**PLOJF: Polar Lights Optimization with Jumping Historical Search and Fitness-Diversity Balancing**

# Abstract

# Introduction

# Dataset

# The Proposed PLOJF Algorithm

In this section, we introduce the proposed PLOJF algorithm. By integrating a Jumping Historical Search strategy and a Fitness-Diversity Balanced (FDB) selection mechanism, PLOJF enhances the search efficiency and solution precision of the original PLO algorithm.

## Polar Lights Optimization

Polar Lights Optimization (PLO) is a metaheuristic inspired by the auroral phenomena observed near the Earth's poles. It balances exploration and exploitation through three coordinated mechanisms: (i) a gyration motion that provides local exploitation, (ii) an auroral oval walk guided by a Lévy flight to encourage global exploration, and (iii) a particle-collision–based refinement that adjusts solutions at the dimension level. An adaptive weighting scheme modulates the influence of the local and global components over the course of the search, enabling PLO to shift focus as needed.

Gyration motion is motivated by the trajectories of high-energy charged particles moving in a magnetic field. Let denote the particle charge, its velocity, the magnetic field, and the mass. A simplified first-order model for the velocity dynamics is

|  |  |  |
| --- | --- | --- |
|  |  | (1) |

To account for atmospheric damping, a factor is introduced, yielding

|  |  |  |
| --- | --- | --- |
|  |  | (2) |

Solving Eq. (2) gives

|  |  |  |
| --- | --- | --- |
|  |  | (3) |

where is a constant of integration. In the original setting, parameters are typically normalized with , , , and the time variable is tied to the number of function evaluations.

To strengthen global exploration, PLO employs an auroral oval walk guided by a Lévy flight and the population mean. For individual , the exploratory step is

|  |  |  |
| --- | --- | --- |
|  |  | (4) |

where denotes a Lévy-flight–based perturbation, is the population mean, and are the lower and upper bounds, and .

Local and global moves are fused through an adaptive update rule

|  |  |  |
| --- | --- | --- |
|  |  | (5) |

where scales the step size, and and are time-varying weights defined as

|  |  |  |
| --- | --- | --- |
|  |  | (6) |
|  |  | (7) |

where, is the current number of function evaluations and is the predefined maximum.

Finally, PLO incorporates a dimension-wise refinement inspired by particle collisions in the solar wind:

|  |  |  |
| --- | --- | --- |
|  |  | (8) |

where , is a trigger threshold, and is a randomly selected peer. This refinement is applied sparingly across iterations, serving as a lightweight local search to polish candidate solutions.

## PLO Limitations and Motivation

While PLO provides a principled coupling of local exploitation (gyration), global exploration (auroral oval walk), and occasional dimension-wise refinement, its baseline dynamics can still exhibit two limiting behaviors on complex, multimodal landscapes.

First, PLO is prone to diversity erosion as the run progresses. As the population mean concentrates and the adaptive weights emphasize exploitation, individuals increasingly crowd the current basins of attraction. Because the exploratory step in Eq. (4) is anchored to , and the refinement in Eq. (8) is applied sparingly, underperforming individuals may lack sufficient impetus to escape entrenched regions, risking premature convergence. An effective diversification mechanism should therefore provide targeted, history-informed perturbations that relocate poor performers to novel regions without disrupting promising trajectories.

Second, PLO typically relies on greedy, one-to-one replacement, admitting an offspring only if it immediately improves fitness. This myopic policy can discard candidates that, despite having slightly worse fitness, contribute valuable spatial diversity (for example, by residing far from the incumbent best). When the search stalls, such a policy accelerates collapse around suboptimal attractors and reduces the algorithm's long-term exploratory capacity. A more robust survivor selection should, at least under stagnation, account for both quality and spatial contribution.

These observations motivate two complementary enhancements. To counter diversity loss, we introduce a Jumping Historical Search (JHS) strategy that leverages an archive of recently replaced parents to construct differential, event-driven jumps for underperforming individuals, synergistically augmenting the exploratory move in Eq. (4). To mitigate myopic selection, we adopt a Fitness–Diversity Balanced (FDB) survivor selection that, when stagnation is detected, merges parents and offspring, preserves an elite set by pure fitness, and fills the remaining slots using a score that balances normalized fitness with a diversity component measured via distance to the current best. The following subsections detail JHS and FDB and how they integrate with PLO's operators.

## Jumping Historical Search (JHS)

To counter diversity loss without disrupting promising trajectories, we employ an event-driven Jumping Historical Search (JHS). An external archive stores recently replaced parents. For a small fraction of underperforming individuals (bottom of the population), and only within a short window after an FDB event, JHS perturbs the exploratory candidate by a history-informed differential jump. Specifically, letting be a randomly selected archived parent and a small magnitude, we form

|  |  |  |
| --- | --- | --- |
|  |  | (9) |

and orthogonalize it against the current exploitation direction to avoid fighting the intensification drive:

|  |  |  |
| --- | --- | --- |
|  |  | (10) |

after which the exploratory move becomes

|  |  |  |
| --- | --- | --- |
|  |  | (11) |

where, is the hybrid guidance combining mean-, personal-best-, global-best-, and partner-driven components used in the baseline generator, and safeguards against division by zero. JHS is applied with a small probability to eligible individuals during a limited number of generations following an FDB trigger, providing targeted diversification precisely when stagnation is detected.

## Fitness-Diversity Balanced (FDB) Selection

To mitigate myopic replacement and restore exploratory capacity under stagnation, we adopt a Fitness–Diversity Balanced (FDB) survivor selection. When triggered, parents and offspring are merged, and survivors are chosen in two phases. First, an elite set comprising roughly half the population is selected purely by fitness. Second, the remaining survivors are chosen by minimizing a composite score that balances normalized fitness and spatial diversity (diversity implemented as the Euclidean distance to the current best in the combined pool):

|  |  |
| --- | --- |
|  | (12) |

This scheme preserves high-quality solutions while deliberately retaining individuals far from the current incumbent, preventing rapid collapse around a single basin and improving coverage of the search space. In our implementation, FDB is considered only after an initial portion of the budget has elapsed and when both low improvement and low diversity are detected; after an FDB event, a short cooldown is enforced and an event window activates JHS for the next few generations, coupling selection and diversification.

