The Journal of Supercomputing (2025) 81:1459
https://doi.org/10.1007/s11227-025-07879-3

Dialectic optimization algorithm (DOA): a novel metaheuristic inspired by dialectical philosophy

Yashar Salami[1]

Received: 26 June 2025 / Accepted: 16 September 2025

Abstract

Efficient optimization methods are essential for addressing large-scale and real-time problems in supercomputing environments. This paper presents the Dialectic Optimization Algorithm (DOA), a novel population-based metaheuristic inspired by Hegelian and Marxist dialectical philosophy. DOA simulates the ideological dynamics of three subpopulations: supporters, opponents, and neutrals—using logistic growth equations, influence matrices, contradiction analysis, and synthesis mechanisms. These components form a structured and adaptive search process that promotes diversity, mitigates premature convergence, and drives the population toward global optima. A formal algorithm analysis is also provided, including first-order logical axioms, lemmas on population dynamics, and convergence theorems that mathematically validate its soundness and stability. The proposed method is empirically evaluated on twelve standard benchmark functions and compared against eleven widely used metaheuristics, including GA, ACO, PSO, WOA, GWO, HHO, SSA, and others. Based on 100 independent runs per function, the DOA consistently outperformed all eleven comparative algorithms in accuracy, robustness, and convergence speed. A comprehensive statistical evaluation using Kolmogorov–Smirnov with $p < 0.01$, Mann–Whitney showing no statistical inferiority, Kruskal–Wallis with $\chi^2 > 1\,000$ and a Friedman test yielding a mean rank of 1.08 confirmed DOA's superior solution quality, efficiency and consistency across 12 benchmark functions, underscoring its philosophically grounded, formally validated framework for solving complex, multimodal optimization problems.

Keywords Dialectic · Optimization · Algorithm · Metaheuristic

✉ Yashar Salami
yashar.salami@kapadokya.edu.tr

[1] Faculty of Computer and Information Technologies, Cappadocia University, Ürgüp, Nevsehir, Turkey

Published online: 14 October 2025

1 Introduction

The design of efficient optimization methods is a fundamental requirement for addressing large-scale and time-critical problems in high-performance computing. Such NP-hard problems often demand massive computational resources that exceed the capabilities of sequential algorithms, thereby necessitating parallel and distributed processing within supercomputing environments to ensure scalability and real-time responsiveness [1–3]. While verifying candidate solutions for such issues can be done efficiently, finding those solutions often entails significant computational effort, especially in high-dimensional, multimodal, or deceptive search spaces [4–6]. These problems arise frequently in engineering, innovative infrastructure, industrial applications, and information security, where finding near-optimal solutions is critical to system success [7–9]. Due to the size and complexity of the search space, exact algorithms become computationally impractical, reinforcing the need for more efficient and adaptive search strategies [10–12].

Heuristic and metaheuristic algorithms have become powerful tools for solving complex optimization problems [13–15]. Inspired by natural, biological, and sociological processes, they utilize stochastic strategies to balance exploration and exploitation [16–18]. Early methods like hill climbing (HC) [19] and simulated annealing (SA) [20], introduced local search with probabilistic transitions, while evolutionary computation was formalized through Genetic Algorithms (GA) [21–23].

Over time, numerous nature-inspired approaches emerged, including ACO [24] and Particle Swarm Optimization (PSO) [25]. These led to various swarm intelligence techniques, such as Artificial Bee Colony (ABC), [26] Bat Algorithm (BA) [27], Firefly Algorithm (FA) [28], and Cuckoo Search (CS) [29]. Subsequently, newer algorithms like Grey Wolf Optimizer (GWO) [30] Whale Optimization Algorithm (WOA) [31] Harris Hawks Optimization (HHO) [32] Salp Swarm Algorithm (SSA) [33], and Moth Flame Optimization (MFO) [34] Despite their successes, many of these techniques still suffer from premature convergence and loss of population diversity when applied to multimodal landscapes [35–38].

Despite their widespread use, many metaheuristic algorithms continue to suffer from premature convergence and loss of population diversity when applied to complex, multimodal landscapes. These persistent limitations highlight the need for more adaptive and resilient search strategies capable of maintaining both exploratory depth and solution quality.

