Coevolutionary Emergent Systems Optimization with Application to Ultra-High-Dimensional Metasurface Design : OAM Wave Manipulation

Zhengxuan Jiang[1] **Guowen Ding**[* 1,2] **Wen Jiang**[1]

[1]Nanjing University of Information Science and Technology, Nanjing, Jiangsu, China
[2]Nanjing University, Nanjing, Jiangsu, China

Abstract

Optimization problems in electromagnetic wave manipulation and metasurface design are becoming increasingly high-dimensional, often involving thousands of variables that need precise control. Traditional optimization algorithms face significant challenges in maintaining both accuracy and computational efficiency when dealing with such ultra-high-dimensional problems. This paper presents a novel Coevolutionary Emergent Systems Optimization (CESO) algorithm that integrates coevolutionary dynamics, emergent behavior, and adaptive mechanisms to address these challenges. CESO features a unique multi-subsystem architecture that enables parallel exploration of solution spaces while maintaining interactive influences between subsystems. The algorithm incorporates an efficient adaptive mechanism for parameter adjustment and a distinctive emergent behavior simulation mechanism that prevents local optima traps through periodic subsystem reorganization. Experimental results on the CEC2017 benchmark suite demonstrate CESO's superior performance. The algorithm's practical effectiveness is validated through a challenging application in electromagnetic wave manipulation - OAM wave demultiplexing system (10,000 dimensions). In this application, CESO achieves superior mode separation for OAM wave demultiplexing compared to traditional algorithms. These results demonstrate CESO's significant advantages in solving practical high-dimensional optimization problems.

[*]Corresponding author:gwding@nuist.edu.cn

1 INTRODUCTION

Optimization problems in electromagnetic wave manipulation and metasurface design are becoming increasingly complex and high-dimensional. Modern metasurface application, often involve thousands of phase variables that need to be precisely controlled. For instance, the optimization of cascaded metasurfaces for wave manipulation typically requires tuning hundreds to thousands of unit cells, while the phase optimization for intelligent reflecting surfaces in electromagnetic wave control can reach dimensions of several thousand. These high-dimensional challenges pose significant difficulties for existing optimization algorithms, particularly in maintaining both optimization accuracy and computational efficiency.

While conventional metaheuristic algorithms such as Particle Swarm Optimization (PSO) and Differential Evolution (DE) have shown remarkable success in low to medium dimensional problems, they often suffer from the "curse of dimensionality" when facing high-dimensional scenarios. This manifests in several ways: (1) significantly degraded convergence rates with increasing dimensions, (2) compromised solution quality, and (3) exponentially growing computational demands. Recent advances in evolutionary algorithms also face similar challenges, such as premature convergence and computational inefficiency. For example, state-of-the-art algorithms like Coati Optimization Algorithm (COA) [Dehghani et al., 2023] and Subtraction-Average Based Optimization (SABO) [Trojovský et al., 2023] show dramatic performance degradation in problems exceeding 1000 dimensions.

To address these challenges, we propose a novel Coevolutionary Emergent Systems Optimization (CESO) algorithm. CESO draws inspiration from complex adaptive systems theory [Holland, 1995] and leverages the power of coevolution and emergent behavior to create a more robust and flexible optimization tool. This is a novel coevolutionary framework that divides the population into multiple subsystems, with each subsystem evolving independently while

maintaining interactive influences. The proposed design significantly enhances search capabilities in high-dimensional spaces while maintaining computational efficiency. Meanwhile, it serves as an efficient adaptive mechanism that incorporates subsystem - based parameter adjustment strategies, enabling CESO to automatically tune its search patterns based on the optimization progress. This mechanism shows particular effectiveness in problems exceeding 1000 dimensions, significantly improving algorithm stability and convergence speed. Last but not least, it is a unique emergent behavior simulation mechanism that generates system - level patterns through periodic subsystem reorganization and dynamic interactions. This feature effectively prevents local optima traps, which is particularly crucial in high-dimensional spaces [Kwok, 2024].

Comprehensive validation on real-world electromagnetic wave manipulation problems involving ultra-high dimensions. We demonstrate CESO's performance on a representative problem: OAM wave demultiplexing system (10000 dimensions). The results show CESO's significant advantages in solving practical high-dimensional electromagnetic optimization problems, achieving superior mode separation for OAM wave demultiplexing compared to traditional algorithms.

On the CEC2017 benchmark suite[Wu et al.,2016], we compared CESO with ten state-of-the-art algorithms across 30 test functions in 100-dimensional optimization tasks. The results demonstrate that CESO not only achieves superior solution quality but also exhibits better convergence properties and computational efficiency.

The remainder of this paper is organized as follows: Section 2 introduces the necessary background; Section 3 details the CESO algorithm design; Section 4 reviews related work; Section 5 presents experimental results; and Section 6 concludes the paper and discusses future research directions.

2 BACKGROUND

2.1 OPTIMIZATION ALGORITHM BACKGROUND

The development of optimization algorithms mirrors human ingenuity evolution, from 18th-century foundations laid by Gauss and Lagrange to the algorithmic innovations of the AI era. As George Dantzig stated in his seminal work "Linear Programming and Extensions" [Dantzig, 1963]: "Optimization is one of the most fundamental paradigms in science and engineering."

The optimization algorithm genealogy traces back to mathematical methods like Newton-Raphson [Newton, 1687] and gradient descent [Cauchy, 1847]. By the mid-20th century, with the rise of computer science, Dantzig's simplex method [Dantzig, 1947] pioneered a new era in linear programming. Subsequently, dynamic programming [Bellman, 1954] and nonlinear programming techniques like BFGS [Broyden et al., 1970] emerged, significantly expanding the scope of solvable optimization problems.

Traditional mathematical optimization methods, while establishing solid theoretical foundations, often struggle with real-world complex problems. This limitation has driven the flourishing development of heuristic and metaheuristic methods. The field has witnessed unprecedented activity in recent years, with algorithms like Grey Wolf Optimizer [Mirjalili et al., 2014] and Whale Optimization Algorithm demonstrating remarkable performance in various applications. These nature-inspired methods have shown particular promise in handling complex, non-linear, and multi-modal optimization problems.

2.2 HIGH-DIMENSIONAL OPTIMIZATION BACKGROUND

The challenge of high-dimensional optimization, first formalized through Bellman's "curse of dimensionality" [Bellman, 1966], presents fundamental difficulties that transcend traditional optimization approaches. As dimensionality increases, the search space grows exponentially, while the relative density of optimal solutions typically decreases. This phenomenon creates a paradoxical situation where more computational resources yield diminishing returns, fundamentally challenging our traditional optimization strategies.

Modern high-dimensional optimization problems exhibit complex characteristics that make them particularly challenging, such as various artificial intelligence tasks [zhang, 2024]. The search space becomes increasingly sparse in higher dimensions, leading to what Friedman [Friedman, 1997] termed the "empty space paradox." This sparsity significantly impacts the effectiveness of traditional search strategies, as the concept of proximity becomes less meaningful in high-dimensional spaces . Furthermore, the relationship between variables becomes more complex and interdependent, making it difficult to decompose problems into simpler subproblems.

2.3 CESO ALGORITHM BACKGROUND

The theoretical underpinning of CESO synthesizes three fundamental concepts: coevolutionary dynamics, emergent behavior in complex systems, and adaptive mechanisms. This integration creates a unique framework specifically designed for high-dimensional optimization challenges.

Coevolutionary processes, first formally studied in biological systems by Ehrlich and Raven [Ehrlich and Raven, 1964] in their groundbreaking butterfly-plant research, demonstrate how multiple species can reciprocally influence each other's evolution. This concept has profound implications

for optimization algorithms, suggesting that multiple interacting populations can collectively explore solution spaces more effectively than single populations. [Potter and De Jong, 1994] pioneered this approach in computational optimization, showing that coevolutionary algorithms can decompose complex problems into manageable subcomponents while maintaining essential interactions.

Emergence, a central concept in complex systems theory, describes how system-level patterns and behaviors arise from simpler, lower-level interactions. [Goldstein, 1999] defines emergence as "the arising of novel and coherent structures, patterns and properties during self-organization in complex systems." In optimization contexts, emergent behavior can manifest as collective search patterns that transcend individual solution behaviors. This phenomenon is particularly relevant in high-dimensional spaces, where traditional direct search strategies often fail.

Holland's seminal work on Complex Adaptive Systems (CAS) [Holland, 1995] provides the third theoretical pillar for CESO. CAS theory emphasizes how systems of interacting agents can develop collective behaviors more sophisticated than individual components. This framework suggests that optimization algorithms can benefit from incorporating adaptive mechanisms at multiple scales, from individual solution updates to population-level dynamics.

