy of grey wolves

- Grey wolves live in groups (packs) of 5 to 12.
- They are members of a social hierarchy:
  - α wolves: leaders, the most dominant, they only are allowed to mate, not necessarily the strongest but the best managers
  - β wolves: submit to α wolves, help α wolves in decision-making
  - δ wolves: submit to α and β wolves, involving these roles:
    - ★ scouts: watching the boundaries of the territory
    - ★ sentinels: protecting and and guaranteeing the safety of the group
    - ★ elders: experiences wolves which used to be α or β wolves
    - ★ hunters: helping α and β wolves in hunting
    - ★ caretakers: taking care of the weak, sick, and wounded wolves in the group
  - ω wolves: submit to all wolves, last wolves allowed to eat, sometimes they babysit



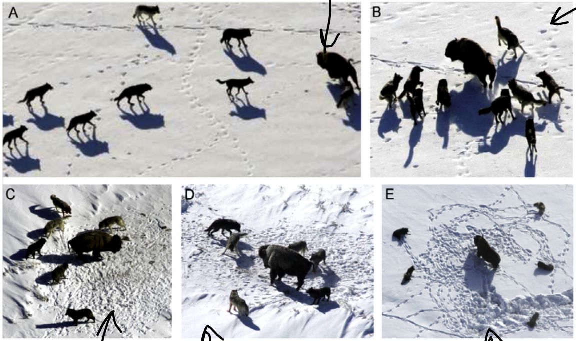
Credit of image: [1]

Hierarchy of grey wolves in GWO

- In the GWO algorithm, we consider the three best candidate solutions found so far as α, β, and δ wolves, respectively:
  - α wolf: the best candidate solution found so far
  - β wolf: the second best candidate solution found so far
  - δ wolf: the third best candidate solution found so far
- The rest of the candidate solutions are ω wolves.

Hunting strategy of grey wolves

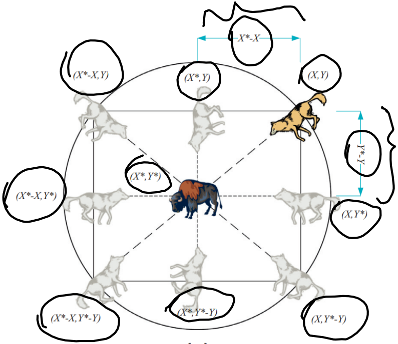
- The hunting strategy of grey wolves involves three steps [2]:
  - Searching for the prey (chasing, approaching, and tracking prey) - Fig. A
  - Encircling the prey (pursuing, harassing, and encircling) - Figs. B to D
  - Attacking the prey (stationary situation and attack) - Fig. E



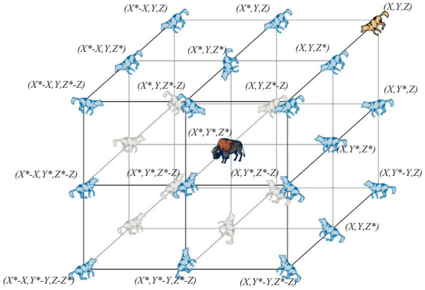
Credit of image: [2]

Hunting strategy of grey wolves in GWO

- The grey wolves encircle around the prey. We model it mathematically as the following figures in 2D and 3D:



(a)



(b)

Credit of image: [1]

Hunting strategy of grey wolves in GWO

- The grey wolves encircle around the prey. We model it mathematically as:

$$\mathbb{R}^d \ni \boldsymbol{x}^{(t+1)} := \boldsymbol{x}^{(t)} - (\boldsymbol{a} \odot \boldsymbol{d}), \tag{1}$$
$$\mathbb{R}^d \ni \boldsymbol{d} := |(\boldsymbol{c} \odot \boldsymbol{x}_p) - \boldsymbol{x}^{(t)}|, \tag{2}$$
$$[-\boldsymbol{b}, \boldsymbol{b}] = [-2, 2]^d \ni \boldsymbol{a} := 2(\boldsymbol{b} \odot \boldsymbol{r}_1) - \boldsymbol{b}, \tag{3}$$
$$[-2, 2]^d \ni \boldsymbol{c} := 2\boldsymbol{r}_2, \tag{4}$$

where d is the dimensionality of the optimization variable, t is the iteration index, $\boldsymbol{x}_p$ is the position of the prey, $\odot$ denotes the Hadamard (element-wise) product, $|.|$ is the absolute value, $\boldsymbol{r}_1, \boldsymbol{r}_2 \in [0,1]^d$ are uniform random vectors with elements between zero and one, and $\boldsymbol{b}$ is a vector whose elements are initially 2 but gradually decrease to 0 by the iterations of the algorithm.