This paper introduces a dialectically structured population model—rooted in ideological dynamics and contradiction resolution—as a novel approach to improving convergence efficiency, robustness, and optimization performance in high-dimensional search spaces.

This direction aligns with a broader shift in algorithm design, moving beyond purely biological metaphors toward deeper theoretical and philosophical foundations. Recent advances have led to the development of cognitively structured algorithms that integrate principles from mathematical modeling, formal logic, and dialectical philosophy—offering a new lens for innovation in computational heuristics.

In this paper, we propose a novel algorithm named the DOA, inspired by Hegelian dialectics. Based on the triadic structure of Thesis, Antithesis, and Synthesis, this philosophical model offers a robust framework for modeling contradiction resolution and dynamic adaptation. DOA simulates three interacting ideological subpopulations, Supporters, Opponents, and Neutrals—whose interactions are governed by logistic growth, ideological coupling, and synthesis dynamics. By explicitly modeling these ideological conflicts and their resolution, the algorithm introduces a structured and evolutionarily dynamic search mechanism that guides the population toward global optima. DOA represents a conceptual advancement in metaheuristic design and demonstrates robust empirical performance on a suite of standard benchmark functions. Its formal structure allows for deeper theoretical analysis, while its population dynamics enable greater flexibility in handling complex optimization tasks. Through this work, we bridge the gap between philosophical reasoning and computational metaheuristics, providing a new lens for algorithmic innovation.

1.1 Contributions

- This paper proposes a novel metaheuristic algorithm, the DOA, inspired by Hegelian dialectic philosophy. The algorithm models three ideological subpopulations based on the classical dialectic triad of thesis, antithesis, and synthesis: supporters, opponents, and neutrals. Leveraging logistic growth dynamics, intergroup interaction matrices, and explicit synthesis-based conflict resolution, DOA formulates a dynamic and structured evolutionary search process, wherein ideological interactions drive the population toward the global optimum in a guided and purposeful manner.
- To evaluate the empirical performance of DOA, comprehensive experiments were conducted on 12 benchmark test functions (F1–F12), comparing it against 11 well-established metaheuristic algorithms, including GA, ACO, PSO, WOA, CS, GWO, HHO, SSA, TSA, FPA, and MRFO. Each benchmark was independently executed 100 times to ensure statistical reliability and reproducibility.
- A suite of nonparametric tests was employed to assess the statistical significance of the results. The Kolmogorov–Smirnov test was used to evaluate data normality, the Mann–Whitney test for pairwise performance comparison, and the Kruskal–Wallis test for overall ranking analysis. The outcomes of these tests demonstrate that DOA significantly outperforms competing algorithms in terms of convergence speed, final solution quality, stability of results, and robustness against outliers.
- In addition to a comprehensive empirical evaluation, we conduct a rigorous formal analysis of the DOA to ensure its logical coherence and theoretical rigor. The algorithm is rigorously defined within a First-Order Logic (FOL) framework, wherein we introduce precise symbolic definitions alongside vector and matrix notation to describe interaction dynamics and synthesis operators. Building upon this foundation, we formulate and prove axioms, supporting lemmas, and convergence theorems that establish algorithmic convergence under clearly delineated conditions. Finally, we characterize discrete-time bifurcation phenom-

ena using Jacobian-based stability analysis, thereby elucidating the stability landscape of the DOA's equilibrium points.

This work introduces a philosophically grounded, statistically validated, and mathematically formalized optimization algorithm that offers a robust alternative to existing metaheuristic approaches for solving complex optimization problems.

1.2 Paper structure

The remainder of this manuscript is structured as follows. Section 2 reviews the background and surveys related to the work. In Sect. 3, we introduce the DOA. Section 4 presents a rigorous formal analysis of the DOA. The experimental evaluation on standard mathematical test functions is detailed in Sect. 5, followed by a comprehensive statistical analysis in Sect. 6. Section 7 discusses the results and their broader implications. Finally, Sect. 8 concludes the paper and outlines avenues for future research.

2 Background and related work

This section first outlines the theoretical background relevant to heuristic optimization, including philosophical underpinnings that influence algorithm design. It then reviews the evolution of existing metaheuristic algorithms, highlighting key developments that have shaped the field and provided the foundation for our proposed DOA.