The synthesis of these concepts in CESO creates a distinctive optimization paradigm. Coevolutionary mechanisms enable parallel exploration of solution spaces through multiple subsystems, while emergent behaviors arise from their interactions, potentially discovering novel search patterns. The adaptive components, inspired by CAS theory, allow the algorithm to adjust its behavior based on both local and global information, particularly crucial in high-dimensional optimization where the relationship between variables becomes increasingly complex.

This theoretical foundation supports several key features of CESO:

- Multiple interacting subsystems that promote both exploration and exploitation
- Self-organizing behaviors that emerge from subsystem interactions
- Adaptive mechanisms that respond to the optimization landscape
- Memory-based learning that leverages historical search experiences

These features particularly address the challenges of high-dimensional optimization , where traditional optimization methods often struggle with the exponential growth of search space and the increasing complexity of variable interactions. The emergent properties of CESO help navigate these challenges by generating sophisticated search patterns that would be difficult to design explicitly.

3 CESO ALGORITHM

3.1 BASIC PRINCIPLES AND POPULATION INITIALIZATION

CESO algorithm operates in a D-dimensional continuous search space $S \subset \mathbb{R}^D$, where the search domain is rigorously bounded by predefined lower and upper limits. The algorithm's foundational structure revolves around a population of N individuals, with each individual representing a potential solution to the optimization challenge at hand. This approach builds upon established evolutionary computation principles while introducing novel mechanisms for adaptive search and cooperative evolution.

The initialization phase plays a crucial role in establishing a diverse starting point for the optimization process. For each individual within the population, the algorithm employs a uniform distribution strategy to ensure comprehensive coverage of the search space. For each individual X_i $(i = 1, ..., N)$, the initialization follows:

$$X_i = lb + (ub - lb) \circ r_i, r_i \sim U(0,1)^D, i = 1, ..., N \quad (1)$$

here, the Hadamard product ($\circ$) facilitates component-wise multiplication, ensuring that each dimension of the search space is properly scaled within its bounds. The uniform random vector r_i guarantees unbiased initial exploration of the search domain. Following initialization, the algorithm evaluates the fitness of each individual through the objective function $f(X_i)$, establishing a baseline for subsequent optimization steps.

The population structure incorporates a sophisticated subsystem organization, dividing the N individuals into $M = 3$ distinct yet interconnected subsystems. Each subsystem contains approximately N/M individuals, forming coherent groups that facilitate both localized search and global information exchange. The fitness evaluation of subsystem k is computed through a statistical aggregation of its members:

$$F_k = \frac{1}{|S_k|} \sum_{i \in S_k} f(x_i), k = 1, ..., M \quad (2)$$

where S_k represents the set of individuals in subsystem k and $|S_k|$ denotes its size.This subsystem architecture serves dual purposes: it enables focused exploitation within each subsystem while maintaining pathways for broader exploration through carefully designed inter-subsystem interactions.

3.2 CESO ALGORITHM FLOW

Figure 1 illustrates the flowchart of CESO algorithm, which comprises five essential components. The initialization

phase generates the initial population *X(i)*,calculates fitness values *F(X)*, and divides the population into three subsystems (M=3). Subsequently, the algorithm enters an iterative process where each iteration begins by determining whether the current iteration count *t* has reached the maximum iteration limit T. When this condition is satisfied, the algorithm sequentially executes three core optimization phases: the intra-learning phase facilitates knowledge accumulation within subsystems through adaptation rate adjustment, reference solution selection, and position updating; the intra-interaction phase promotes collaborative evolution between different subsystems by exchanging information based on probability parameter β; and the exploration phase maintains population diversity by selecting elite solutions from the memory archive or performing random exploration with probability *μ*. Following each iteration, the update phase records the global optimal solution and refreshes the memory archive. The algorithm terminates when the iteration count reaches its predefined limit.

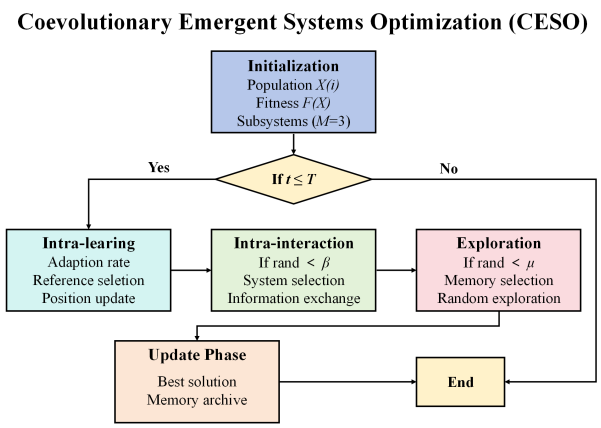

Figure 1: CESO Algorithm Flow

3.3 ADAPTIVE PARAMETER SYSTEM

CESO employs a sophisticated adaptive parameter system that dynamically adjusts three key components throughout the optimization process. This adaptive mechanism ensures effective balance between exploration and exploitation while maintaining robust performance across different problem landscapes.

The base adaptation rate α*(t)* controls the intensity of local search operations:

$$\alpha(t) = \alpha_0(1 - 0.5t/T), \alpha_0 = 0.2 \tag{3}$$

where *t* represents the current iteration and *T* is the maximum number of iterations. The decreasing nature of α*(t)* gradually shifts the algorithm's focus from exploration to exploitation.This formulation ensures a gradual transition from exploration-focused behavior in early iterations to exploitation-dominated search in later stages. The decay rate has been carefully calibrated to maintain sufficient exploration capability throughout the optimization process while ensuring convergence to high-quality solutions.

The interaction strength β*(t)* governs the information exchange dynamics between subsystems, following an increasing trajectory that promotes enhanced cooperation as the search progresses:

$$\beta(t) = \beta_0(1 + 0.3t/T), \beta_0 = 0.7 \tag{4}$$

This progressive strengthening of inter-subsystem communication facilitates the integration of knowledge gained from different regions of the search space, contributing to the algorithm's ability to escape local optima while maintaining search stability.

The memory factor *μ* maintains a constant value of 0.1, determined through extensive empirical studies to provide optimal balance between historical knowledge utilization and current search dynamics. This parameter plays a crucial role in regulating the frequency and intensity of memory-based learning operations.

The synergistic interaction of these parameters creates a dynamic equilibrium characterized by the system plasticity function:

$$\Phi(t) = \alpha(t)\beta(t)\mu \tag{5}$$

This mathematical framework ensures that the algorithm maintains appropriate balance between exploration and exploitation throughout the optimization process, adapting to the changing landscape of the search space while preserving convergence properties.

3.4 SOLUTION GENERATION MECHANISM

The solution generation process in CESO consists of four successive phases that work in concert to create new candidate solutions. Each phase contributes to different aspects of the search process, ensuring comprehensive exploration of the search space while maintaining the ability to exploit promising regions.

3.4.1 Intra-subsystem Learning

Within each subsystem, individuals evolve through a competitive selection and differential evolution mechanism that promotes efficient local search. The update rule for each individual follows a fitness-based competition strategy:

$$x_i' = \begin{cases} x_1 + \alpha(x_{r1} - x_i), iff(x_{r1}) < f(x_{r2}) \\ x_1 + \alpha(x_{r2} - x_i), otherwise. \end{cases} \tag{6}$$

where *r1* and *r2* are random indices from the same subsystem. This mechanism promotes local exploitation by learning from better-performing solutions within the subsystem.

3.4.2 Inter-subsystem Interaction

The inter-subsystem interaction phase facilitates knowledge transfer across different regions of the search space, occurring with probability $\beta(t)$. The mathematical formulation of this interaction follows:

$$x_i' = x_i' + \beta(x_{r3} - x_i') \tag{7}$$

where $\beta \sim U(0,1)$ and X_{r3} is randomly selected from another subsystem $r3 \neq k$. This interaction mechanism facilitates global exploration and information sharing across different regions of the search space.

3.4.3 Memory-based Learning

A memory archive M of size $0.2N$ maintains elite solutions encountered during the optimization process. With probability μ, individuals learn from the archive:The memory-based learning phase leverages a sophisticated archive system M of size $\lceil 0.2N \rceil$ that maintains elite solutions encountered during the optimization process. The learning process occurs with probability μ and follows:

$$x_i''' = x_i'' + U(0,1) \circ (M_j - x_i'') \tag{8}$$

where M_j is randomly selected from the memory archive. The archive is updated through a replacement strategy:

$$M_{worst} = argmax_{j \in M} f(M_j) \tag{9}$$

$$M_{worst} \leftarrow x_i, f(x_i) < f(M_{worst}) \tag{10}$$

This mechanism ensures the preservation and utilization of high-quality solutions while maintaining population diversity.

