Jul 24

Grey Wolf Optimization

Initialization: Start with a random group of solutions

Ranking: Identify the best solution (alpha), the second best (beta) and then delta, gamma the followers.

Keep iterating: keep iterating for the pre-defined num_of_iterations, and the final ranking will be reached.

After all the iterations are done the alpha will be the most optimal solution followed by, beta, ~~gamma~~ delta, gamma.

The algorithm stops when it finds a sufficiently good solution or reaches the max iterations.

Implementation in Image processing + Disaster

```
num_wolves = 20
num_iterations = 50

Fitness = evaluate_Fitness(wolves)

alpha, beta, delta = rank_wolves(wolves)

For i in range(num_iterations)
        wolf = update_position(wolf, alpha, beta, delta)
    Fitness = evaluate_Fitness(wolf)
print("Best solution is:", alpha)
```

Image Enhancement:

GWO can optimize the & alpha (contrast) and beta (brightness or sharpness) for a function to improve the visual quality of the image.

Noise Reduction:

GWO can help in the filtering /denoising of the image, making it more clear

The objective is to maximize the PSNR which is the Peak Signal to Noise Ratio which measures the quality of the denoised image.

We can minimize the MSE as minimizing this will result in better denoising.

Disadvantages:

- Sensitivity to Parameter Settings: This means that if the parameters are not set appropriately, it may struggle to find the solution

- ~~dependency on~~

Search space dependency: The algorithm converges to a local optima too early so it may not be able to find the global optima

Limited Fine-Tuning Capabilities:

Some tasks require precise parameter tuning, GWO may struggle to find highly accurate solutions.

To improve GWO we can use EGWO which is Enhanced Grey WOlF Optimization:

which makes it so that the parameters used will be dynamic rather than static.