## Operational Procedure of PLOJF

The proposed PLOJF enhances the original PLO by integrating the two strategies previously described: Jumping Historical Search (JHS) and Fitness–Diversity Balanced (FDB) selection. JHS provides targeted, event-driven diversification for underperforming individuals via archive-based differential jumps (9)–(11), while FDB replaces myopic greedy replacement with an adaptive, two-phase survivor selection that balances fitness and diversity (12). Their integration follows a detection-and-response cycle: when stagnation and low diversity are detected, FDB is triggered and a short window activates JHS to diversify the subsequent generations.

The complete operational flow is outlined in Algorithm 1.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  | | --- | | **Algorithm 1** The Proposed CDPLO Algorithm | | **Input:** , , , , objective function | | **Output:** Best solution | | Initialize population ; evaluate ; sort; set | | Initialize personal bests ; archive ; set adaptation and trigger parameters | | **While** | | Compute , adaptive weights , , success rate and scale; estimate diversity | | Derive probabilities and gain ; mark bottom- individuals | | **For** | | Build local step *LS* (gyratioin) and global step *GS* (auroral) using population statistics | | Select partner (early split vs. later random in half); form hybrid guidance | | Generate two candidates: (exploit) and (explore) | | **If** JHS window active and is bottom-*q* and and | | Pick ; compute ; orthogonalize to | | Update | | **End If** | | **If** | | Apply Cauchy jump to around and | | **End If** | | **If** | | Set | | **End If** | | Boundary-control candidates; evaluate ; select better | | **If** | | Push parent into archive (bounded); accept | | **End If** | | **If** | | Update | | **End If** | | **End For** | | Update sliding success history; update no-improvement counter | | ; compute stall and diversity thresholds; check cooldown | | **If** FDB trigger: stagnation and low diversity and beyond onset | | Combine ; select survivors: top-≈ by fitness; remainder by FDB score | | Replace with selected survivors; start cooldown; set JHS boost window for next generations | | **End If** | | Decrease cooldown and JHS boost counters | | Finalize generation: ; sort; update ; update | | **End While** | | **Return** . | |

**Fig. 1.** Flowchart of PLOJF.

# Proposed bPLOJF- model

## bPLOJF feature selection algorithm

## Implementation of bPLOJF- model

# Benchmarks validation

To verify the performance of the PLOJF, we conducted comprehensive experiments in this section, including quality analysis of the algorithm with the aim of finding the hidden exploration and exploitation balance during the optimization procedure; an ablation experiment to qualitify the contribution of the proposed strategies; and the comparison experiment which compare the proposed PLOJF with the most recently proposed state-of-the-art optimization algorithm, to validate its optimization performance.

For a fair comparison, we implemented the compared algorithm with their original publication and its original parameters in a windows system with MATLAB 2024a, 13th Gen Intel(R) Core(TM) i5-13400F with 2.5 GHz, embeded with 32 GB RAM. For the experimental settings, we define the population size with the value of 30, maximum function evaluations with , thus a 10,000 iterations will be conducted for each algorithm, and each algorithm with a independent runs of 30. And the benchmark used this study is the mostly used IEEE CEC2017.

## Quality Analysis of PLOJF

The quality analysis meant to test the balance between exploration and exploitation during the optimization process, as well as the population diversity. As can been seen from the **Fig. 2**, we have included five columns information: the 3-dimentional benchmark landscape which describe the shape of the function, the balance analysis of PLO, serve as a baseline of analysis, the balance analysis of PLOJF, to test what impact did the introduced strategies bring, and the algorithm diversity analysis which coherent with the exploration metrics in during optimization, with higher exploration comes with high population diversity, vice vesa; as the convergence curve which shows the convergence speed and the solution precision. Those information provide us with the attribution of the proposed strategies, and how did their impact the convergence speed and solution quality. The detailed analysis of the result detailed below.

For this analysis, we have select the four presentative functions in the test suite: unimodal function F1, multi-modal function F4, hybrid function F11, and composition function F29. To clarify that we conduct the experiment with dimension 30, the provided funtion landscape in **Fig. 2**(a) are for a simple insight to the function shape with just three dimensionality. Seen from the **Fig. 2**(b) and **Fig. 2**(c), we found that the exploration ratio in the original PLO are very high, especially for the hybrid function F11 and composition function F29, shown as red curve in the plot, especially for the hybrid function and the composition function, the exploration rate start to decrease at the very first iterations, while near end iterations, the curve shows a increase trend. Specifically, for hybrid function F11, the trend continue with the optimization ending, leading to a higher population diversity compare with the initial population diversity which is enterintuive with the optimization convergence where population should be clustered in a local region mostly have have a relatively low population diversity at the end of the optimization. While for the composition function F29, the curve shows the increase trend while decrease immediately, and the population diversity have the same trend, higher when the exploration shows increasing and decrease when the exploration goes down. For the unimodal function F1, and multi-modal function F4, the exploration trend are simply, dreasing with the optimization process though still very high. The blue curve shown in curve is the exploitation trend of the algorithm, this is just the inverse trend of the exploration, we do not detail them as we have detail the exploration enough, and this is enough for the quality speculation of the algorithm. Also, we can seen there is a green curve shows in plot, this curve goes up when the exploration dominates the optimization, and goes down when the exploitation dominates the optimization.