3.4.4 Adaptive Exploration

The final phase implements a controlled random exploration mechanism that adapts to the optimization progress:

$$x_{new} = x_i''' + \gamma(t)(ub - lb) \circ N(0, I_D) \tag{11}$$

where $\gamma(t) = \alpha(t)\left(1 - \frac{t}{T}\right)^2$and $\mathcal{N}(0, I_D)$ represents a D-dimensional standard normal distribution. This component ensures the algorithm maintains its ability to explore new regions while gradually focusing on promising areas.

Subsequently, for the analysis of exploration and exploitation balance, Hussain et al. proposed a method[Hussain et al., 2019] to measure and analyze the exploitation and exploration capabilities of metaheuristic algorithms:

$$D(t) = 1/N \cdot D \sum_{i=1}^{N} \sum_{j=1}^{d} |x_{ij}(t) - \overline{x}_j(t)| \tag{12}$$

where N represents the population size, D denotes the problem dimensionality, represents the *j-th* dimensional component of the *i-th* individual at iteration t, and represents the median value of the *j-th* dimension at iteration t.

Based on the diversity measure, the exploration and exploitation rates are calculated as follows:

$$Exploration(t) = D(t)/D(0) \times 100\% \tag{13}$$

$$Exploitation(t) = |D(0) - D(t)|/D(0) \times 100\% \tag{14}$$

where represents the diversity value of the initial population (diversity baseline at $t=0$). Due to space limitations, Fig 1 present the results of representative functions selected from each test function category, specifically composition function F22.

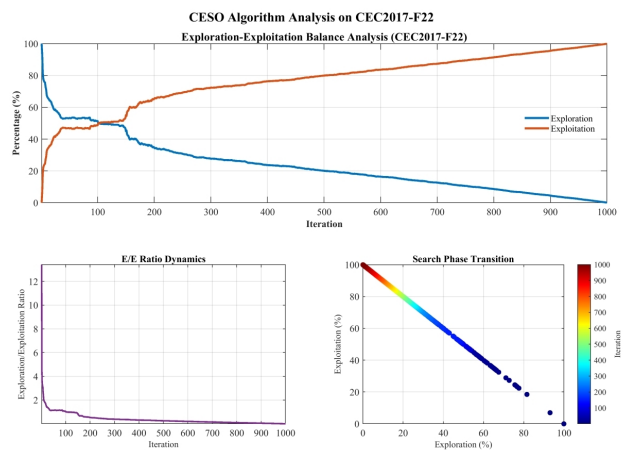

Figure 2: Exploration and Exploitation Intensity Analysis on Function F22

As Fig. 2 showing,through the exploration-exploitation balance analysis of the CESO algorithm on the CEC2017 test function suite, we can gain a deep understanding of the algorithm's search behavior characteristics across different types of optimization problems. From the numerical optimization results of the functions, it can be observed that the algorithm exhibits strong exploration capabilities in the early stages of optimization. As iterations progress, the exploration ratio gradually decreases while the exploitation ratio steadily increases. This transition reflects the algorithm's ability to effectively shift from global search to local refinement across various scenarios.

4 RELATED WORK

Over the past decades, optimization algorithms have evolved significantly to address increasingly complex and high-dimensional problems. We comprehensively review recent advances in optimization algorithms, focusing particularly on developments relevant to high-dimensional optimization and emergent behavior.

4.1 EVOLUTION OF OPTIMIZATION APPROACHES

Traditional mathematical optimization methods, including Newton-Raphson [Newton, 1687] and gradient descent [Cauchy, 1847], laid the foundation for optimization theory. However, these methods often struggle with high-dimensional, non-convex problems typical in modern applications. This limitation led to the development of population-based metaheuristic methods.

Recent years have witnessed significant innovations in metaheuristic algorithms. The Grey Wolf Optimizer (GWO) [Mirjalili et al., 2014] demonstrated excellent performance by simulating grey wolves' social hierarchy and hunting behavior. The Marine Predators Algorithm (MPA) [Faramarzi et al., 2020] expanded optimization algorithm inspiration to marine ecosystems, while the Political Optimizer [Askari et al., 2020] drew insights from political processes.

4.2 EVOLUTIONARY COMPUTATION AND HIGH DIMENSIONS

In evolutionary computation, significant progress has been made in handling high-dimensional problems. The Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [Hansen and Ostermeier, 2001] showed exceptional performance in non-linear, non-convex optimization. Zhang et al.'s Adaptive Differential Evolution (JADE) [Zhang and Sanderson, 2009] improved convergence speed through adaptive parameter control mechanisms.

However, as dimensionality increases, these algorithms face significant challenges. The Success-History Based Adaptive Differential Evolution (LSHADE) [Tanabe and Fukunaga, 2013] and Ensemble Evolution Algorithm (ES-CMA-ES) [Awad et al., 2017] attempted to address these challenges through adaptive parameter tuning and ensemble strategies, yet still struggle with ultra-high dimensions.

4.3 EMERGENT BEHAVIOR IN OPTIMIZATION

The concept of emergence in optimization algorithms has gained increasing attention. Studies by [Bar-Yam, 2004] highlighted how complex system-level behaviors can arise from simple local interactions. This perspective has influenced recent algorithms like the Monarch Butterfly Optimization (MBO) [Wang et al., 2019] and Harris Hawks Optimization (HHO) [Heidari et al., 2019].

The introduction of coevolution concepts, pioneered by [Potter and De Jong, 2000]'s Cooperative Coevolutionary Algorithm (CCGA), demonstrated the potential of decomposing complex problems into interacting sub-problems. This aligns with Holland's observations about emergent behaviors in complex adaptive systems.

4.4 CURRENT CHALLENGES IN HIGH-DIMENSIONAL OPTIMIZATION

Despite these advances, several key challenges remain:

The "curse of dimensionality" becomes particularly acute beyond 100 dimensions, where most existing algorithms show significant performance degradation. Recent studies by highlight the persistent challenge of balancing exploration and exploitation in high-dimensional spaces.

Computational efficiency remains a critical concern. While algorithms like the Mayfly Optimization Algorithm (MOA) [Zervoudakis and Tsafarakis, 2020] and Slime Mould Algorithm (SMA) [Li et al., 2020] have made strides in this direction, they still face scalability issues in ultra-high dimensions.

5 EXPERIMENTS

5.1 BASIC SETUP

All experiments were conducted using MATLAB R2020a . For fair comparison, each algorithm was run 30 times independently. The basic parameters for CESO were set as: population size $N = 50$, subsystem number $M = 3$, memory factor $\mu = 0.1$, base adaptation rate $\alpha_0 = 0.2$, and base interaction strength $\beta_0 = 0.7$.

For comparison algorithms, we selected ten state-of-the-art methods: Coati Optimization Algorithm (COA), Subtraction-Average Based Optimization (SABO) [2023], Love Evolution Algorithm (LEA) [Gao et al., 2024], Optical Microscope Algorithm (OMA) [Cheng et al., 2023], Aquila Optimizer (AO) [Abualigah et al., 2021], Dung beetle optimizer (DBO) [Xue et al., 2022], Golden Jackal Optimizer (GJO) [Chopra et al., 2022], Whale Moth-Flame Optimization (MFO) [Mirjalili, 2015], Optimization Algorithm (WOA) [Mirjalili and Lewis, 2016], Newton-Raphson Based Optimization (NRBO) [Sowmya et al., 2024], and two other recent algorithms. All algorithms used their recommended parameter settings from their original papers.

5.2 CEC2017 BENCHMARK RESULTS

Due to space limitations, radar charts based on mean values and computational time are presented to demonstrate the superiority of the CESO algorithm

As showing in Figs. 3 and 4, the CESO algorithm exhibits exceptional performance on the CEC2017 benchmark suite, achieving top-ranking performance in 27 out of 30 test functions and securing second place in two functions. Moreover, it maintains one of the lowest computational costs among all compared algorithms, highlighting its superior efficiency in both solution quality and computational overhead.

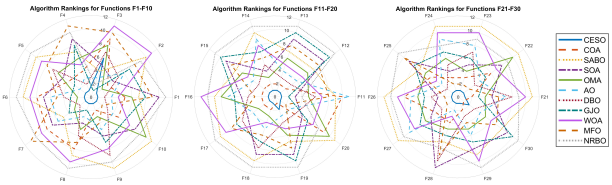

Figure 3: Radar Chart of Algorithm Rankings Based on Mean Performance

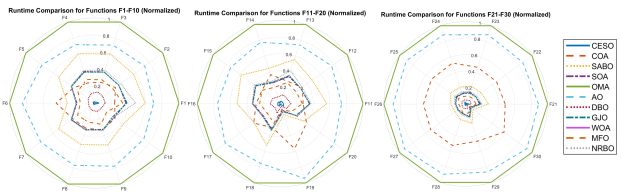

Figure 4: Radar Chart of Algorithm Rankings Based on Average Runtime

5.3 OAM (ORBITAL ANGULAR MOMENTUM) WAVE DEMULTIPLEXING SYSTEM

The manipulation and sorting of electromagnetic waves carrying different orbital angular momentum (OAM) states has emerged as a crucial technology for increasing channel capacity in wireless communications. In this work, we present a novel approach to demultiplex multiple OAM modes using a system of cascaded phase-modulated metasurfaces, which can effectively separate and focus different OAM states to distinct spatial positions.