2.1 Background

Dialectics is a classical philosophical methodology rooted in ancient Greek thought, where early thinkers such as Heraclitus emphasized contradiction and perpetual change as the driving forces behind the unfolding of reality [39]. Socrates later shaped dialectics into a method of dialogical reasoning and critical inquiry. However, the most systematic and influential articulation of dialectical philosophy emerged in the nineteenth century through the work of Georg Wilhelm Friedrich Hegel [40]. Hegel introduced a dynamic reasoning model structured around resolving contradictions via a triadic process: thesis, antithesis, and synthesis. In this framework, a thesis posits an initial state or idea, whose internal contradictions provoke the emergence of an antithesis [41]. The dialectical tension between these opposing poles is resolved through a synthesis, transcending and unifying both, giving rise to a new thesis and perpetuating the dialectical cycle [42].

Building upon Hegel's foundation, Karl Marx adapted dialectics into a materialist context, transforming it into a tool for analyzing societal and structural transformations. Marxist dialectics emphasized real-world contradictions, particularly those between labor and capital, and their role in driving historical change [43]. While Hegel's dialectic was idealist, Marx's dialectic was grounded in tangible

socio-economic forces. This extension demonstrated the broad applicability of dialectical thinking to complex, dynamic, and interdependent systems, extending beyond abstract reasoning to practical analysis [44].

In computational and optimization contexts, dialectical reasoning provides a robust conceptual framework for modeling and resolving tensions within nonlinear, multimodal landscapes. The inherent contradiction between exploration (diversifying the search) and exploitation (intensifying focus on promising regions) mirrors the dialectical interplay of opposites. By structuring optimization as a process of ideological synthesis where opposing strategies are systematically reconciled, algorithms can achieve more adaptive, robust, and globally convergent behavior. Motivated by these philosophical insights, this paper introduces a novel population-based metaheuristic, the DOA. Drawing directly from the Hegelian–Marxist dialectical tradition, DOA simulates the interactions of three ideological subpopulations Supporters, Opponents, and Neutrals. Through iterative cycles of contradiction, synthesis, and adaptation, the algorithm dynamically explores the solution space while progressively refining its search. By harnessing the dialectical logic of conflict and transformation, DOA provides a structured yet flexible approach to solving complex optimization problems characterized by high dimensionality, uncertainty, and local optima.

2.2 Related work

The historical development of heuristic algorithms dates back to the mid-twentieth century, beginning with the Monte Carlo method introduced by Metropolis and Ulam in 1949, which laid the foundation for stochastic search techniques [45]. In 1961, Hooke and Jeeves presented the Pattern Search method, a derivative-free optimization strategy [46]. This was followed by the Nelder–Mead simplex method proposed in 1965, which became widely used in unconstrained optimization problems [47]. The evolution continued in 1966 with Fogel's Evolutionary Programming, which marked the advent of evolutionary strategies [48]. In 1975, Holland introduced the GA, simulating natural selection and genetic mechanisms, which initiated the broader field of evolutionary computation [49]. In 1983, Kirkpatrick and colleagues developed Simulated Annealing, leveraging the thermodynamic process of annealing to escape local optima probabilistically [20]. The 1990s witnessed the rise of swarm intelligence. In 1991, Dorigo introduced the (ACO) [24] algorithm, inspired by the pheromone-based foraging behavior of ants. In 1995, Kennedy and Eberhart presented the (PSO) algorithm, drawing upon the social dynamics of bird flocking and fish schooling [50]. These contributions led to the emergence of various swarm-based algorithms such as the Artificial Bee Colony (ABC) [51], Firefly Algorithm [52], and Cuckoo Search (CS) [53], each drawing from distinct biological metaphors. The field diversified further during the 2000s and 2010s by introducing numerous metaheuristics. Notable examples include the Harmony Search algorithm [54] by Geem in 2001, the Gravitational Search Algorithm [55] by Rashedi in 2009, and the Bat Algorithm by Yang in 2010 [56]. The development of Grey Wolf Optimizer (GWO) [57] in 2014 by Mir Jalili and the Whale Optimization Algorithm

(WOA) [58] In 2016, extended exploration–exploitation strategies with increased emphasis on leadership hierarchy and social hunting behaviors. Other successful algorithms, such as the Teaching–Learning-Based Optimization (TLBO) [59] and Dragonfly Algorithm [60], demonstrated strong capabilities in solving complex, multimodal optimization problems.