Move from balance analysis of the PLO to the PLOJF, we found a interesting phenomenon, the introduced JHS and FDB strategy remedy the overly high exploration ratio of the original PLO, with the exploration goes down gradually, no trend of increasing during the optimization, this is a sign of strategy beneficial. Althouth we have repeatedly mention that the JHS and FDB helps to maintain population diversity in the proposed algorithm section, while this is for exploration-exploitation balance consideration, to prevent the algorithm stuck in the local optimal and can finally converge when the optimization ends. To seen it clearly how did the introduced strategies maintain the exploration and exploitation balance, focus on the green curve shows in the **Fig. 2**(b) and **Fig. 2**(c). For PLO plots, the green curve’s top point either too early or at the end of the optimization, while for the PLOJF, it is mostly lies between the optimization phase, this is the point where the exploration and exploitation have the same rate, after this point, comes to the exploitation phase where exploitation dominates the optimization, and diversity fastly decrease at this phase, as can be seen from the **Fig. 2**(d). The more balanced trade off between the exploration and exploitation had lead to a better search efficiency (convergence speed) and solution quality seen from the **Fig. 2**(e).

|  |
| --- |
|  |
| **Fig. 2.** (a) landscape of the functions; (b) balance analysis of PLO; (c) balance analysis of PLOJF; (d) diversity analysis; (e) best-so-far fitness curve. |

## Ablation Study

The quality of the proposed PLOJF had been discussed in the above section, the hidden balancing trade off had been seen from the introduced two strategies. In this section, we aim to found the contribution credit of each strategy. Specifically, the JHS and FDB strategy, we name the PLO variant integrated with JHS as PLOJ, and PLOF with PLO integrate with FDB, while PLOJF had integrate both strategies. In **Table 1**, the number of ‘0’ indicate no integration, and ‘1’ indicate had strategy integration. The detailed comparison result of PLO and its variants listed in the **Table 2**, with the Friedman rank (FR) result and the Wilcoxon signed-rank test (WSRT) [5] result listed in the end of the table. The Wilcoxon signed-rank test is implemented with threshold of 5%. We detail the analysis of those results below.

The best result of each function gained by the algorithm had been bold for better recognition, see from the statistic result, we found that PLOJF outperform 3 functions with PLOJ and PLOF, while the rest of 26 functions had no statistic significant. For this end we may think the JHS and FDB strategy had the same contribution, but look carefully in the FR result, the PLOF obtained a mean rank result of 2.2414 which is better than the PLOJ’s 2.3793. Based on this result, we can seen that the FDB strategy had a bigger contribution in the overall optimization performance of the PLOJF. Also the result indicate that the PLOJF only marginal exceed the mean rank of PLOF, with a value 2.2069, does that mean there is no need for the FDB strategy? The answer is no. For both strategy had been introduced for remedy the limitation of the PLO algorithm, for each of strategy integration, it may only improve a specific aspect of the algorithm, e.g., one may push PLOJF had a superior performance on the hybrid function, another may push PLOJF had a superior performance on the composition function. Those two combined will have the best result seen from the PLOJF. The convergence curves can been seem from the **Fig. 3**, aside by the boxplot, from both plot combination, we aim to see the performance consistency and stability.

Six funtions are selected for presentation, unimodal function F1, multi-modal function F4, hybrid function F10, and F12, composition function F20 and F27. The curves shown in the F1 obvisouly indicate that the PLOJF lead the optimization performance, and the boxplot aside shows that the performance stability and the consistency is across the 30 independent runs, with a dense and lower box. This indicate that the PLOJF had a good exploitation ability, as there is only one global optimal in the unimodal function, mostly those kind of functions are testing the performance of the exploitation ability. While for the multi-modal function F2, PLOF lead the optimization performance, with a lower mean fitness value while bigger box shown in the boxplot, for PLOJ, it have a bettter optimization stability while have a worse solution quality compare with the PLOF and PLOJF. This result indidcate that the exploration ability of the PLOF is better than the PLOJF, while the JHS integration had worsen the exploration ability of PLOF.

Two hybrid function and two composition function curves show that the PLOFJ lead the optimziation performance, but look closely we can see that the PLOJ have better performance compare with the PLOF in the hybrid function, while for the composition function, PLOF have better optimization performance compare with the PLOJ. The PLOJF benefit from both strategies integration, lead the best performance of all four functions.

The Friedman mean rank result (ARV) and the final rank is ploted in the **Fig. 4**(a), and each algorithm rank per function plotted in the **Fig. 4**(b). The FR result had been discussed above, the each functions rank can give us more insight about the beneficial of those two strategy. The plot shows that the red line (PLOJF) have a bigger wrap range in the radar plot, indicate a better overall performance, also we can seen that in some functions, the PLOJF had worse performance compare with the PLOJ or PLOF, this is because the two strategies integration are meant to improve the PLOJF in the total benchmark suite, thus a balance is needed for it, while this indicate that the PLOJ and PLOF both have their advantages in that kind of function optimization.

**Table 1.** The PLO and its variations.

|  |  |  |
| --- | --- | --- |
|  | Cluster guidance | Differential recombination |
| PLOJF | 1 | 1 |
| PLOJ | 0 | 0 |
| PLOF | 1 | 0 |
| PLO | 0 | 0 |