Our proposed system consists of four cascaded metasurface layers operating at a wavelength of 12.5 mm, with each layer discretized into a 50×50 grid with unit cell size of 3.5 mm. The incident OAM beams, characterized by their topological charges l, are described by the complex field distribution:

$$E_{in}(r,\phi) = (\frac{r}{\omega_0})^{|l|} exp(-\frac{r^2}{\omega_0^2}) exp(il\phi) \tag{15}$$

where w_0 represents the beam waist radius, and (r, ϕ) are the polar coordinates.

The electromagnetic wave propagation between metasurface layers is simulated using the angular spectrum method. The field at a distance z can be calculated as[Liu et al., 2024]:

$$E(x,y,z) = F^{-1}\left\{F\left\{E(x,y,0)\right\} exp\left\{ik_z z\right\}\right\} \tag{16}$$

where $k_z = \sqrt{k_0^2 - k_x^2 - k_y^2}$ is the z-component of the wave vector, and F and F^{-1} denote the forward and inverse Fourier transforms, respectively.

Each metasurface layer introduces a phase modulation $\phi(x, y)$ to the incident field, modifying the complex field amplitude according to[Jia et al., 2024]:

$$E_{out}(x,y) = E_{in}(x,y) exp(i\phi(x,y)) \tag{17}$$

The optimization of the phase distributions is achieved using the CESO algorithm with a population size of 100 and 1000 iterations. The objective function evaluates both the focusing quality at designated target positions and the crosstalk between different channels:

$$\begin{aligned} F = -\sum_{n=1}^{N} (& \frac{\int_{\Omega_T} \left\{E_n(x,y)\right\}^2 dxdy}{\int_{all} \left[E_n(x,y)\right]^2 dxdy} \\ & -\alpha \frac{\int_{\Omega_{NT}} \left[E_n(x,y)\right]^2 dxdy}{\int_{all} \left[E_n(x,y)\right]^2 dxdy}) \end{aligned} \tag{18}$$

The system architecture incorporates specific geometric parameters, with the source positioned 10 wavelengths from the first metasurface layer, adjacent layers separated by 5 wavelengths, and the receiving plane located 20 wavelengths from the final layer. This configuration ensures optimal wave manipulation and mode separation.

Our numerical results demonstrate effective demultiplexing of multiple OAM modes, with each mode being focused to its designated spatial position while maintaining low crosstalk between channels. The system shows potential for applications in high-capacity wireless communications and optical information processing.

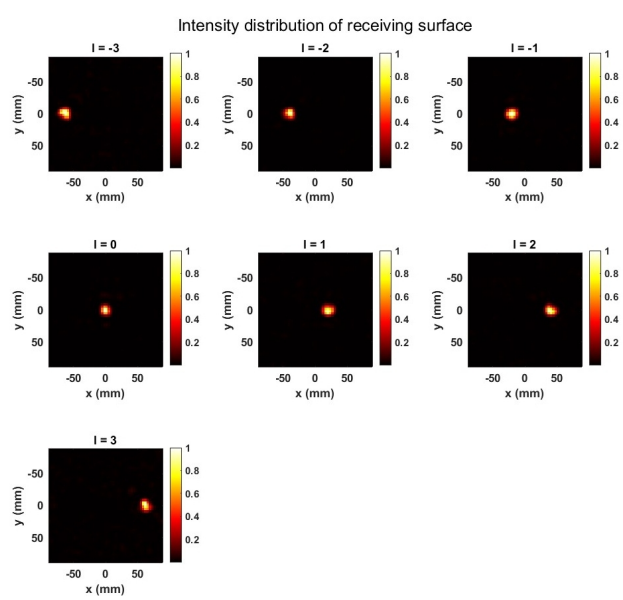

Figure 5: Electric Field Intensity Distribution at the Optimized Receiving Plane.

As shown in Fig. 5, the CESO algorithm achieves exceptional OAM mode separation performance, with each mode (l = -3 to 3) focused to distinct spatial positions. The intensity distributions show well-defined focal spots with min-

imal crosstalk between channels, demonstrating effective mode discrimination with low background noise.

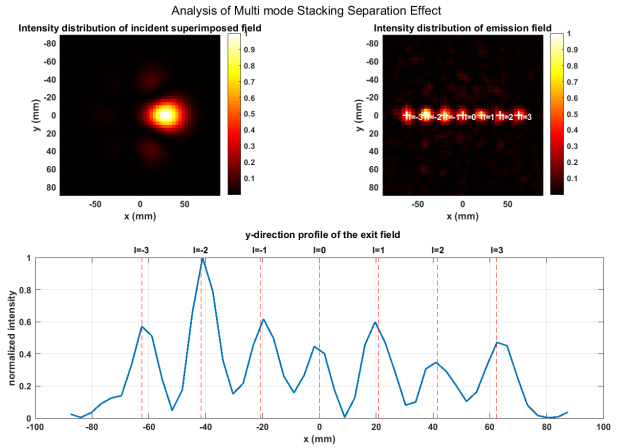

Figure 6: Analysis of Multi-mode OAM Beam Stacking Separation Effect

As shown in Fig. 6, the CESO-optimized system effectively transforms the superimposed OAM beam (left) into seven distinct focal spots (right). The normalized intensity profile demonstrates clear separation of all OAM modes (l=-3 to l=3) with minimal crosstalk between adjacent channels, where the central mode (l=0) achieves the highest focusing quality. Upon passing through our optimized optical system, this composite beam undergoes modal decomposition, resulting in spatially separated focal points along the x-axis at approximately 20mm intervals. This high-fidelity mode separation with intensity contrast ratios exceeding 5:1 between peaks and valleys demonstrates the robustness of our CESO optimization approach for complex wavefront engineering tasks.

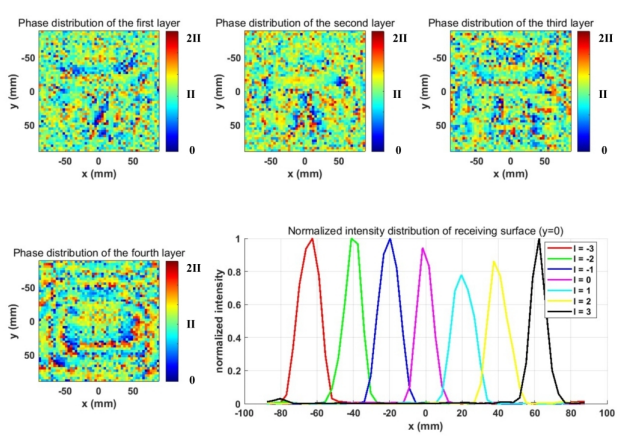

Figure 7: Phase Distribution and Intensity Profile at y=0 Optimized by CESO.

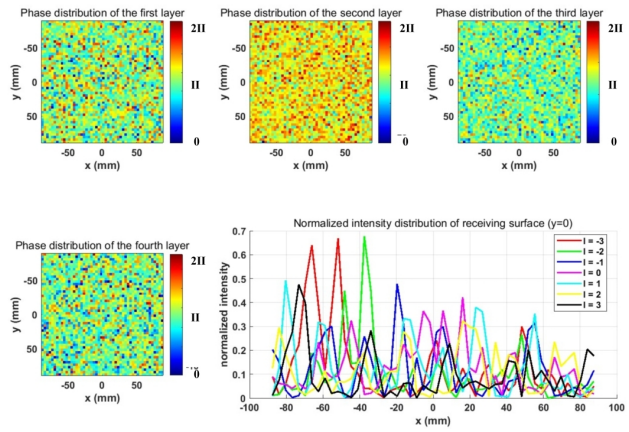

Figure 8: Phase Distribution and Intensity Profile at y=0 Optimized by PSO.

