1. Experimental content

2.1 Initialization Function

Added the function of recording the best fitness data in every 500 iterations (1 round) in the main function section; Added the function of selecting the best fitness data from the 20 best fitness data; Added the function of recording the total time required for running 20 rounds; Added the function of calculating the median of data.

2.2 Analysis of the advantages of the other three algorithms

2.2.1. Dandelion Optimization Algorithm (DO)

• *Global search capability*

DO utilizes the Levy flight mechanism and random mutation to enhance its ability to escape from local optima.

• *Diversity maintenance*

DO introduces random factors of diversity through the Rising Stage to ensure population diversity.

• *Boundary adjustment*

Boundary constraints ensure that individuals are always within the feasible domain, preventing invalid searches.

具体替换想法：
考虑到蒲公英优化算法在全局搜索能力和多样性维护方面的优势，我们可以通过以下方式进行替换：
替换探索阶段
理由：蒲公英优化算法的 Rising Stage 利用 Levy 飞行机制和随机变异，能够有效增强全局搜索能力和多样性维护。
替换内容：
在白鲸优化算法的探索阶段，可以用蒲公英优化算法的 Rising Stage 来替代现有的游泳模式和 Levy 飞行部分。

2.2.2. Crested Porcupine Optimizer（CPO）

• *Global search capability*

CPO adopts multi strategy simulation to simulate the four defense mechanisms of the

Crown Porcupine - the first defense strategy (visual intimidation), the second defense strategy (sound intimidation), the third defense strategy (odor attack), and the fourth defense strategy (physical attack). The first and second defense strategies are used for global exploration, while the third and fourth defense strategies are used for local development. Especially in the third and fourth stages, the convergence accuracy has been improved.

By combining Levy flight and random walk strategies, CPO enhances its ability to jump out of local optima, ensuring that the algorithm can effectively search in a vast solution space.

• *Diversity maintenance*

CPO adopts the technique of cyclic population reduction (CPR) to dynamically adjust population size and maintain population diversity. This mechanism not only accelerates the convergence speed, but also prevents the occurrence of premature convergence phenomenon.

• *Boundary adjustment*

As the iteration progresses, CPO can adaptively adjust the strictness of the boundaries, ensuring the comprehensiveness of the search and promoting local fine search. The key parameters are adaptively adjusted according to the operation of the algorithm, reducing manual parameter tuning while improving the robustness of the algorithm.

具体替换想法：

2.2.3. Mountain Gazelle Optimizer（MGO）

• *Global search capability*

MGO adopts an efficient global exploration mechanism by mimicking the behavior of gazelles searching for food in complex terrain. This mechanism utilizes the jumping, running, and climbing movements of the mountain gazelle, effectively covering the solution space and increasing the chance of finding the global optimal solution. Especially in the later stage, MGO utilized the precise foraging behavior of mountain gazelles to improve convergence accuracy.

By combining Levy flight and random walk strategies, MGO enhances its ability to jump out of local optima, ensuring that the algorithm can effectively search in a vast solution space.

• *Diversity maintenance*

Group dynamic adjustment: MGO introduces a group dynamic adjustment mechanism to adaptively adjust the interaction patterns between individuals based on environmental changes and search progress. This helps maintain population diversity and prevent premature convergence.

By introducing random perturbation factors, MGO can continuously introduce new solutions during the search process, maintaining the vitality and diversity of the population.

• *Boundary adjustment*

MGO has a built-in intelligent boundary constraint processing mechanism to ensure that all individuals are always within the feasible domain, avoiding invalid searches. This mechanism improves search efficiency and ensures the quality of understanding.

As the iteration progresses, MGO can adaptively adjust the strictness of the boundaries according to the search situation, ensuring the comprehensiveness of the search and promoting local fine search.

• *Easy to use*

Easy to implement: MGO has a relatively simple structure, is easy to program, and requires fewer control parameters, reducing the difficulty of parameter tuning and making it suitable for researchers and engineers to quickly get started.

• *Strong adaptability*

Multi objective optimization: MGO can be used not only for single objective optimization, but also for multi-objective optimization problems. It can effectively handle the trade-off problem in multi-objective optimization by simulating the decision-making process of a gazelle facing multiple choices.

具体替换想法：

2.3 Further optimization attempts

2.　Experimental result

We ran the BWO initial code 5 times and tested the function's performance as follows:

Table1. Initial BWO

Test function	The magnitude of the median	The magnitude of variance	The magnitude of best adaptation data	Average runtime per run (seconds)
F1	e^-8	e^-3	e^-34	14
F2	e^-3	e^-2	e^-24	13
F3	e^-35	e^-28	e^-130	34
F4	e^-5	e^-1	e^-27	27
F5	e^-4	e^-3	e^-14	12
F6	e^0	e^-1	e^-4	16
F7	e^-4	e^-3	e^-6	14
F8	e^-3	e^0	e^-30	18
F9	e^-7	e^-2	e^-28	19

Hint：　Running once will present the best adaptation data after 20 iterations and 500 iterations. The test function is generally stable only in terms of runtime, and has a certain degree of randomness in other aspects, so we can only make relative comparisons. (F7, F8 are relatively stable in all four aspects)

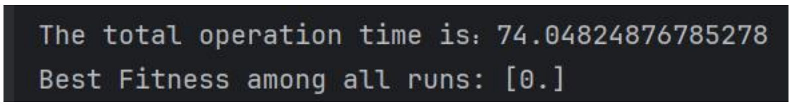
The first stage of DO replacing:

Table2. DO First Stage

Test function	The magnitude of best adaptation data	Average runtime per run (seconds)

F1	e^-30	15
F2	e^-17	16
F3	e^-20	49
F4	e^-14	15
F5	e^-4	18
F6	e^-6	16
F7	e^-5	18
F8	e^-307	74
F9	e^-20	18

After running F8 multiple times, the optimal fitness shown in the figure below is [0.], indicating that the data is already very close.



```
The total operation time is: 74.04824876785278
Best Fitness among all runs: [0.]
```

Figure1. Close to Zero

Summary: Regarding the F8 testing function, there has been an improvement in the level of optimal data adaptation, but the corresponding running time has also increased. The improvement effect of other testing functions is not significant, and even inferior to BWO (such as F2, F3).

The second stage of DO replacing:

Table3. DO Second Stage

Test function	The magnitude of best adaptation data	Average runtime per run (seconds)
F1	e^-2	20
F2	e^-2	21
F3	e^1	54
F4	e^-2	20

F5	e^1	23
F6	e^-1	20
F7	e^-5	22
F8	e^-17	80
F9	e^-4	21

Summary: In terms of magnitude and running time, it is not as good as BWO, and the improvement effect is very poor.

The first stage of CPO replacing:

Table4. CPO First Stage

Test function	The magnitude of best adaptation data	Average runtime per run (seconds)
F1	e^-24	17
F2	e^-20	18
F3	e^-26	52
F4	e^-13	16
F5	e^-2	19
F6	e^-4	17
F7	e^-4	19
F8	[0.]	74
F9	e^-20	19

After running F8 multiple times, the optimal fitness was [0.], indicating that the data is already very close.

Summary: Regarding the F8 testing function, there has been a significant improvement in the level of optimal data adaptation, but the corresponding running time has also increased. The improvement effect of other testing functions is not significant, and even inferior to BWO (such as F1).

The second stage of CPO replacing:

Table5. CPO Second Stage

Test function	The magnitude of best adaptation data	Average runtime per run (seconds)
F1	e^-3	24
F2	e^-2	25
F3	e^-2	57
F4	e^-2	23
F5	e^1	25
F6	e^-1	24
F7	e^-3	26
F8	e^-30	78
F9	e^-4	26

Summary: The improvements are not significant or even inferior to BWO.

The first stage of MGO replacing:

Table6. MGO First Stage

Test function	The magnitude of best adaptation data	Average runtime per run (seconds)
F1	e^-4	32
F2	e^-2	33
F3	e^-2	67
F4	e^-3	31
F5	e^-3	34
F6	e^-3	32
F7	e^-4	34
F8	e^-39	89

F9	e^-6	34

Summary: Regarding the F8 testing function, there has been an improvement in the optimal order of data adaptation, but the effect is weak and the running time has also increased significantly. The improvement effect of the remaining test functions is not significant, and even most of them are inferior to BWO.

BWO Self-improved:

Table7. Self-improved

测试函数	最佳适应数据的量级	每次运行平均时长（秒）
F1	e^	
F2	e^	
F3	e^	
F4	e^	
F5	e^	
F6	e^	
F7	e^	
F8	e^	
F9	e^	

总结：

BWO Self-improved and DO first stage replacing:

Table8. Self and DO First Stage

测试函数	最佳适应数据的量级	每次运行平均时长（秒）
F1	e^23	15
F2	e^14	16

F3	e^31	48
F4	e^	
F5	e^	
F6	e^	
F7	e^	
F8	e^	
F9	e^	

总结：

BWO Self-improved, DO first stage and CPO second stage replacing:

Table9. Self, DO and CPO

测试函数	最佳适应数据的量级	每次运行平均时长（秒）
F1	e^	
F2	e^	
F3	e^	
F4	e^	
F5	e^	
F6	e^	
F7	e^	
F8	e^	
F9	e^	

总结:

综上所述，在已有的几种替换方法中，我们综合评估，决定采用（）作为目前的算法优化的最佳方案。

3. Program code

In this section, the program code that remains indented is listed (not screenshots).

根据第三部分，我们决定采用（）为最终的替换优化算法。具体代码内容在附件的 zip 压缩文件中。

4. Conclusions

Everyone should give the problems and solutions encountered in the experiment process, experience, opinions and suggestions.

金垲峰的困难在于对 levy 飞行阶段的改进还没有找到合适的方法，有困难。以及对 BWO 的三个阶段进行其他算法的替换操作时，改进不是很明显，只有对特定的测试函数有效果。

黄家睿的困难在于认为不同的算法对于初始矩阵的处理是不一样的，其中 MGO 算法与我们原本算法 BWO 算法的区别最大，MGO 算法的全局搜索和局部搜索是同时进行的，这给算法部分的替换带来了一些问题。在第一阶段的位置变换后，矩阵的格式并不与下一阶段的算法相适应，因此在不同阶段替换后，需要注意不同算法对初始化矩阵的处理

另外金和黄都发现，函数进行全局探索和收敛的时候，随机性很强，尤其是某些测试函数，经常发生运行多次之后，没有一个相对稳定的最佳收敛结果数量级。我们只能尽可能去进行测试和替换。
由于时间和精力原因，我们并未完全试过所有的替换可能。如果未来有可能，我们会进一步探索、优化该算法。