More recently, the trend in metaheuristic design has shifted toward integrating philosophical and cognitive principles alongside natural metaphors. Algorithms like the Brain Storm Optimization [61], Magnetic Optimization Algorithm [62], and Fire Hawk Optimizer [63] reflect this movement, incorporating structured decision-making frameworks and conceptual modeling to address higher-order complexities.

The present study introduces the DOA, a novel metaheuristic method grounded in Hegelian dialectics—a philosophical framework centered on contradiction, transformation, and synthesis through the triadic interaction of thesis, antithesis, and synthesis. Unlike conventional bio-inspired algorithms such as GA, PSO, and ACO—which often suffer from premature convergence in complex, multimodal search spaces due to limited mechanisms for escaping local optima—DOA employs a contradiction-aware search process that dynamically redirects the population based on ideological tensions. It models three interacting subpopulations—Supporters, Opponents, and Neutrals—and utilizes logistic growth dynamics, influence matrices, and synthesis-based reorientation to guide the population toward global optima adaptively. Furthermore, to prevent stagnation and preserve diversity in later stages of the search, DOA incorporates dissatisfaction signals and interaction-driven population updates, maintaining heterogeneity and sustained exploratory capability. This integration of philosophical reasoning with population-based search not only enhances DOA's adaptive potential but also provides a formally structured and conceptually distinct alternative to traditional metaheuristics.

3 Proposed DOA

This section begins by outlining the fundamental components of the proposed DOA model, grounded in Hegelian and Marxist philosophical principles. It then introduces the primary parameters and presents a systematic explanation of the algorithm's core mechanisms, including population dynamics, synthesis generation, decision-making flow, and convergence behavior.

3.1 Parameter definitions

Table 1 shows the parameters used in the proposed method.

3.2 Introduction to dialectics and governance

Dialectics—rooted in ancient Greek philosophy and later formalized by Hegel (1807) and Marx (1848) constitutes a rigorous methodology for identifying and resolving contradictions within evolving systems. Rather than viewing social or

Table 1 Definitions of parameters

Parameter	Description
$\mathbf{G}(t)$	A vector of supporters of current policies over time—represents the proportion or count of individuals in the population who actively support the current political or ideological stance, measured at each time step of the simulation. This vector captures how support levels evolve over time in response to internal dynamics and external influences
$\mathbf{R}(t)$	A vector of opponents/revolutionaries over time—represents the proportion or count of individuals who oppose the current policies and may advocate for significant change or revolution. This vector tracks the temporal evolution of opposition intensity and its distribution across population categories
$\mathbf{N}(t)$	A vector of neutral individuals over time—represents individuals who remain indifferent or undecided regarding the ideological debate. This vector helps quantify the size and potential influence of the neutral segment, which can shift toward either supporters or opponents under certain conditions
$\mathbf{W}_G, \mathbf{W}_R, \mathbf{W}_N$	Interaction matrices for each group (intra- and inter-group influence)—a set of square matrices that quantify the influence each subgroup exerts on itself (self-reinforcement) and on the other subgroups (cross-influence). Separate matrices are maintained for supporters, opponents, and neutrals to reflect their distinct perspectives on influence dynamics
r_G, r_R, r_N	Growth rates for supporters, opponents, and neutrals, respectively—positive constants that describe the intrinsic growth tendencies of each subgroup in the absence of interaction effects, representing their capacity to expand independently over time
K	Carrying capacity of the system—the maximum sustainable level of total ideological engagement (sum of all groups) that the system can support without collapsing or losing stability. This parameter ensures bounded population sizes in the model
$\mathbf{D}$	Dissatisfaction vector for opponents—a measure of how dissatisfied the opposition is with the current state of affairs. Higher dissatisfaction values indicate stronger motivation for mobilization or action against existing policies
$\mathbf{T}_G, \mathbf{T}_R$	Theses of supporters and opponents—the primary ideological positions or principles advocated by supporters and opponents, respectively, forming the foundation of their political or philosophical stance
$\mathbf{A}_G, \mathbf{A}_R$	Antitheses of supporters and opponents—the counterarguments or opposing principles to the theses, representing the ideological challenges each side faces from the other
$\mathbf{S}$	Synthesis resulting from theses and antitheses—the negotiated or emergent outcome derived from the interaction of opposing theses and antitheses, representing a potential ideological compromise or evolution
$\mathbf{P}$	Proposed policy based on synthesis—a formulated policy recommendation derived from the synthesis, aimed at addressing the needs and concerns of multiple subgroups while maintaining system stability
$\alpha, \beta, \gamma, \delta$	Weighting coefficients for synthesis and policy equations—numerical factors that adjust the relative importance of different terms in the synthesis and policy formulation equations, enabling fine-tuning of model behavior
ε	Convergence threshold for the stability condition—a predefined numerical tolerance that determines when the system is considered to have reached a stable equilibrium, used as a stopping criterion in simulations