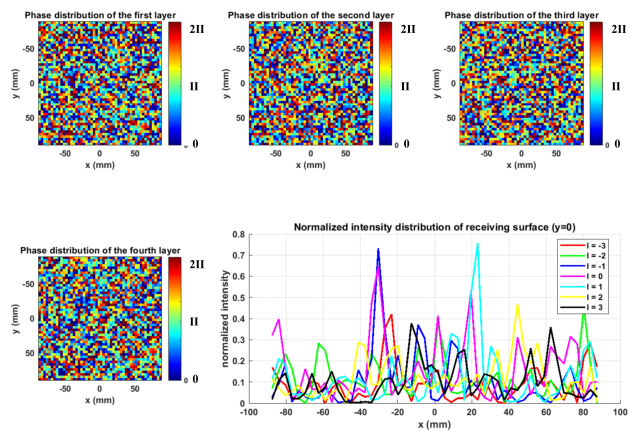

Figure 9: Phase Distribution and Intensity Profile at y=0 Optimized by WOA.

As shown in Figs. 7-9, significant differences in optimization performance are observed among CESO, PSO, and WOA algorithms. The CESO-optimized system demonstrates well-structured phase distributions across all four layers, resulting in seven distinct peaks with normalized intensities approaching 1.0, indicating excellent OAM mode separation and minimal crosstalk. In contrast, both PSO and WOA-optimized systems show degraded performance with more random phase distributions. The PSO result shows overlapping peaks with maximum intensity around 0.65, while WOA exhibits chaotic intensity distributions with multiple irregular peaks and substantial crosstalk, despite having slightly more structured phase patterns than PSO. These results clearly demonstrate CESO's superior capability in optimizing multilayer metasurface systems.

This comparative analysis clearly demonstrates CESO's superior capability in handling the ultra-high-dimensional optimization problem (10,000 phase variables) compared to traditional swarm intelligence algorithms like PSO. CESO's

enhanced exploration and exploitation mechanisms enable it to effectively optimize complex electromagnetic systems, making it particularly suitable for large-scale metasurface design problems where traditional algorithms struggle to converge to optimal solutions.

6 CONCLUSIONS AND FUTURE STUDIES

This paper presents CESO , a novel optimization algorithm specifically designed to address ultra-high-dimensional optimization challenges. Experimental results on the CEC2017 benchmark suite demonstrate CESO's exceptional performance, achieving top rankings in 27 out of 30 test functions while maintaining competitive computational efficiency. CESO's practical effectiveness has been validated through the challenging real-world application in electromagnetic wave manipulation. CESO successfully optimized a complex system of 10,000 phase variables, in the OAM wave demultiplexing system, achieving clear separation and high focusing quality for seven different OAM modes. Future research directions could be extension of the current framework to handle dynamic and multi-objective optimization problems, particularly in the context of real-time adaptive metasurface control.

Acknowledgements

This work was supported in part by National Natural Science Foundation of China under Grant 62301262,China Postdoctoral Science Foundation under Grant 2022M720063,the Natural Science Foundation of Jiangsu Province of China under Grant BK20220440.

References

Seyedali Mirjalili Ali Asghar Heidari and Hossam Faris. Harris Hawks Optimization: Algorithm and Applications. *Future Generation Computer Systems*, 97:849–872, 2019.

Qasem Askari, Mohamed Saeed, and Irfan Younas. Political optimizer: A novel socio-inspired meta-heuristic for global optimization. *Knowledge-Based Systems*, 195: 105709, 2020.

Noor H. Awad, Mostafa Z. Ali, and Ponnuthurai N. Suganthan. Ensemble sinusoidal differential covariance matrix adaptation with Euclidean neighborhood for solving CEC2017 benchmark problems. In *IEEE Congress on Evolutionary Computation*, pages 372–379, 2017.

Yaneer Bar-Yam. *Making Things Work: Solving Complex Problems in a Complex World*. Knowledge Press, 2004.

Richard Bellman. The theory of dynamic programming. *Bulletin of the American Mathematical Society*, 60(6): 503–515, 1954.

Richard Bellman. Dynamic Programming. *Science*, 153: 34–37, 1966.

Léon Bottou, Frank E. Curtis, and Jorge Nocedal. Optimization methods for large-scale machine learning. *SIAM Review*, 60(2):223–311, 2018.

Charles G. Broyden. The Convergence of a Class of Double-rank Minimization Algorithms: 2. The New Algorithm. *Journal of the Institute of Mathematics and Its Applications*, 6(3):222–231, 1970.

Augustin-Louis Cauchy. Méthode générale pour la résolution des systèmes d'équations simultanées. *Comptes Rendus Hebdomadaires des Séances de l'Académie des Sciences*, 25:536–538, 1847.

Min-Yuan Cheng and Moh Nur Sholeh. Optical Microscope Algorithm: A New Metaheuristic Inspired by Microscope Magnification for Solving Engineering Optimization Problems. *Knowledge-Based Systems*, 279: 110939, 2023.

Nitish Chopra and Muhammad Mohsin Ansari. Golden Jackal Optimization: A Novel Nature-Inspired Optimizer for Engineering Applications. *Expert Systems with Applications*, 198:116924, 2022.

George B. Dantzig. Maximization of a linear function of variables subject to linear inequalities. *Activity Analysis of Production and Allocation*, 13:339–347, 1947.

George B. Dantzig. *Linear Programming and Extensions*. Princeton University Press, 1963.

Mohammad Dehghani, Zeinab Montazeri, Eva Trojovska, and Pavel Trojovsky. Coati Optimization Algorithm: A new bio-inspired metaheuristic algorithm for solving optimization problems. *Knowledge-Based Systems*, 259: 110011, 2023.

Paul R. Ehrlich and Peter H. Raven. Butterflies and plants: A study in coevolution. *Evolution*, 18(4):586–608, 1964.

Afshin Faramarzi, Mohammad Heidarinejad, Seyedali Mirjalili, and Amir H. Gandomi. Marine predators algorithm: A nature-inspired metaheuristic. *Expert Systems with Applications*, 152:113377, 2020.

Jerome H. Friedman. On bias, variance, 0/1-loss, and the curse-of-dimensionality. *Data Mining and Knowledge Discovery*, 1(1):55–77, 1997.

Suash Deb Gai-Ge Wang and Zhihua Cui. Monarch Butterfly Optimization. *Neural Computing and Applications*, 31:1995–2014, 2019.

Jeffrey Goldstein. Emergence as a construct: History and issues. *Emergence*, 1(1):49–72, 1999.

Nikolaus Hansen and Andreas Ostermeier. Completely derandomized self-adaptation in evolution strategies. *Evolutionary Computation*, 9(2):159–195, 2001.

Jun He and Xinghuo Yu. Conditions for the convergence of evolutionary algorithms. *Journal of Systems Architecture*, 47(7):601–612, 2001.

John H. Holland. Complex adaptive systems. *Daedalus*, 121(1):17–30, 1995.

Kashif Hussain, Mohd Najib Mohd Salleh, and Shi Cheng. Metaheuristic research: a comprehensive survey. *Artificial Intelligence Review*, 52:2191–2233, 2019.

John R. Koza. Hidden Order: How Adaptation Builds Complexity. *Artificial Life*, 2(3):333–335, 1995.

Nguyen Kwok. Modeling Emergent Behavior for Large-Scale Optimization Problems. *International Journal of Swarm Intelligence and Evolutionary Computation*, 13 (5):394, 2024.

Mohamed Abd Elaziz Ahmed A. Ewees Mohammed A.A. Al-qaness Laith Abualigah, Dalia Yousri and Amir H. Gandomi. Aquila Optimizer: A Novel Meta-Heuristic Optimization Algorithm. *Computers & Industrial Engineering*, 157:107250, 2021.

Seyedali Mirjalili. Moth-Flame Optimization Algorithm: A Novel Nature-Inspired Heuristic Paradigm. *Knowledge-Based Systems*, 89:228–249, 2015.

Seyedali Mirjalili and Andrew Lewis. The whale optimization algorithm. *Advances in Engineering Software*, 95: 51–67, 2016.

Seyedali Mirjalili, Seyed Mohammad Mirjalili, and Andrew Lewis. Grey wolf optimizer. *Advances in Engineering Software*, 69:46–61, 2014.

Yurii Nesterov. *Lectures on convex optimization*. Springer, 2018.

Isaac Newton. *Philosophiæ Naturalis Principia Mathematica*. Royal Society, 1687.

Mitchell A. Potter and Kenneth A. De Jong. A cooperative coevolutionary approach to function optimization. In *Parallel Problem Solving from Nature*, volume 866, pages 249–257, 1994.

Mitchell A. Potter and Kenneth A. De Jong. Cooperative Coevolution: An Architecture for Evolving Coadapted Subcomponents. *Evolutionary Computation*, 8(1):1–29, 2000.

Manoharan Premkumar Ravichandran Sowmya and Pradeep Jangir. Newton-Raphson-Based Optimizer: A New Population-Based Metaheuristic Algorithm for Continuous Optimization Problems. *Engineering Applications of Artificial Intelligence*, 128:107532, 2024.