**Fig. 3.** Convergence curves of PLOJF and its two single-component variants on six test functions.

**Fig. 4.** (a) Friedman Ranking (ARV) and final rank results (b) function-wise average rank

**Table 2**. Comparison results of PLO and its variants.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **F1** | | **F2** | | **F3** | | **F4** | | **F5** | | **F6** | | **F7** | | **F8** | |
| **Algorithms** | **Avg** | **Std** | **Avg** | **Std** | **Avg** | **Std** | **Avg** | **Std** | **Avg** | **Std** | **Avg** | **Std** | **Avg** | **Std** | **Avg** | **Std** |
| PLOJF | **1.7303E+03** | **1.3239E+03** | 4.1028E+03 | **1.2736E+03** | 4.8786E+02 | 1.9026E+01 | 5.3594E+02 | 1.0969E+01 | 6.0097E+02 | 4.3909E-01 | 7.5793E+02 | 9.6643E+00 | 8.3827E+02 | 9.5599E+00 | 9.2494E+02 | 1.6917E+01 |
| PLOJ | 2.1935E+04 | 6.6595E+03 | 3.9712E+03 | 1.4594E+03 | 4.9138E+02 | **1.6512E+01** | **5.3531E+02** | **7.3747E+00** | 6.0086E+02 | **2.7021E-01** | 7.5909E+02 | **7.3332E+00** | 8.3772E+02 | 8.3749E+00 | **9.2030E+02** | **1.4510E+01** |
| PLOF | 1.9599E+03 | 1.3745E+03 | **3.8440E+03** | 1.3224E+03 | 4.8623E+02 | 1.6545E+01 | 5.3671E+02 | 8.8772E+00 | **6.0086E+02** | 3.9697E-01 | **7.5758E+02** | 9.1922E+00 | **8.3709E+02** | 8.1306E+00 | 9.2347E+02 | 1.4977E+01 |
| PLO | 1.0354E+04 | 1.9824E+03 | 2.4367E+04 | 5.5231E+03 | **4.7093E+02** | 1.6839E+01 | 5.5130E+02 | 8.7525E+00 | 6.0438E+02 | 1.0772E+00 | 8.2139E+02 | 1.6624E+01 | 8.4916E+02 | **7.5873E+00** | 1.3179E+03 | 1.3174E+02 |
|  | **F9** | | **F10** | | **F11** | | **F12** | | **F13** | | **F14** | | **F15** | | **F16** | |
| **Algorithms** | **Avg** | **Std** | **Avg** | **Std** | **Avg** | **Std** | **Avg** | **Std** | **Avg** | **Std** | **Avg** | **Std** | **Avg** | **Std** | **Avg** | **Std** |
| PLOJF | 3.0337E+03 | 3.2654E+02 | **1.1431E+03** | **9.9888E+00** | **3.8702E+04** | **3.8964E+04** | **2.8424E+03** | 5.4914E+02 | 1.6622E+03 | 6.6183E+01 | 1.8881E+03 | 1.7365E+02 | 1.9308E+03 | 1.1769E+02 | 1.8283E+03 | 2.6901E+01 |
| PLOJ | 3.0726E+03 | **2.6879E+02** | 1.1450E+03 | 1.0736E+01 | 6.1857E+04 | 7.4996E+04 | 2.9049E+03 | **4.3370E+02** | **1.6483E+03** | **5.2070E+01** | **1.8782E+03** | 1.6709E+02 | 1.9622E+03 | 1.4250E+02 | **1.8197E+03** | **2.4278E+01** |
| PLOF | 3.0068E+03 | 3.0701E+02 | 1.1523E+03 | 1.6855E+01 | 5.5335E+04 | 7.0319E+04 | 3.1524E+03 | 6.0873E+02 | 1.6547E+03 | 6.0906E+01 | 1.8982E+03 | **1.2282E+02** | 1.9369E+03 | 1.3328E+02 | 1.8305E+03 | 3.4987E+01 |
| PLO | **3.0043E+03** | 2.9511E+02 | 1.1825E+03 | 1.9357E+01 | 8.3586E+05 | 8.5463E+05 | 6.9967E+03 | 1.7915E+03 | 2.6437E+03 | 8.9399E+02 | 4.0526E+03 | 9.4256E+02 | **1.8883E+03** | **9.6587E+01** | 1.8422E+03 | 3.6553E+01 |
|  | **F17** | | **F18** | | **F19** | | **F20** | | **F21** | | **F22** | | **F23** | | **F24** | |
| **Algorithms** | **Avg** | **Std** | **Avg** | **Std** | **Avg** | **Std** | **Avg** | **Std** | **Avg** | **Std** | **Avg** | **Std** | **Avg** | **Std** | **Avg** | **Std** |
| PLOJF | 9.7709E+03 | 5.4658E+03 | **2.0012E+03** | 2.2840E+01 | 2.1714E+03 | 3.7518E+01 | **2.2000E+03** | 8.2390E-04 | 2.3000E+03 | 1.2011E-03 | 2.8322E+03 | 1.3707E+01 | 2.7020E+03 | 3.2063E-01 | 2.9087E+03 | 8.9420E+00 |
| PLOJ | 9.8145E+03 | **4.2782E+03** | 2.0029E+03 | **2.0801E+01** | 2.1660E+03 | 5.2004E+01 | 2.2000E+03 | 1.9721E-03 | 2.3000E+03 | 1.9042E-03 | **2.8310E+03** | 1.2436E+01 | 2.7020E+03 | **2.7057E-01** | 2.9082E+03 | 7.2848E+00 |
| PLOF | **9.4685E+03** | 5.1112E+03 | 2.0091E+03 | 2.3601E+01 | **2.1562E+03** | **3.1180E+01** | 2.2000E+03 | 1.3391E-03 | **2.3000E+03** | 1.0719E-03 | 2.8345E+03 | 1.0175E+01 | **2.7019E+03** | 3.6654E-01 | 2.9100E+03 | 8.3938E+00 |
| PLO | 7.9841E+04 | 3.7126E+04 | 2.2801E+03 | 1.3418E+02 | 2.2032E+03 | 5.3927E+01 | 2.2000E+03 | **5.7897E-04** | 2.3000E+03 | **6.7509E-04** | 2.8352E+03 | **9.4950E+00** | 3.3887E+03 | 7.5886E+00 | **2.9031E+03** | **4.3674E+00** |
|  | **F25** | | **F26** | | **F27** | | **F28** | | **F29** | | **Statistical comparisons** | | |  |  |  |
| **Algorithms** | **Avg** | **Std** | **Avg** | **Std** | **Avg** | **Std** | **Avg** | **Std** | **Avg** | **Std** | WSRT | FR | Rank |  |  |  |
| PLOJF | 4.7892E+03 | **1.0484E+02** | 3.4872E+03 | 3.4760E+01 | **3.2218E+03** | 5.3347E+01 | 3.3975E+03 | 6.2773E+01 | 1.4536E+04 | 7.5272E+03 | **~** | **2.2069** | **1** |  |  |  |
| PLOJ | 4.7238E+03 | 3.7505E+02 | 3.5000E+03 | 2.7372E+01 | 3.2312E+03 | 4.1562E+01 | 3.4122E+03 | 6.0554E+01 | **1.1874E+04** | 3.8632E+03 | 3/0/26 | 2.3793 | 3 |  |  |  |
| PLOF | 4.7686E+03 | 1.1319E+02 | 3.5079E+03 | 4.0333E+01 | 3.2239E+03 | 4.3692E+01 | 3.4262E+03 | 5.4965E+01 | 1.2234E+04 | **3.4126E+03** | 3/0/26 | 2.2414 | 2 |  |  |  |
| PLO | **4.5223E+03** | 6.5304E+02 | **3.4379E+03** | **1.8449E+01** | 3.2348E+03 | **2.2368E+01** | **3.3822E+03** | **5.2750E+01** | 5.1506E+04 | 2.0131E+04 | 18/3/8 | 3.1724 | 4 |  |  |  |