- As $\boldsymbol{r}_1 \in [0,1]^d$ and $\boldsymbol{b} \in [0,2]^d$, the range of $\boldsymbol{a}$ is $\boldsymbol{a} \in [-\boldsymbol{b}, \boldsymbol{b}] = [-2, 2]^d$.
- As $\boldsymbol{r}_2 \in [0,1]^d$, the range of $\boldsymbol{c}$ is $\boldsymbol{c} \in [-2, 2]^d$.

$[0,2]^d$

Hunting strategy of grey wolves in GWO

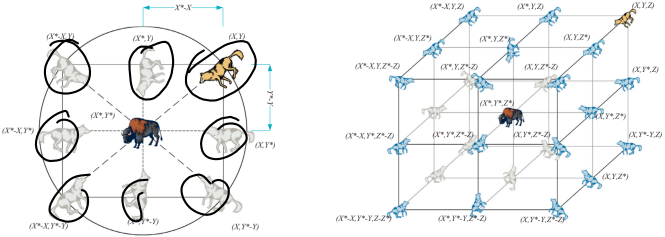
- We had:

$$\begin{cases} \mathbb{R}^d \ni \boldsymbol{x}^{(t+1)} := \boldsymbol{x}^{(t)} - (\boldsymbol{a} \odot \boldsymbol{d}), & \mathbb{R}^d \ni \boldsymbol{d} := |(\boldsymbol{c} \odot \boldsymbol{x}_p) - \boldsymbol{x}^{(t)}|, \\ [-\boldsymbol{b}, \boldsymbol{b}] = [-2, 2]^d \ni \boldsymbol{a} := 2(\boldsymbol{b} \odot \boldsymbol{r}_1) - \boldsymbol{b}, & [-2, 2]^d \ni \boldsymbol{c} := 2\boldsymbol{r}_2. \end{cases}$$

- These formulas model all the possibility of a hyper-cube around the prey.
- For example, in 2D, if we have $\boldsymbol{a} = [1, 0]^\top$ and $\boldsymbol{c} = [1, 1]^\top$, we have:

$$\boldsymbol{c} = [1, 1]^\top \implies \boldsymbol{d} = |\boldsymbol{x}_p - \boldsymbol{x}^{(t)}| = \begin{bmatrix} |x_{p,1} - x_1^{(t)}| \\ |x_{p,2} - x_2^{(t)}| \end{bmatrix}.$$

$$\boldsymbol{a} = [1, 0]^\top \implies \boldsymbol{x}^{(t+1)} = \begin{bmatrix} x_1^{(t)} \\ x_2^{(t)} \end{bmatrix} - \begin{bmatrix} |x_{p,1} - x_1^{(t)}| \\ 0 \end{bmatrix} = \begin{bmatrix} 2x_1^{(t)} - x_{p,1} \\ x_2^{(t)} \end{bmatrix}.$$



(a) (b)

Hunting strategy of grey wolves in GWO

- In reality, hunting is mainly performed by the α, β, and δ wolves.
- At every iteration, we consider the positions of the α, β, and δ candidate solutions as the candidate preys. So, we use the above-mentioned formulas three times:

$$\boldsymbol{x}_1^{(t+1)} := \boldsymbol{x}_\alpha^{(t)} - (\boldsymbol{a}_1 \odot \boldsymbol{d}_\alpha), \quad \boldsymbol{x}_2^{(t+1)} := \boldsymbol{x}_\beta^{(t)} - (\boldsymbol{a}_2 \odot \boldsymbol{d}_\beta), \quad \boldsymbol{x}_3^{(t+1)} := \boldsymbol{x}_\delta^{(t)} - (\boldsymbol{a}_3 \odot \boldsymbol{d}_\delta),$$
$$\boldsymbol{d}_\alpha := |(\boldsymbol{c}_1 \odot \boldsymbol{x}_\alpha) - \boldsymbol{x}^{(t)}|, \quad \boldsymbol{d}_\beta := |(\boldsymbol{c}_2 \odot \boldsymbol{x}_\beta) - \boldsymbol{x}^{(t)}|, \quad \boldsymbol{d}_\delta := |(\boldsymbol{c}_3 \odot \boldsymbol{x}_\delta) - \boldsymbol{x}^{(t)}|,$$