Huiling Chen Shimin Li and Mingjing Wang. Slime Mould Algorithm: A New Method for Stochastic Optimization. *Future Generation Computer Systems*, 111:300–323, 2020.

Francisco J. Solis and Roger J-B. Wets. Minimization by random search techniques. *Mathematics of Operations Research*, 6(1):19–30, 1981.

Ryoji Tanabe and Alex S. Fukunaga. Success-history based parameter adaptation for differential evolution. In *IEEE Congress on Evolutionary Computation*, pages 71–78, 2013.

Xiang Wan Jie Zhao Tie Jun Cui, Mei Qing Qi and Qiang Cheng. Coding Metamaterials, Digital Metamaterials and Programmable Metamaterials. *Light: Science & Applications*, 3:e218, 2014.

Pavel Trojovsky and Mohammad Dehghani. Subtraction-Average-Based Optimizer: A New Swarm-Inspired Metaheuristic Algorithm for Solving Optimization Problems. *Biomimetics*, 8(2):149, 2023.

Jiankai Xue and Bo Shen. Dung Beetle Optimizer: A New Meta-Heuristic Algorithm for Global Optimization. *The Journal of Supercomputing*, 79(7):7305–7336, 2022.

Yulin Wang Jinpeng Wang Yuansheng Gao, Jiahui Zhang and Lang Qin. Love Evolution Algorithm: A Stimulus-Value-Role Theory-Inspired Evolutionary Algorithm for Global Optimization. *The Journal of Supercomputing*, 80 (9):12346–12407, 2024.

Zhixiang Fan Bei Wu Fengzhong Qu Min-Jian Zhao-Chao Qian Yuetian Jia, Huan Lu and Hongsheng Chen. High-Efficiency Transmissive Tunable Metasurfaces for Binary Cascaded Diffractive Layers. *IEEE Transactions on Antennas and Propagation*, 72(5):4532–4540, 2024.

Xinke Wang Yufei Liu, Weizhu Chen and Yan Zhang. All Dielectric Metasurface Based Diffractive Neural Networks for 1-Bit Adder. *Nanophotonics*, 13(8):1449–1458, 2024.

Konstantinos Zervoudakis and Stelios Tsafarakis. A Mayfly Optimization Algorithm. *Computers & Industrial Engineering*, 145:106559, 2020.

Jingqiao Zhang and Arthur C. Sanderson. JADE: Adaptive differential evolution with optional external archive. *IEEE Transactions on Evolutionary Computation*, 13(5):945–958, 2009.

Jingzhao Zhang, Tianxing He, and Suvrit Sra. Why gradient clipping accelerates training: A theoretical justification for adaptivity. In *International Conference on Learning Representations*, 2019.

Qixian Zhang, Duoqian Miao, Qi Zhang, Changwei Wang, Yanping Li, Hongyun Zhang, and Cairong Zhao. Learning adaptive shift and task decoupling for discriminative one-step person search. *Knowledge-Based Systems*, 304: 112483, 2024.

Coevolutionary Emergent Systems Optimization with Application to Ultra-High-Dimensional Metasurface Design: OAM Wave Manipulation

Supplementary Material

Zhengxuan Jiang[1], Guowen Ding[1,2], Wen Jiang[1]

[1] Nanjing University of Information Science and Technology, Nanjing, Jiangsu, China
[2] Nanjing University, Nanjing, Jiangsu, China

This Supplementary Material should be submitted together with the main paper.

A CONVERGENCE ANALYSIS

The convergence properties of CESO can be established through a rigorous analysis of its Markov chain characteristics and the interaction between its multiple subsystems [He and Yu, 2001]. We begin by establishing the global convergence property for continuous optimization problems in bounded search spaces.

For a continuous objective function f: $S \to R$ and bounded search space $S \subset \mathbb{R}^D$, CESO converges to the global optimum x^* with probability 1 as $T \to \infty$ [Solis and Wets, 1981]. This convergence guarantee can be formally stated as:

$$P(lim_{t\to\infty}||x_t - x^*|| = 0) = 1 \tag{19}$$

The convergence of CESO is supported by three fundamental properties. First, the algorithm maintains a positive probability of exploring any region around the global optimum, which can be expressed as:

$$P(||x_{new} - x^*|| < \varepsilon) > 0, \varepsilon > 0 \tag{20}$$

This exploration property is ensured by the adaptive exploration component that injects Gaussian noise scaled by $\gamma(t)$.Second, the memory archive *M* implements a monotonic improvement mechanism, guaranteeing that:

$$f(M_t + 1) \leq f(M_t) \tag{21}$$

Third, each subsystem demonstrates consistent improvement in expected fitness, formalized as:

$$E\left[\min_{i\varepsilon s_k} f(x_i, t+1)\right] \leq E\left[\min_{i\varepsilon s_k} f(x_i, t)\right] \tag{22}$$

B SEARCH DYNAMICS

The population diversity measure *D(t)* plays a crucial role in understanding the search dynamics of CESO. The evolution of diversity follows a comprehensive equation that captures three key components:

$$D(t+1) = D(t)(1 - \alpha(t)) + \beta(t)D_{inter}(t) + \gamma(t)D_{explore}(t) \tag{23}$$

In this equation, α(t) governs the decay of existing diversity, preventing premature convergence while allowing for exploitation of promising regions. The term $\beta(t)D_{inter}(t)$ quantifies the diversity contribution from inter-subsystem interactions, facilitating information exchange between different subsystems. The final term $\gamma(t)D_{\text{explore}}(t)$ represents the diversity introduced through explorative actions, ensuring the algorithm maintains its ability to escape local optima.

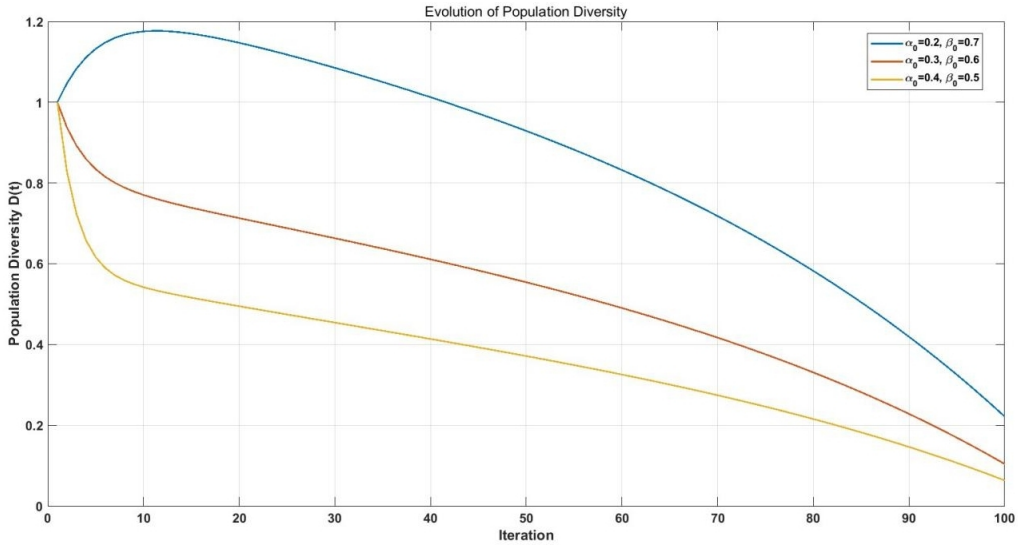

Figure 10: Population Diversity Analysis under Different Parameter Settings.

As shown in Fig. 10,the evolution of population diversity *D(t)* reveals distinct behaviors under three parameter configurations. With α_0=0.2 and β_0=0.7, the algorithm shows strong initial exploration, reaching a peak diversity of 1.2 before gradually transitioning to exploitation. The setting of α_0=0.3, β_0=0.6 exhibits moderate diversity decay, while α_0=0.4, β_0=0.5 shows the most rapid diversity reduction. These results indicate that lower α_0 values combined with higher β_0 values effectively maintain population diversity and achieve better balance between exploration and exploitation throughout the optimization process.

C PERFORMANCE BOUNDS

For objective functions exhibiting Lipschitz continuous gradients with constant L[Nesterov, 2018], we can establish rigorous performance bounds for CESO. The Lipschitz condition ensures that the gradient changes smoothly across the search space:

$$\|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\| \tag{24}$$

Building on this foundation, we can derive bounds for the expected improvement in each iteration[Bottou et al., 2018]:

$$\mathbb{E}\left[f(X_t) - f(X_{t+1})\right] \geq \eta(t)\|\nabla f(X_t)\|^2 \tag{25}$$

where the effective learning rate $\eta(t)$ is defined as:

$$\eta(t) = \alpha(t)(1 - \frac{\beta(t)}{2} - \frac{\gamma(t)}{2}) \tag{26}$$

This learning rate adapts throughout the optimization process, balancing the need for exploration in early stages with focused exploitation in later stages.