## Testing on IEEE CEC 2017

The attribution analysis had been conducted in the ablation study above, in this section, we try to validate the overall optimization performance of the PLOJF, the copmared algorithms in this study including Exponential-Trigonometric Optimization (ETO), Moss Growth Optimization (MGO), Status-based Optimization (SBO), Multi-strategy Artemisinin Optimization (MSAO), Comprehensive Learning Particle Swarm Optimization Algorithm (CLPSO), SHADE with Linear population size reduction (LSHADE), Roulette wheel and Mutation improved RIME algorithm (RMRIME), Migration and Divergent thinking improved PLO algorithm (MDPLO), and Powell mechanism and DE improved Slime Mould Algorithm (PSMADE). Those compared peers including three recently proposed meta-heuristic algorithms (MA): ETO, MGO, and SBO. With four recently proposed advanced algorithms: MSAO, RMRIME, MDPLO, and PSMADE. Two classic and champion algorithm: CLPSO and LSHADE.

The detailed comparison result are show in **Table 3**, with the statistic result listed in the end of the table. As per WSRT result, the proposed PLOJF outperform 28, 23, 18, 9, 16, 11, 19, 14, and 26 functions compare with ETO, MGO, SBO, MSAO, CLPSO, LSHADE, RMRIME, MDPLO, and PSMADE, while underperform 0, 3, 4, 10, 9, 13, 3, 8, and 1 functions. With 1, 3, 7, 10, 4, 5, 7, 7, and 2 functions without statistic significance. From the FR result, the PLOJF obtained a value of 3.3448, with the leading optimization performance, follows by the LSHADE’s 3.6207, the once champion algorithm in IEEE CEC 2017 benchmark test suite. The visulized rank result and per function rank can be shown in the **Fig. 5**, the FR result can be seen from the **Fig. 5**(a), this shows the general optimization performance of each algorithms, while we are more interested in **Fig. 5**(b), this function wise rank provide more information about each algorithms advantage and disadavantage. The MDPLO shows good optimization performance in the single-modal functions are some of the composite functions, while for most of the multi-modal functions, hybrid functions, and rest of the composite functions, the proposed PLOJF have a general better performance. This indicate that the exploitation ability of the PLOJF still have the improvement room, since the JHS and FDB strategies are introduced to balance the convergence and diversity, with no target of improve the exploitation ability of PLO.

For a better analysis of each algorithm’s convergence quality and its optimization stability and consistency, we provide six representative functions in **Fig. 6**, including unimodal function F1, multi-modal function F6, hybrid function F16 and F19, and composition function F21 and F23. As we discussed above, there still have exploitation ability improvement room for the proposed PLOJF, the convergence result seen from F1 regain validate the statement, even a total maximum of 300,000 function evaluations are not enough for the PLOJF optimization, as it continuously shows a decreasing trend in the curve, while for other optimizers, like LSHADE, and MDPLO, they convergence insanely fast, only spend a samll portion of the function evaluations to finish the optimization. While for the rest of the selected functions, PLOJF exceed other algorithms performance, and the following good performance algorithms are not identical, this shows a good sign of optimization generalization, and overall good optimization ability, seen from the boxplot aside of the convergence speed, the most unstable optimization stability is the optimizer ETO, while for the rest of the algorithms, their optimization stability distribute across functions, with some of the functions, they show stability, while others, they are not. For proposed PLOJF, among those algorithm PLOJF always shows a dense box and lower mean fitness value, this validate the proposed algorithm have a good global optimization ability across different problems.

The introduced strategies improve the overall optimization ability of the PLO, and a real-world feature selection task will be conducted in the following section to test the practical application of the proposed method.