  where $\boldsymbol{x}_\alpha$, $\boldsymbol{x}_\beta$, and $\boldsymbol{x}_\delta$ are the positions of the α, β, and δ candidate solutions (three best solutions found so far), and $\boldsymbol{x}^{(t)}$ is the current position of a candidate solution (one of the wolves).
- For every candidate solution (wolf), the update candidate solution is the average of these three updates, because we assume that the α, β, and δ wolves are encircling around the prey:

$$\boldsymbol{x}^{(t+1)} := \frac{\boldsymbol{x}_1^{(t+1)} + \boldsymbol{x}_2^{(t+1)} + \boldsymbol{x}_3^{(t+1)}}{3}. \tag{5}$$

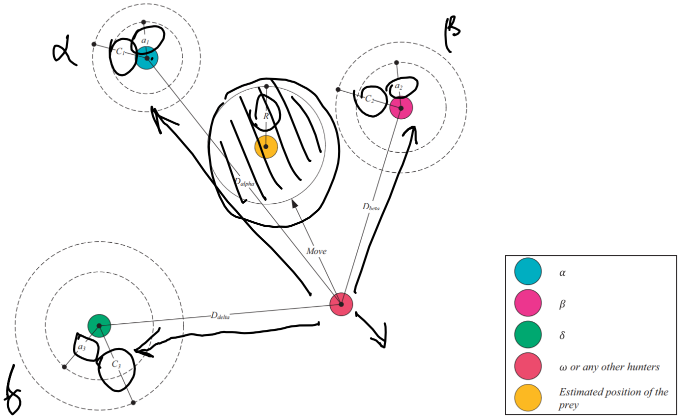
Hunting strategy of grey wolves in GWO

- We had:

$$\boldsymbol{x}_1^{(t+1)} := \boldsymbol{x}_\alpha^{(t)} - (\boldsymbol{a}_1 \odot \boldsymbol{d}_\alpha), \quad \boldsymbol{x}_2^{(t+1)} := \boldsymbol{x}_\beta^{(t)} - (\boldsymbol{a}_2 \odot \boldsymbol{d}_\beta), \quad \boldsymbol{x}_3^{(t+1)} := \boldsymbol{x}_\delta^{(t)} - (\boldsymbol{a}_3 \odot \boldsymbol{d}_\delta),$$

$$\boldsymbol{d}_\alpha := |(\boldsymbol{c}_1 \odot \boldsymbol{x}_\alpha) - \boldsymbol{x}^{(t)}|, \quad \boldsymbol{d}_\beta := |(\boldsymbol{c}_2 \odot \boldsymbol{x}_\beta) - \boldsymbol{x}^{(t)}|, \quad \boldsymbol{d}_\delta := |(\boldsymbol{c}_3 \odot \boldsymbol{x}_\delta) - \boldsymbol{x}^{(t)}|,$$

$$\boldsymbol{x}^{(t+1)} := \frac{\boldsymbol{x}_1^{(t+1)} + \boldsymbol{x}_2^{(t+1)} + \boldsymbol{x}_3^{(t+1)}}{3}.$$

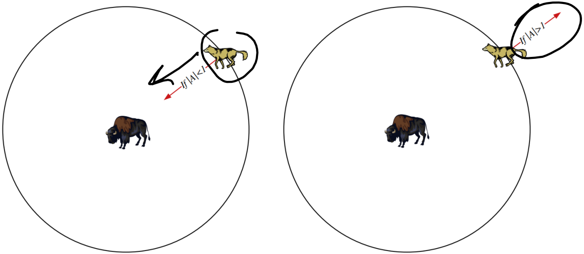


Exploration and Exploitation in GWO

- We had:

$$\mathbb{R}^d \ni \boldsymbol{x}^{(t+1)} := \boldsymbol{x}^{(t)} - (\boldsymbol{a} \odot \boldsymbol{d}).$$

- Recall the range of $\boldsymbol{a} \in [-\boldsymbol{b}, \boldsymbol{b}] = [-2, 2]^d$.
- If $\boldsymbol{a} \in [-1, 1]^d$, the candidate solution is moved toward inside the hypercube, i.e., toward the prey.
- If $\boldsymbol{a} \in [-2, -1]^d$ or $\boldsymbol{a} \in [1, 2]^d$, the candidate solution is moved toward outside the hypercube, i.e., away from the prey.
- Initially, $\boldsymbol{a} \approx \{-2, 2\}^d$, so it moves away from the prey to **explore** more.
- Later, $\boldsymbol{a}$ is decreases and when it becomes $|\boldsymbol{a}| \in [1, 2]^d$, it moves toward the prey to **exploit** more.