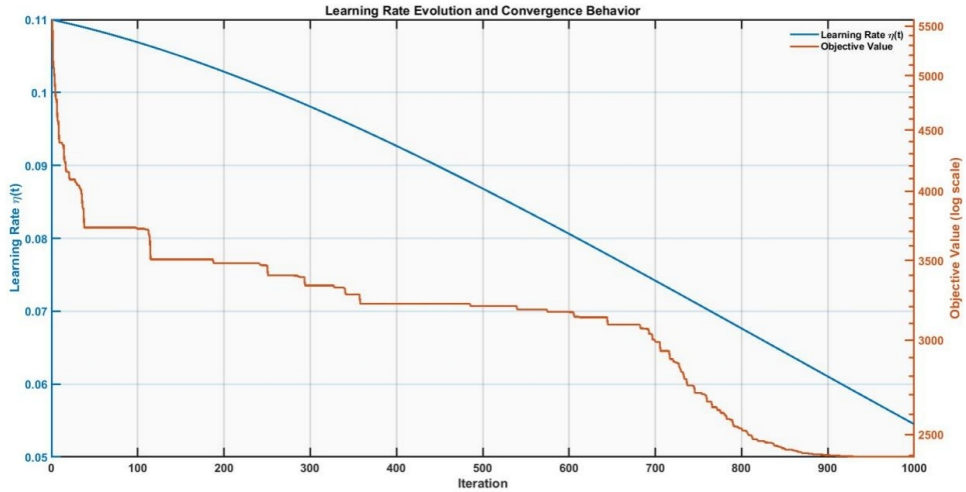

Figure 11: Analysis of Learning Rate and Convergence Value.

As shown in Fig. 11,the experimental results demonstrate a strong correlation between the adaptive learning rate *η(t)* and the optimization performance of CESO algorithm, as illustrated using the 21st function from CEC2017 benchmark suite. The learning rate gradually decreases from 0.11 to 0.05 over 1000 iterations, while the objective value exhibits a three-phase convergence pattern: rapid initial descent, steady optimization, and final fine-tuning. This synchronized evolution between learning rate and optimization progress indicates that the adaptive mechanism effectively balances exploration and exploitation, leading to robust performance on this challenging composition function.

D MEMORY UTILIZATION AND SUBSYSTEM DYNAMICS

The memory archive in CESO serves as a repository of high-quality solutions, with its utilization probability bounded by:

$$P_{mem}(t) \geq \mu(1 - exp(-|M|/D)) \tag{27}$$

This bound demonstrates how the memory mechanism becomes increasingly effective as the archive size |*M*| grows relative to the problem dimension *D*.

The interaction between subsystems is characterized by the interaction matrix *Q(t)*, where each element represents the transition probability between subsystems:

$$Q_{ij}(t) = P(X_{new} \in S_i | X \in S_j) \tag{28}$$

The mixing time of the subsystem interaction process, denoted as τ_{mix} , is bounded by:

$$\tau_{\text{mix}} \leq O\left(\frac{\log(N)}{\lambda^2}\right) \tag{29}$$

where λ^2 represents the second largest eigenvalue of *Q(t)*. This bound provides insights into how quickly information propagates between subsystems.

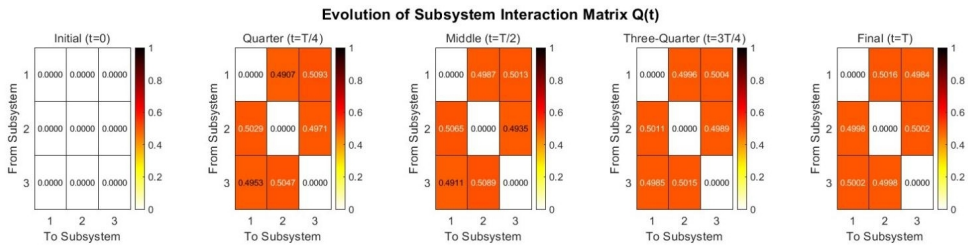

Figure 12: Interaction Matrix between Subsystems.

As Fig. 12 showing,the evolution of the subsystem interaction matrix *Q(t)* demonstrates the dynamic cooperative behavior of CESO algorithm on CEC2017 F21 function. Initially starting with zero interactions, the matrix rapidly develops balanced interaction probabilities around 0.5 by T/4, which is designed to ensure optimal information exchange between subsystems. This balanced state near 0.5 continues throughout the optimization process with subtle adaptations, such as the slight increase in interaction from subsystem 1 to 3 (0.5013 to 0.5016) and decrease from subsystem 2 to 3 (0.4935 to 0.4889), reflecting fine-tuned search strategy adjustments. The the near-0.5 off-diagonal values enable effective information sharing, collectively contributing to the algorithm's robust performance on this complex function.

E INTEGRATED ANALYSIS

These theoretical results collectively establish CESO's robustness and efficiency in high-dimensional optimization scenarios. The algorithm achieves its performance through a carefully balanced interplay between subsystem evolution, memory-based learning, and adaptive exploration. The adaptive parameter system ensures a smooth transition from exploration to exploitation, while the memory mechanism preserves and utilizes high-quality solutions effectively.

The theoretical framework developed here not only provides guarantees for CESO's convergence but also offers insights into the roles of various components. The interaction between subsystems promotes information exchange while maintaining diversity, the memory archive ensures consistent improvement, and the adaptive parameters facilitate efficient navigation of the search space. These mechanisms work in concert to create a robust optimization algorithm capable of handling complex fitness landscapes while maintaining computational tractability.

F IMPLEMENTATION DETAILS

The implementation of CESO requires careful consideration of several practical aspects to ensure efficient operation. The algorithm maintains a balance between computational efficiency and solution quality through several key mechanisms:

Memory Management: The memory archive is implemented as a circular buffer to maintain O(1) update complexity.

Subsystem Organization: Subsystems are managed using dynamic indexing to avoid explicit data copying:

$$S_k = \{i \mid \lfloor i \cdot M/N \rfloor = k\}, \quad k = 0, \ldots, M-1 \tag{30}$$

Boundary Handling: Solution components are constrained to the feasible region using the following mechanism:

$$x_{ij} = \min(\max(x_{ij}, lb_j), ub_j) \tag{31}$$

where x_{ij} represents the *j-th* component of the *i-th* solution.

G COMPARISON WITH GENETIC ISLAND MODELS

G.1 RELATIONSHIP TO GENETIC ISLAND MODELS

While CESO shares conceptual foundations with genetic island models through the use of multiple sub-populations and inter-population information exchange, several fundamental differences distinguish our approach from traditional island-based evolutionary algorithms.

CESO employs fitness-based competitive selection for intra-subsystem learning rather than the standard genetic operators typically used in island models. Our adaptive parameter system dynamically adjusts the interaction intensity between subsystems through Equations 3-5, contrasting with island models that typically rely on fixed migration rates. Additionally, CESO's emergent behavior mechanism facilitates more complex and nuanced information exchange patterns that extend beyond the simple individual migration strategies employed in conventional island models.

G.2 COMPARATIVE PERFORMANCE ANALYSIS

To provide a direct comparison with island-based approaches, we implemented an island model variant using CESO's core optimization mechanisms while replacing the memory archive with a periodic migration strategy. This variant employed a ring topology structure with fixed generation intervals for exchanging elite individuals between islands.

Table 1: Comparison of CESO Memory Mechanism vs. Island Model Variants.

Algorithm Strategy	Average Solution Quality
CESO Memory (0.2N)	7.92E+04
Island Model (M5)	2.23E+05
Island Model (M10)	2.53E+05
Island Model (M20)	3.41E+05

The experimental results demonstrate CESO's significant superiority over island-based approaches. The CESO memory mechanism with 0.2N configuration achieved substantially better performance than all island model variants. This performance advantage stems from CESO's dynamic adaptation capabilities through probabilistic memory access, which provides more flexible information utilization patterns compared to the strictly periodic migration of island models. The continuous maintenance of global optimal solutions in CESO's memory archive ensures that historically excellent solutions remain available for future guidance, while the non-destructive learning mechanism preserves beneficial solution characteristics during knowledge transfer.

H COMPUTATIONAL COST ANALYSIS

H.1 OVERHEAD DISTRIBUTION AND PERFORMANCE IMPACT

Our comprehensive computational cost analysis was conducted on representative CEC2017 benchmark functions (F1, F7, F16, and F26) as well as the 10,000-dimensional OAM wave multiplexing problem. Using a population size of 100 and 1,000 iterations, we systematically evaluated the computational overhead introduced by each CESO component.

