Comparison of Gradient-Free Optimization Algorithms: PSO vs. CBO

Tytus Felbor

January 7, 2026

Contents

1 Introduction

Optimization is the mathematical discipline of finding the best element from some set of available alternatives. In the machine learning paradigm, this typically involves minimizing a cost function $f(x)$ to find the optimal parameters for a model.

While gradient-based methods are standard, they often converge to local minima in complex landscapes. The algorithms analyzed here—Particle Swarm Optimization (PSO) and Consensus-Based Optimization (CBO)—are part of a class of meta-heuristic, bio-inspired algorithms. They are "gradient-free," meaning they do not require the function to be differentiable, making them highly robust for global optimization in non-convex or high-dimensional problems.

2 Mathematical Dynamics

2.1 Particle Swarm Optimization (PSO)

PSO mimics the social behavior of flocks. Agents update their velocity V_i and position x_i based on personal bests b_i and the swarm's global best g:

$$V_i^{(k+1)} = \omega V_i^{(k)} + C_1 r_1 (b_i^{(k)} - x_i^{(k)}) + C_2 r_2 (g^{(k)} - x_i^{(k)}) \tag{1}$$

2.2 Consensus-Based Optimization (CBO)

CBO is based on opinion formation dynamics. It moves the population toward a weighted baricenter m^α calculated via a Gibbs distribution:

$$m^\alpha = \frac{\sum x_i e^{-\alpha f(x_i)}}{\sum e^{-\alpha f(x_i)}} \tag{2}$$

The update includes Brownian motion for exploration:

$$X_i^{(k+1)} = X_i^{(k)} + \lambda(m^\alpha - X_i^{(k)})\Delta t + \sigma\sqrt{\Delta t}\|m^\alpha - X_i^{(k)}\| Z_i^{(k)} \tag{3}$$

3 Implementation

The following snippets illustrate the core update logic for both algorithms implemented in Python.

```
# Velocity update based on personal and global bests
v = w * v + c1 * r1 * (p_best - x) + c2 * r2 * (g_best - x)
x = x + v
# Boundary constraint handling
x = np.clip(x, a, b)
```

Listing 1: PSO Velocity and Position Update

```
# Compute weighted consensus moment
weights = np.exp(-alpha * (fx - np.min(fx)))
m_alpha = np.sum(x * weights[:, None], axis=0) / np.sum(weights)

# Update with drift and Brownian motion
drift = lambd * (m_alpha - x) * dt
noise = sigma * np.sqrt(dt) * np.linalg.norm(x - m_alpha, axis=1)[:, None] * np.random.
    randn(n, d)
x = x + drift + noise
```

Listing 2: CBO Consensus and Stochastic Update

4 Cost Function Visualizations

To validate the algorithms, three benchmark functions were used: the Sphere (convex), Rastrigin (multimodal), and Ackley (steep central hole).

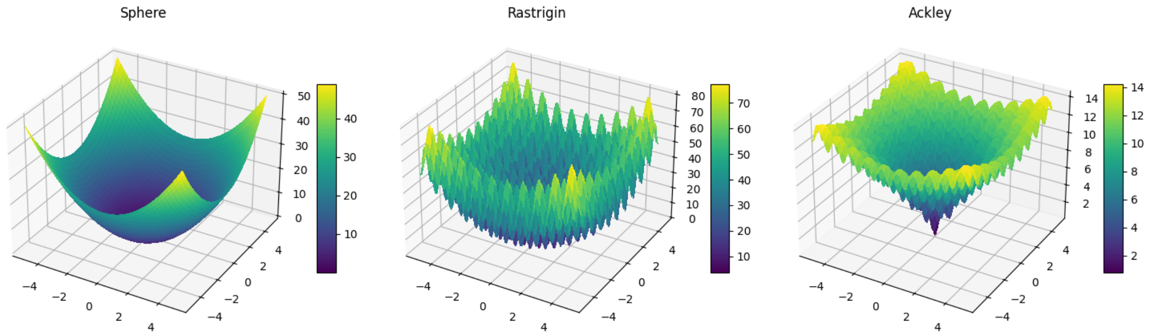

Figure 1: Shapes Of The Cost Functions

5 Results

Table 1: Performance Metrics

Function	Algorithm	Final Value (Error)	Budget (Evals)	CPU Time (s)
Sphere	PSO	1.469×10^{-21}	7,650	0.0056
	CBO	5.558×10^{-4}	5,950	0.0052
Rastrigin	PSO	0.000×10^{0}	9,050	0.0068
	CBO	1.094×10^{-2}	5,550	0.0059
Ackley	PSO	7.987×10^{-10}	7,450	0.0069
	CBO	1.254×10^{-2}	6,050	0.0067