**Fig. 5.** (a) Friedman Ranking (ARV) and final rank results (b) function-wise average rank

**Fig. 6.** Convergence behavior of PLOJF versus nine competitors on six representative CEC 2017 functions.

**Table 3.** Comparison results of PLOJF and other algorithms.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **F1** | | **F2** | | **F3** | | **F4** | | **F5** | | **F6** | | **F7** | | **F8** | |
| **Algorithms** | **Avg** | **Std** | **Avg** | **Std** | **Avg** | **Std** | **Avg** | **Std** | **Avg** | **Std** | **Avg** | **Std** | **Avg** | **Std** | **Avg** | **Std** |
| PLOJF | 2.5548E+03 | 1.7382E+03 | 3.6652E+03 | 1.1267E+03 | 4.9011E+02 | 1.5530E+01 | 5.3644E+02 | 8.4346E+00 | 6.0372E+02 | 1.1965E+00 | **7.5770E+02** | 8.3902E+00 | 8.5089E+02 | 1.0908E+01 | 9.1296E+02 | **9.5929E+00** |
| ETO | 6.6936E+09 | 2.8619E+09 | 4.1797E+04 | 1.0070E+04 | 1.1408E+03 | 6.5574E+02 | 7.4486E+02 | 3.1953E+01 | 6.6615E+02 | 1.1378E+01 | 1.0636E+03 | 5.2716E+01 | 9.7161E+02 | 3.1818E+01 | 5.9474E+03 | 1.5832E+03 |
| MGO | 1.2824E+05 | 2.0040E+05 | 6.2408E+03 | 2.3290E+03 | 4.8714E+02 | 1.1794E+01 | 5.4956E+02 | 9.1651E+00 | 6.0314E+02 | 1.5852E+00 | 7.8012E+02 | 9.5040E+00 | 8.6282E+02 | 1.1247E+01 | 9.0757E+02 | 1.0796E+01 |
| SBO | 1.1036E+02 | 3.1532E+01 | 3.0000E+02 | 2.0853E-03 | 4.5934E+02 | 3.5772E+01 | 6.3594E+02 | 3.1961E+01 | 6.3892E+02 | 1.3801E+01 | 8.8744E+02 | 4.9506E+01 | 9.0987E+02 | 2.1396E+01 | 1.8385E+03 | 7.8406E+02 |
| MSAO | 4.0427E+03 | 5.7660E+03 | 3.0024E+02 | 8.9817E-02 | 4.8696E+02 | 9.4448E+00 | 5.3596E+02 | 8.4904E+00 | **6.0025E+02** | **3.0015E-01** | 7.7426E+02 | 1.1906E+01 | 8.5259E+02 | 8.7155E+00 | **9.0506E+02** | 3.8528E+01 |
| CLPSO | 1.6251E+02 | 1.3027E+02 | 8.4888E+03 | 1.9344E+03 | 4.6594E+02 | 2.2679E+01 | 5.4636E+02 | **7.5757E+00** | 6.0692E+02 | 1.7075E+00 | 7.8292E+02 | **6.7396E+00** | 8.4542E+02 | **7.5278E+00** | 9.1272E+02 | 1.4764E+01 |
| LSHADE | **1.0000E+02** | **3.7776E-11** | 1.0572E+04 | 2.6054E+04 | 4.3689E+02 | 2.8560E+01 | **5.3299E+02** | 1.0317E+01 | 6.0684E+02 | 3.9003E+00 | 7.8538E+02 | 2.0603E+01 | **8.3866E+02** | 9.9300E+00 | 1.0070E+03 | 9.8593E+01 |
| RMRIME | 1.1468E+02 | 8.8360E+01 | 3.3355E+02 | 3.4869E+01 | 4.8293E+02 | 3.0169E+01 | 5.5387E+02 | 1.3559E+01 | 6.0647E+02 | 2.8435E+00 | 7.9354E+02 | 1.2699E+01 | 8.5533E+02 | 1.1860E+01 | 9.2417E+02 | 2.4479E+01 |
| MDPLO | 1.0000E+02 | 2.1893E-07 | **3.0000E+02** | **1.2971E-12** | **4.2793E+02** | 3.1963E+01 | 5.7823E+02 | 2.6819E+01 | 6.0591E+02 | 5.8285E+00 | 8.6291E+02 | 3.5361E+01 | 8.9547E+02 | 2.8218E+01 | 9.7147E+02 | 1.9489E+02 |
| PSMADE | 2.4487E+06 | 2.4809E+06 | 4.1178E+04 | 1.8382E+04 | 4.9163E+02 | **5.5848E+00** | 7.2302E+02 | 2.0560E+01 | 6.0721E+02 | 2.4008E+00 | 9.7051E+02 | 1.6008E+01 | 1.0072E+03 | 1.7392E+01 | 9.7444E+02 | 6.5071E+01 |
|  | **F9** | | **F10** | | **F11** | | **F12** | | **F13** | | **F14** | | **F15** | | **F16** | |
| **Algorithms** | **Avg** | **Std** | **Avg** | **Std** | **Avg** | **Std** | **Avg** | **Std** | **Avg** | **Std** | **Avg** | **Std** | **Avg** | **Std** | **Avg** | **Std** |
| PLOJF | 3.3540E+03 | **2.7813E+02** | 1.1473E+03 | **1.9767E+01** | 2.4156E+04 | 1.7598E+04 | 4.9273E+03 | 1.4474E+03 | 1.7160E+03 | **6.0114E+01** | 2.0077E+03 | 1.5845E+02 | 2.0513E+03 | 1.5044E+02 | **1.7841E+03** | **2.9444E+01** |
| ETO | 7.0952E+03 | 7.3017E+02 | 2.1735E+03 | 9.5632E+02 | 3.3151E+08 | 4.0031E+08 | 2.3583E+08 | 8.4978E+08 | 3.2660E+05 | 4.6285E+05 | 2.1288E+06 | 8.5410E+06 | 3.0876E+03 | 3.3846E+02 | 2.2758E+03 | 2.0474E+02 |
| MGO | 3.7587E+03 | 3.9508E+02 | 1.1728E+03 | 2.4751E+01 | 8.2236E+05 | 8.2445E+05 | 2.2206E+04 | 1.4658E+04 | 7.6604E+03 | 5.4185E+03 | 1.1599E+04 | 8.0179E+03 | 2.1521E+03 | **1.3575E+02** | 1.8514E+03 | 4.4327E+01 |
| SBO | 4.1538E+03 | 6.5582E+02 | 1.1967E+03 | 3.4907E+01 | 2.6416E+04 | 1.3810E+04 | 1.4900E+04 | 1.7529E+04 | 2.7499E+03 | 2.2105E+03 | 2.4721E+03 | 2.1631E+03 | 2.6271E+03 | 3.1388E+02 | 2.1885E+03 | 2.2782E+02 |
| MSAO | 3.1287E+03 | 4.9621E+02 | **1.1248E+03** | 2.0903E+01 | 2.1062E+05 | 1.8270E+05 | 1.0856E+04 | 1.5785E+04 | 7.3492E+03 | 4.8205E+03 | 1.0439E+04 | 1.0616E+04 | **1.9975E+03** | 1.7906E+02 | 1.8817E+03 | 1.2646E+02 |
| CLPSO | 3.3242E+03 | 3.2692E+02 | 1.1507E+03 | 2.1353E+01 | 5.0669E+05 | 3.2326E+05 | **1.6650E+03** | **3.2598E+02** | 3.1002E+04 | 3.2375E+04 | **1.6626E+03** | **1.2042E+02** | 2.1170E+03 | 1.4182E+02 | 1.8287E+03 | 5.9257E+01 |
| LSHADE | **3.0758E+03** | 3.8301E+02 | 1.2371E+03 | 5.4015E+01 | **9.4222E+03** | 9.9807E+03 | 1.8051E+03 | 7.6735E+02 | **1.6578E+03** | 8.1614E+01 | 1.7484E+03 | 1.2971E+02 | 2.1523E+03 | 1.7767E+02 | 1.8766E+03 | 1.0255E+02 |
| RMRIME | 3.5875E+03 | 4.3267E+02 | 1.1735E+03 | 3.1389E+01 | 3.6006E+04 | 1.7325E+04 | 1.5146E+04 | 1.4294E+04 | 1.8616E+03 | 5.9389E+02 | 3.7460E+03 | 3.3181E+03 | 2.1019E+03 | 1.7804E+02 | 1.8529E+03 | 9.4521E+01 |