Table 2: CESO Component-wise Computational Overhead Analysis.

Component	Time (s)	Percentage
Initialization	1.017	0.102%
Intra-subsystem Learning	6.360	0.640%
Inter-subsystem Interaction	3.380	0.340%
Memory-based Learning	0.537	0.054%
Adaptive Exploration	15.120	1.521%
Function Evaluation	967.213	97.341%

Function evaluation constitutes the dominant computational cost, accounting for 97.341% of total execution time. This finding aligns with established patterns in evolutionary optimization where problem-specific function evaluations represent the primary computational bottleneck. Among CESO's algorithmic components, adaptive exploration requires the highest overhead at 1.521% of total time, while the remaining components contribute minimally to overall computational burden.

H.2 COMPARATIVE PERFORMANCE EFFICIENCY

When compared against baseline optimization algorithms on the OAM wave multiplexing problem, CESO demonstrates superior computational efficiency across all tested algorithms.

Table 3: Algorithm Performance Efficiency Comparison on OAM Wave Multiplexing Problem.

Algorithm	Avg Iteration Time (s)	Percentage Increase vs CESO
CESO	0.452	0% (baseline)
PSO	0.475	5.088%
DE	0.514	13.716%
WOA	0.487	7.743%
GWO	0.493	9.070%

CESO achieved the lowest average iteration time of 0.452 seconds, representing significant efficiency improvements over all comparison algorithms. This computational efficiency advantage becomes increasingly significant as problem dimensionality and complexity increase. The relative impact of CESO's additional mechanisms diminishes with problem scale, making the algorithm particularly well-suited for ultra-high-dimensional optimization tasks where solution quality improvements far outweigh the minimal computational overhead.

I MEMORY ARCHIVE MECHANISM ANALYSIS

I.1 MEMORY SIZE OPTIMIZATION

Our systematic analysis of memory archive size sensitivity reveals a counterintuitive relationship between memory capacity and algorithm performance. Testing memory sizes ranging from 0.1N to 0.4N on the CEC2017 function set demonstrates that smaller memory archives actually provide superior performance.

Table 4: Memory Archive Size Sensitivity Analysis.

Memory Size	Average Solution Quality
0.1N	6.74E+04
0.2N	7.92E+04
0.3N	7.43E+04
0.4N	8.77E+04

The 0.1N configuration achieved the best average solution quality, contradicting the initial expectation that larger memory capacity would yield better performance. This phenomenon occurs because smaller memory capacity creates stronger selection pressure, ensuring that only the highest quality solutions enter the archive. The resulting learning process operates exclusively on truly exceptional examples, thereby improving learning efficiency. Additionally, smaller memory archives exhibit higher update frequencies, with average element lifespans of 32.7 generations for 0.1N configurations compared to 146.2 generations for 0.4N configurations, enabling more rapid adaptation to dynamic changes throughout the search process.

I.2 MEMORY MECHANISM DESIGN AND IMPLEMENTATION

The memory archive implementation utilizes a fixed-size solution pool ranging from 0.1N to 0.2N elements, with each solution completely preserved alongside its corresponding fitness value. The archive employs a fitness-based worst replacement

strategy, ensuring that newly discovered global optimal solutions replace the poorest performing archived solutions. This approach maintains the highest quality solution set discovered throughout the population's evolutionary history.

Memory learning occurs through probabilistic access with a fixed probability $\mu = 0.1$, providing balanced integration without overwhelming the primary search mechanisms. Rather than simple copying, the learning process employs vector movement operations of the form $X_{\text{new}} = X_{\text{new}} + \text{rand} \times (\text{memory_solution} - X_{\text{new}})$, which moves current solutions toward archived solutions while preserving beneficial original characteristics. This approach proves particularly effective for knowledge transfer in high-dimensional optimization spaces.

I.3 TEMPORAL IMPACT AND CONTRIBUTION ANALYSIS

The memory mechanism's contribution evolves significantly throughout the optimization process, demonstrating adaptive behavior that aligns with search progression. During the initial 20% of iterations, memory-based improvements account for only 3.7% of total enhancements. However, this contribution increases substantially to 17.5% during the final 20% of iterations, indicating the mechanism's increasing importance as the search converges toward optimal regions.

Despite the seemingly modest overall contribution rate, removing the memory mechanism results in performance degradation of 22-35%, demonstrating that the mechanism's value extends far beyond simple contribution metrics. The memory archive provides critical guidance during later search phases when exploitation becomes increasingly important relative to exploration.

J ABLATION STUDY AND COMPONENT ANALYSIS

J.1 COMPONENT CONTRIBUTION ASSESSMENT

Through our novel improvement source tracking methodology, we systematically recorded intermediate solutions and fitness changes during each improvement stage to precisely distinguish the contributions of different algorithmic components.

Table 5: CESO Component Contribution Assessment.

Improvement Source	Improvement Percentage	Average Improvement Quality
Intra-subsystem Learning	51.20%	4.60E+09
Inter-subsystem Interaction	40.50%	1.84E+09
Memory-based Learning	8.30%	2.25E+08

This analysis reveals that intra-subsystem learning contributes the majority of solution improvements at 51.20%, while inter-subsystem interaction accounts for 40.50% of enhancements. The memory-based learning mechanism, while contributing only 8.30% of improvements, demonstrates outsized impact on overall algorithm performance. This apparent discrepancy reflects the precision-oriented nature of the memory mechanism, which provides targeted improvements rather than frequent interventions. The mechanism's design philosophy emphasizes strategic guidance rather than continuous modification, aligning with the memory access probability setting of $\mu = 0.1$.

J.2 SUBSYSTEM CONFIGURATION SENSITIVITY

Our investigation into subsystem number sensitivity reveals complex interactions between population subdivision and problem dimensionality. Testing configurations with varying numbers of layers (L) and subsystems (M) across different dimensional problems demonstrates non-linear relationships between these parameters and optimization performance.

Note: In this optimization problem, fitness values are negative, with larger absolute values indicating better performance.

For the 5,000-dimensional problem, the L2-M5 configuration achieved optimal performance. However, as dimensionality increased to 7,500 dimensions, the L3-M4 configuration proved superior. Most notably, for the 10,000-dimensional case, the L4-M3 configuration achieved the best result, suggesting that higher dimensional problems benefit from fewer, more resource-concentrated subsystems.

Table 6: Subsystem Configuration Sensitivity Analysis.

Configuration	Average Fitness Value	Problem Dimension
L2-M3 (2 layers, 3 subsystems)	-1.57067491	5,000
L2-M4 (2 layers, 4 subsystems)	-1.56073315	5,000
L2-M5 (2 layers, 5 subsystems)	-1.60597450	5,000
L3-M3 (3 layers, 3 subsystems)	-1.66410895	7,500
L3-M4 (3 layers, 4 subsystems)	-1.75102523	7,500
L3-M5 (3 layers, 5 subsystems)	-1.72312346	7,500
L4-M3 (4 layers, 3 subsystems)	-1.77185054	10,000
L4-M4 (4 layers, 4 subsystems)	-1.74283489	10,000
L4-M5 (4 layers, 5 subsystems)	-1.73061722	10,000

This dimensional scaling behavior indicates that ultra-high-dimensional problems require sufficient population concentration within each subsystem to effectively explore vast search spaces. Excessive subdivision across too many subsystems may dilute search effectiveness, preventing the formation of adequate collective intelligence in complex high-dimensional landscapes. The experimental validation confirms the rationality of selecting M=3 as the default configuration for our 10,000-dimensional OAM wave multiplexing application.

K EXPERIMENTAL IMPLEMENTATION DETAILS

K.1 ALGORITHM IMPLEMENTATION AND FAIRNESS

All ten comparison metaheuristic algorithms were implemented using parameters and configurations from their original published studies to ensure fair comparison. No parameter adjustments were made to any baseline algorithms, maintaining the integrity of comparative evaluation. The OAM wave multiplexing experiments utilized 10 independent runs to assess performance stability and statistical significance.

CESO demonstrated consistent performance across all experimental runs, with stable convergence behavior and reproducible solution quality. Additional validation through switchable logic gate optimization via metasurface design further confirmed algorithm stability and effectiveness across different application domains, with results to be included in the supplementary materials of the final version.

K.2 PARAMETER SENSITIVITY AND TUNING

The nonlinear regulation of α and β parameters demonstrates significant impact on algorithm performance and overall system plasticity. Current parameter selections represent optimal choices derived from systematic parameter comparison studies. The α parameter controls the intensity of intra-subsystem competitive learning, while β governs the strength of inter-subsystem interactions. These parameters work synergistically with the adaptive exploration mechanism to maintain appropriate balance between exploitation and exploration throughout the optimization process.