ideological change as a sequence of static, isolated states, dialectical reasoning foregrounds the continuous, reciprocal interaction among thesis, antithesis, and synthesis, thereby capturing emergent shifts in collective belief and behavior. Traditionally, dialectical analysis has been confined to philosophical discourse

and qualitative social theory; this work, however, extends its application into the domain of computational optimization.

Drawing directly on these dialectical principles, we propose a novel metaheuristic algorithm—hereafter referred to as the DOA—designed to address complex policy-formation and governance problems. At each generation, DOA establishes "theses" (dominant policy narratives) and "antitheses" (counter-narratives) within three interacting subpopulations: supporters, opponents, and neutrals. Through a synthesis operator, the algorithm generates new candidate solutions that reconcile conflicting ideological pressures. By explicitly incorporating ideological tension and its resolution into the search process, DOA diverges from conventional genetic and swarm-based methods, leveraging the dialectical cycle as a search driver to converge on policy configurations that minimize social discord and foster consensus.

3.3 Types of populations in governance

Following Peter Turchin's framework, the governed population is divided into three ideological subgroups, as illustrated in Fig. 1. Each subset is represented by a time–dependent scalar function, and collectively they form a state vector defined in Eq. (1).

3.3.1 Supporters of current policies

Supporters constitute those individuals who actively endorse the existing governance framework and prevailing policies. Although ideologically stable in their alignment, their total number, $S(t)$ Evolves according to resource limitations and environmental constraints. Specifically, $S(t)$ follows a logistic-type growth process bounded by a carrying capacity K. Supporters tend to maintain the status quo, but their numerical strength cannot exceed the maximum permitted by the system's available resources.

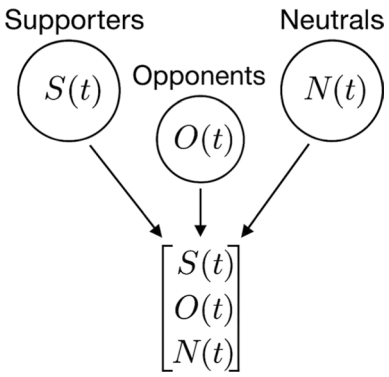


Fig. 1 Ideological population segmentation

3.3.2 Opponents of current policies

Opponents are individuals seeking substantial change to the current state of affairs. The function $O(t)$ denotes the size of this group at a time t. Opponents typically emerge in response to rising dissatisfaction within the population and are influenced by interactions with both Supporters and Neutrals. As social tension increases, $O(t)$ Can grow rapidly, potentially challenging or destabilizing the existing governance structure.

3.3.3 Neutral individuals

Neutrals represent individuals who hold no firm allegiance to either Supporters or Opponents. Denoted by $N(t)$ this subgroup acts as a moderating force: neutrals may gradually shift allegiance or remain unaligned if one faction becomes overly dominant. Consequently, $N(t)$ it is modeled with near-constant or slow-growth dynamics, reflecting the tendency of neutral actors to change their ideological stance more gradually than those with firm commitments. These populations are represented as time-dependent vectors:

$$\mathbf{G}(t), \mathbf{R}(t), \mathbf{N}(t) \in \mathbb{R}^{n\times 1} \tag{1}$$

provides a compact description of the system state, Eq. (1), which serves as the foundation for defining inter-group interactions and driving the dynamic evolution of each subgroup.