| MDPLO | 5.1236E+03 | 7.8176E+02 | 1.1764E+03 | 4.3282E+01 | 1.4019E+04 | **9.4766E+03** | 4.6389E+03 | 7.9109E+03 | 1.7094E+03 | 9.8410E+01 | 1.8631E+03 | 2.4373E+02 | 2.2665E+03 | 3.3207E+02 | 1.9025E+03 | 9.6048E+01 |
| PSMADE | 6.0410E+03 | 1.5980E+03 | 1.2575E+03 | 2.2561E+01 | 3.9547E+05 | 3.6566E+05 | 2.4716E+04 | 2.3135E+04 | 2.4783E+03 | 1.0046E+03 | 1.3011E+04 | 1.1819E+04 | 2.6532E+03 | 5.2555E+02 | 2.0577E+03 | 1.2916E+02 |
|  | **F17** | | **F18** | | **F19** | | **F20** | | **F21** | | **F22** | | **F23** | | **F24** | |
| **Algorithms** | **Avg** | **Std** | **Avg** | **Std** | **Avg** | **Std** | **Avg** | **Std** | **Avg** | **Std** | **Avg** | **Std** | **Avg** | **Std** | **Avg** | **Std** |
| PLOJF | 1.3097E+04 | 8.7873E+03 | 2.0469E+03 | **3.3102E+01** | **2.1463E+03** | **4.3216E+01** | 2.3403E+03 | 1.0304E+01 | **2.3727E+03** | **4.0329E+02** | 2.6874E+03 | 1.0200E+01 | **2.8533E+03** | 9.5125E+00 | 2.8873E+03 | **5.3033E-01** |
| ETO | 1.2618E+06 | 1.6200E+06 | 9.5149E+05 | 5.7953E+05 | 2.6660E+03 | 2.0110E+02 | 2.5325E+03 | 3.3682E+01 | 7.9258E+03 | 1.9361E+03 | 2.9575E+03 | 4.4989E+01 | 3.1724E+03 | 4.4715E+01 | 3.1364E+03 | 1.1324E+02 |
| MGO | 2.0608E+05 | 1.2030E+05 | 8.8837E+03 | 8.7653E+03 | 2.2003E+03 | 6.7727E+01 | 2.3523E+03 | 1.4624E+01 | 2.7555E+03 | 1.0597E+03 | 2.7064E+03 | 1.2363E+01 | 2.8854E+03 | 1.2115E+01 | 2.8869E+03 | 8.9645E-01 |
| SBO | 1.6295E+04 | 1.3984E+04 | 2.3177E+03 | 1.0041E+03 | 2.3911E+03 | 1.4601E+02 | 2.4197E+03 | 3.2721E+01 | 2.3829E+03 | 6.5211E+02 | 2.7744E+03 | 3.2398E+01 | 2.9798E+03 | 4.4012E+01 | 2.8944E+03 | 1.4845E+01 |
| MSAO | 1.4946E+05 | 1.2302E+05 | 9.0459E+03 | 9.4192E+03 | 2.2377E+03 | 1.1410E+02 | 2.3423E+03 | 9.5912E+00 | 3.1728E+03 | 1.1413E+03 | **2.6850E+03** | **7.9050E+00** | 2.8566E+03 | **8.2236E+00** | 2.8866E+03 | 1.0004E+00 |
| CLPSO | 1.5824E+05 | 9.2782E+04 | **2.0296E+03** | 1.6150E+02 | 2.1909E+03 | 7.7769E+01 | 2.3410E+03 | 4.0733E+01 | 2.5620E+03 | 7.1610E+02 | 2.7015E+03 | 9.2433E+00 | 2.8747E+03 | 8.4901E+01 | **2.8865E+03** | 9.4399E-01 |
| LSHADE | **2.7763E+03** | **1.8751E+03** | 2.0504E+03 | 8.2287E+01 | 2.2071E+03 | 9.4722E+01 | **2.3361E+03** | **9.0875E+00** | 2.6314E+03 | 8.3476E+02 | 2.6875E+03 | 1.4849E+01 | 2.8552E+03 | 1.0528E+01 | 2.8887E+03 | 6.9369E+00 |
| RMRIME | 2.6513E+04 | 2.6674E+04 | 3.2410E+03 | 2.1868E+03 | 2.1592E+03 | 8.7091E+01 | 2.3547E+03 | 1.3946E+01 | 2.7185E+03 | 1.0448E+03 | 2.7116E+03 | 1.6150E+01 | 2.8917E+03 | 1.9268E+01 | 2.8948E+03 | 1.4008E+01 |
| MDPLO | 3.2593E+03 | 2.6854E+03 | 2.0508E+03 | 7.3673E+01 | 2.2008E+03 | 1.2959E+02 | 2.3641E+03 | 2.2187E+01 | 4.2978E+03 | 2.4856E+03 | 2.7051E+03 | 2.2800E+01 | 2.8681E+03 | 1.1992E+01 | 2.8884E+03 | 7.2501E+00 |
| PSMADE | 2.3543E+05 | 1.9430E+05 | 1.8230E+04 | 1.5676E+04 | 2.5348E+03 | 1.7429E+02 | 2.5143E+03 | 2.8182E+01 | 7.6739E+03 | 1.9220E+03 | 2.8536E+03 | 3.9406E+01 | 3.0351E+03 | 1.8759E+01 | 2.8894E+03 | 4.9854E+00 |
|  | **F25** | | **F26** | | **F27** | | **F28** | | **F29** | | **Statistical comparisons** | | |  |  | |
| **Algorithms** | **Avg** | **Std** | **Avg** | **Std** | **Avg** | **Std** | **Avg** | **Std** | **Avg** | **Std** | **WSRT** | **FR** | **Rank** |  |  |  |
| PLOJF | 3.9494E+03 | **1.2923E+02** | 3.2073E+03 | 5.0594E+00 | 3.1962E+03 | 2.9722E+01 | 3.5023E+03 | **5.2755E+01** | 1.5330E+04 | 3.9447E+03 | ~ | **3.3448** | **1** |  |  |  |
| ETO | 6.4218E+03 | 7.5305E+02 | 3.4121E+03 | 8.2455E+01 | 3.6959E+03 | 3.3138E+02 | 4.2898E+03 | 2.7360E+02 | 9.7965E+06 | 5.7785E+06 | 28/0/1 | 9.9655 | 10 |  |  |  |
| MGO | 4.0624E+03 | 3.7565E+02 | 3.2086E+03 | 4.8375E+00 | 3.2223E+03 | 1.2933E+01 | 3.5848E+03 | 7.6549E+01 | 6.0372E+04 | 5.0052E+04 | 23/3/3 | 5.8276 | 7 |  |  |  |
| SBO | 4.0212E+03 | 1.5338E+03 | 3.2475E+03 | 1.9851E+01 | 3.1647E+03 | 5.4346E+01 | 3.7622E+03 | 2.2381E+02 | 1.0662E+04 | 3.6715E+03 | 18/4/7 | 6.3793 | 8 |  |  |  |
| MSAO | 3.8870E+03 | 3.1322E+02 | 3.1996E+03 | 8.3374E+00 | 3.1926E+03 | 3.4833E+01 | **3.4104E+03** | 5.8457E+01 | 8.0182E+03 | 2.4548E+03 | 9/10/10 | 3.8966 | 3 |  |  |  |
| CLPSO | **3.3748E+03** | 4.7478E+02 | 3.2116E+03 | **4.4088E+00** | 3.2129E+03 | **4.1882E+00** | 3.4640E+03 | 6.8182E+01 | 8.8599E+03 | 1.7272E+03 | 16/9/4 | 3.8966 | 4 |  |  |  |
| LSHADE | 4.0782E+03 | 1.4786E+02 | 3.2209E+03 | 1.2783E+01 | 3.1577E+03 | 6.2131E+01 | 3.4664E+03 | 1.1173E+02 | **5.7240E+03** | 1.7591E+03 | 11/13/5 | 3.6207 | 2 |  |  |  |
| RMRIME | 4.2976E+03 | 3.8882E+02 | 3.2140E+03 | 8.6753E+00 | 3.2135E+03 | 1.9123E+01 | 3.4686E+03 | 7.7301E+01 | 7.8858E+03 | 2.8797E+03 | 19/3/7 | 5.3103 | 6 |  |  |  |
| MDPLO | 3.8799E+03 | 6.2643E+02 | 3.2199E+03 | 1.2990E+01 | **3.1330E+03** | 5.4695E+01 | 3.5428E+03 | 1.5612E+02 | 6.1476E+03 | **1.6912E+03** | 14/8/7 | 4.5172 | 5 |  |  |  |
| PSMADE | 5.7954E+03 | 2.5953E+02 | **3.1985E+03** | 7.7453E+00 | 3.2475E+03 | 2.8499E+01 | 3.6788E+03 | 2.1535E+02 | 1.1811E+04 | 3.8600E+03 | 26/1/2 | 8.2414 | 9 |  |  |  |

