**This file provides a simple instruction for the related source files for the Fig. 4 of our manuscript.**

The MATLAB source code of different PSO variants and our EVOLER algorithm, as well as their running results, are structured into the corresponding folders.

User Instruction to Fig. 4

1. **Source Code Running:**

Our code can directly run on the proper software, i.e., MATLAB R2019a.

1. **Running time**

For the source code of our proposed EVOLER and other PSO methods on 30-dimensional benchmark functions (including: (a) Shifted Levy Function, (b) Hybrid Composition Function 1, we evaluate the CPU running time respectively (@ CPU 4GHz, RAM 32GB).

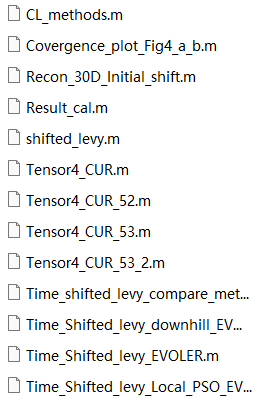
1. **Implementation guidelines**

In the following, we give the implementation guidelines when reproducing Fig. 4

* **Reproducing Fig. 4a-b:**

For convenience, we have alternatively provided the derived numerical results for directly plotting the figures 4. To do so, one may follow the subsequent guidelines.

Firstly, open the MATLAB platform, and then open the file folder ‘Time\_Complexity\_reproducing\_Fig4’ and open the folder ‘Fig4\_ab\_reproducing’, the source code is structured into the corresponding folders, i.e.,



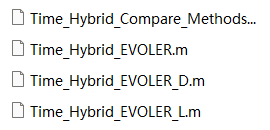
Note that, we can respectively evaluate the performance and CPU running time of the Shifted Levy Function, by using 11 comparative methods and our EVOLER methods, EVOLER Local PSO method, EVOLER Downhill method.

Firstly, we directly run the MATLAB file ‘Time\_CLPSO\_method’ and then run the file ‘Time\_shifted\_levy\_compare\_methods.m’, which outputs the fitness results of 11 PSO variant methods for 100 independent trials (saved in the data matrix ‘ff\_all.mat’) and the corresponding CPU running time (saved in the data matrix ‘time\_all.mat’). Next, we can run the other three MATLAB file ‘Time\_Shifted\_levy\_EVOLER.m’, ‘Time\_Shifted\_levy\_downhill\_EVOLER.m’,‘Time\_Shifted\_levy\_Local\_PSO\_EVOLER.m’, which output the corresponding fitness results (The data is saved in ‘shifted\_Pro\_data\_levy.mat’,‘Local\_data\_levy.mat’,‘Downhill\_Pro\_data\_levy.mat’ respectively) and their corresponding CPU running times.

And finally, Fig.4a and Fig.4b can be obtained by running the source code file ‘Covergence\_plot\_Fig4\_a\_b.m’.

* **Reproducing Fig. 4c:**

Firstly, open the file folder ‘Time\_Complexity\_reproducing\_Fig4’ and open the folder ‘Fig4\_c\_reproducing’, the source code is structured into the corresponding folders, i.e.,



Note that, we can respectively evaluate the CPU running time of the Hybrid Composition Function, by using 11 comparative methods and our EVOLER methods, EVOLER Local PSO method, EVOLER Downhill method.

Firstly, we directly run three MATLAB files ‘Time\_Hybrid\_EVOLER.m’, ‘Time\_Hybrid\_EVOLER\_D.m’,‘Time\_Hybrid\_EVOLER\_L.m’ successively, which output the corresponding running time of different methods (The data is saved in ‘time\_evoler\_hybrid.mat’, ‘time\_evoler\_hybrid\_D.mat’, ‘time\_evoler\_hybrid\_L.mat’ respectively). After then, we run the MATLAB file ‘Time\_Hybrid\_Compare\_Methods.m’, which outputs the CPU running time of 11 PSO variant methods for 10 independent trials and the results is saved in the data matrix ‘time\_end\_all.mat’’.

Finally, we can obtain the Fig.4c by running the source code file ‘Running\_time\_Fig4c.m’.

* **Reproducing Fig. 4d:**

Open the MATLAB platform, and then open the file folder ‘Test\_Function\_reproducing\_Fig3’ and open the folder ‘Test\_Function\_30D’, one can reproduce the Fig.4d according to the ‘Read\_me’.