(a) **(b)**

GWO Algorithm

Algorithm GWO

Initialize the candidate solutions (wolves) $\{\boldsymbol{x}_i\}_{i=1}^{n}$
Initialize the $\boldsymbol{a}$, $\boldsymbol{b}$, and $\boldsymbol{c}$ values
while *not terminated* **do**
  Calculate the cost values at the candidate solutions
  $\boldsymbol{x}_\alpha, \boldsymbol{x}_\beta, \boldsymbol{x}_\delta \leftarrow$ the three best solutions so far
  for *each candidate solution* $\boldsymbol{x}_i$ **do**
    Calculate $\boldsymbol{x}_1^{(t+1)}, \boldsymbol{x}_2^{(t+1)}, \boldsymbol{x}_3^{(t+1)}$
    $\boldsymbol{x}_i^{(t+1)} := (\boldsymbol{x}_1^{(t+1)} + \boldsymbol{x}_2^{(t+1)} + \boldsymbol{x}_3^{(t+1)})/3$
  Update the $\boldsymbol{a}$, $\boldsymbol{b}$, and $\boldsymbol{c}$ values
Return the best solution $\boldsymbol{x}_\alpha$

Acknowledgment

- A scholar in this area: **Seyedali Mirjalili**, Torrens University Australia, Australia, https://scholar.google.com/citations?user=TJHmrREAAAAJ&hl=en&oi=sra
- Videos about GWO by Seyedali Mirjalili: https://seyedalimirjalili.com/gwo
- Codes of GWO in Python, MATLAB, Java, R, C, C++, and Ruby: https://seyedalimirjalili.com/gwo
- More nature-inspired metaheuristic optimization algorithms by Seyedali Mirjalili: https://seyedalimirjalili.com/projects
  Some examples of his work:
  - Whale Optimization Algorithm [3]
  - Ant Lion Optimizer [4]
  - Moth Flame Optimizer [5]
  - Dragonfly Algorithm [6]
  - Grasshopper Optimization Algorithm [7]
  - Salp Swarm Algorithm [8]
- A recent **survey** on nature-inspired optimization is published in 2023 [9].
- One of my initial papers: Pontogammarus Maeoticus Swarm Optimization (PMSO) [10]

References

[1] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Advances in engineering software*, vol. 69, pp. 46–61, 2014.

[2] C. Muro, R. Escobedo, L. Spector, and R. Coppinger, "Wolf-pack (canis lupus) hunting strategies emerge from simple rules in computational simulations," *Behavioural processes*, vol. 88, no. 3, pp. 192–197, 2011.

[3] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Advances in engineering software*, vol. 95, pp. 51–67, 2016.

[4] S. Mirjalili, "The ant lion optimizer," *Advances in engineering software*, vol. 83, pp. 80–98, 2015.

[5] S. Mirjalili, "Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm," *Knowledge-based systems*, vol. 89, pp. 228–249, 2015.

[6] S. Mirjalili, "Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems," *Neural computing and applications*, vol. 27, pp. 1053–1073, 2016.

[7] Y. Meraihi, A. B. Gabis, S. Mirjalili, and A. Ramdane-Cherif, "Grasshopper optimization algorithm: theory, variants, and applications," *Ieee Access*, vol. 9, pp. 50001–50024, 2021.

[8] S. Mirjalili, A. H. Gandomi, S. Z. Mirjalili, S. Saremi, H. Faris, and S. M. Mirjalili, "Salp swarm algorithm: A bio-inspired optimizer for engineering design problems," *Advances in engineering software*, vol. 114, pp. 163–191, 2017.

References (cont.)

[9] S. Darvishpoor, A. Darvishpour, M. Escarcega, and M. Hassanalian, "Nature-inspired algorithms from oceans to space: A comprehensive review of heuristic and meta-heuristic optimization algorithms and their potential applications in drones," *Drones*, vol. 7, no. 7, p. 427, 2023.

[10] B. Ghojogh and S. Sharifian, "Pontogammarus maeoticus swarm optimization: A metaheuristic optimization algorithm," *arXiv preprint arXiv:1807.01844*, 2018.