# Case study

# Conclusions and future perspectives

# Reference

Uncategorized References

1. Ahmed, M., R. Seraj, and S.M.S. Islam, *The k-means algorithm: A comprehensive survey and performance evaluation.* Electronics, 2020. **9**(8): p. 1295.

2. Feoktistov, V., *Differential evolution*. 2006: Springer.

3. Civicioglu, P., *Backtracking search optimization algorithm for numerical optimization problems.* Applied Mathematics and computation, 2013. **219**(15): p. 8121-8144.

4. Wu, G., R. Mallipeddi, and P.N. Suganthan, *Problem definitions and evaluation criteria for the CEC 2017 competition on constrained real-parameter optimization.* National University of Defense Technology, Changsha, Hunan, PR China and Kyungpook National University, Daegu, South Korea and Nanyang Technological University, Singapore, Technical Report, 2017.

5. Woolson, R.F., *Wilcoxon signed‐rank test.* Encyclopedia of biostatistics, 2005. **8**.

6. !!! INVALID CITATION !!! .

7. Wang, J., et al., *The Status-based Optimization: Algorithm and comprehensive performance analysis.* Neurocomputing, 2025: p. 130603.

8. Ghasemi, M., et al., *Ivy Algorithm: A Novel and Efficient Metaheuristic with its Applications to Engineering Optimization.* Available at SSRN 4602579.

9. Qu, Z., et al., *Power cyber-physical system risk area prediction using dependent Markov chain and improved grey wolf optimization.* IEEE Access, 2020. **8**: p. 82844-82854.

10. Ji, Y., et al., *Advancing bankruptcy prediction: a study on an improved rime optimization algorithm and its application in feature selection.* International Journal of Machine Learning and Cybernetics, 2025: p. 1-39.

11. Li, X., et al., *Advanced slime mould algorithm incorporating differential evolution and Powell mechanism for engineering design.* Iscience, 2023. **26**(10).

12. Chen, H., et al., *An efficient double adaptive random spare reinforced whale optimization algorithm.* Expert Systems with Applications, 2020. **154**: p. 113018.

13. Ma, B., et al., *Enhanced sparrow search algorithm with mutation strategy for global optimization.* IEEE Access, 2021. **9**: p. 159218-159261.

14. Chen, X., et al., *Parameters identification of solar cell models using generalized oppositional teaching learning based optimization.* Energy, 2016. **99**: p. 170-180.