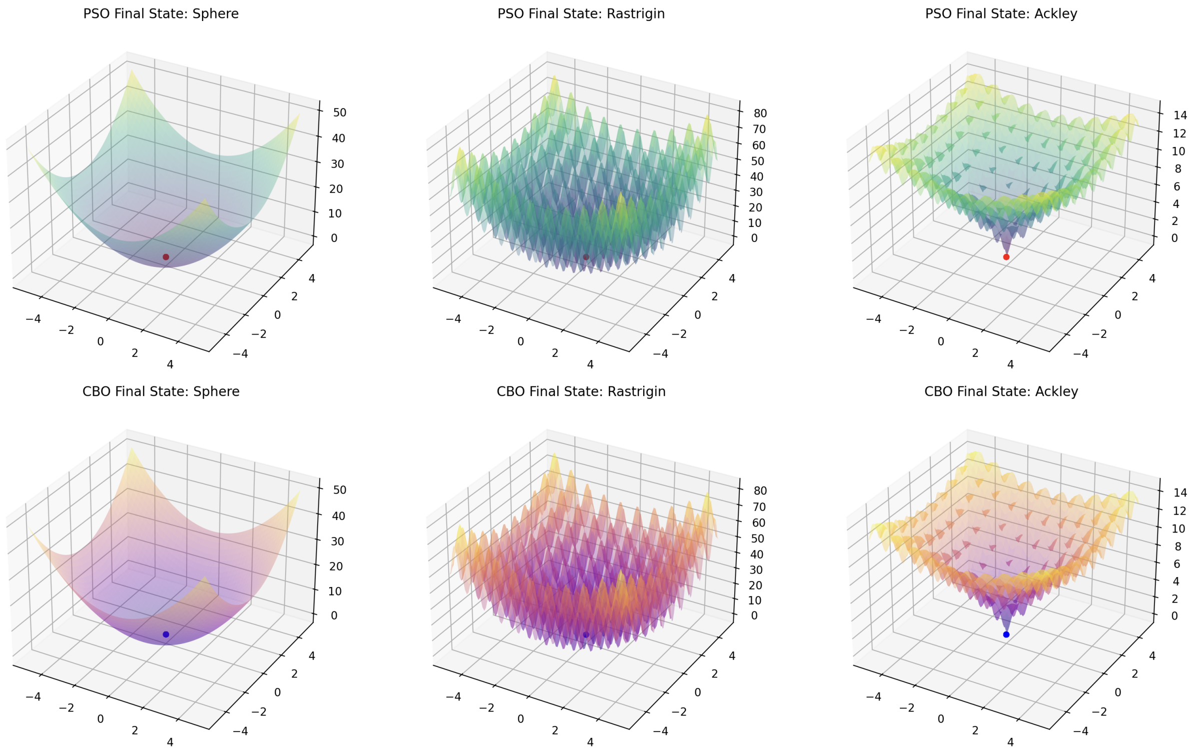

Figure 2: Final Population Distribution at Convergence

6 Comparative Analysis

The experimental results demonstrate that both Particle Swarm Optimization (PSO) and Consensus-Based Optimization (CBO) are highly effective at navigating complex, non-convex landscapes. A critical observation in this study is the high numerical precision achieved by the PSO algorithm following the correct application of the inertia weight ω to the momentum term.

On the **Sphere** and **Ackley** functions, PSO achieved near-analytical precision (1.469×10^{-21} and 7.987×10^{-10} respectively), outperforming CBO in terms of final residual error. Most notably, on the **Rastrigin** function—which is characterized by a high density of local minima—PSO successfully located the absolute global minimum (0.000×10^{0}).

While PSO provided higher precision, **CBO demonstrated superior efficiency** in terms of the computational budget. In every test case, CBO reached convergence with approximately 20-30% fewer function evaluations than PSO. This efficiency stems from the "collective" nature of the consensus moment m^{α}, which allows the entire population to move as a single unit toward the weighted baricenter, whereas PSO particles maintain individual trajectories until the swarm fully settles.

7 Conclusion

This project successfully implemented and validated two gradient-free optimization frameworks. The results indicate a clear trade-off between **numerical precision** and **computational efficiency**.

PSO proved to be the more precise optimizer for these benchmark functions, consistently finding deeper minima with near-zero error. However, CBO remains the preferred choice for scenarios where the cost function $f(x)$ is computationally expensive to evaluate, as it consistently required a lower evaluation budget to find the vicinity of the global optimum. Both algorithms successfully avoided the "local minima traps" that typically hinder standard gradient-based methods, demonstrating the power of bio-inspired stochastic exploration in global optimization.

References

[1] Wikipedia contributors. *Test functions for optimization*. Available at: https://en.wikipedia.org/wiki/Test_functions_for_optimization (Accessed Jan 6, 2026)

[2] Gemini (Google). *Assistance in Algorithm Implementation, LaTeX Formatting, and Data Visualization*.

[3] Tytus Felbor. *Advanced Maths & Stats: Gradient-Free Optimization Notes*. Class Lecture Notes, 2025.