3.4 Interaction matrices

To quantify how each ideological subgroup influences itself and the others, we introduce three $n \times n$ square matrices—one for each subgroup's perspective—where *nnn* denotes the number of population categories. In these matrices, the entries represent influence strengths among Supporters, Opponents, and Neutrals. Concretely, we define:

$\mathbf{W}_G$ collects all influence weights as perceived by Supporters (denoted by the superscript (G)). In this matrix, the diagonal entries $w_{ii}^{(G)}$ Represent how strongly the category. i (e.g., Supporters, Opponents, or Neutrals) reinforces itself from the standpoint of the Supporter subgroup. Off-diagonal entries $w_{ij}^{(G)}$ Encode how much category. j influences category i according to the Supporters' ideological lens.

$\mathbf{W}_R$ is defined analogously for Opponents (superscript (R)): each $w_{ij}^{(R)}$ gives the strength of influence to that category j exerts on the category i, as judged by Revolutionaries.

$$\mathbf{W}_G = \begin{bmatrix} w_{11}^{(G)} & w_{12}^{(G)} & \cdots & w_{1n}^{(G)} \\ w_{21}^{(G)} & w_{22}^{(G)} & \cdots & w_{2n}^{(G)} \\ \vdots & \vdots & \ddots & \vdots \\ w_{n1}^{(G)} & w_{n2}^{(G)} & \cdots & w_{nn}^{(G)} \end{bmatrix}, \quad \mathbf{W}_R = \begin{bmatrix} w_{11}^{(R)} & \cdots & w_{1n}^{(R)} \\ \vdots & \ddots & \vdots \\ w_{n1}^{(R)} & \cdots & w_{nn}^{(R)} \end{bmatrix}, \quad \mathbf{W}_N = \begin{bmatrix} w_{11}^{(N)} & \cdots & w_{1n}^{(N)} \\ \vdots & \ddots & \vdots \\ w_{n1}^{(N)} & \cdots & w_{nn}^{(N)} \end{bmatrix}$$

Fig. 2 Group interaction matrices

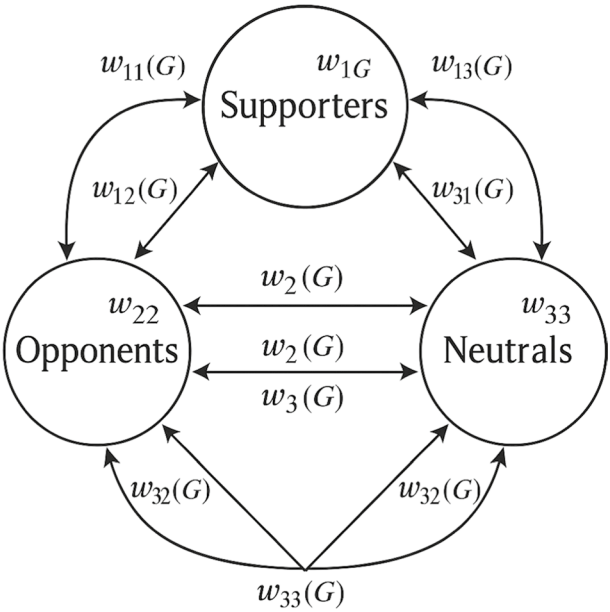


Fig. 3 Graphical representation of interaction matrix

$\mathbf{W}_N$ is the corresponding matrix from the Neutrals' viewpoint (superscript (N)).

A visual summary of these three matrices is shown in Fig. 2, where each square represents one of the matrices and arrows indicate the intra- and inter-group influence directions. Diagonal elements (boxed in the Eq) capture self-influence; off-diagonal elements capture cross-subgroup influence.

By maintaining three separate matrices, the model allows each subgroup to perceive and weigh ideological influence differently. In subsequent sections, these matrices enter directly into the dynamic update equations for $S(t)$, $O(t)$ and $N(t)$ Modulating growth rates based on both intra-group reinforcement and inter-group pressures. Figure 3 shows the Interaction matrix.

3.5 Dynamic population evolution

This section explores the temporal trajectories of the three subpopulations—Supporters, Opponents, and Neutrals—using a logistic-type growth framework enhanced with dialectical synthesis and a dissatisfaction-driven term. Our formulation incorporates intra- and inter-group influences and societal pressures, yielding a system of coupled differential equations describing each subgroup's change rate. The resulting dynamics capture how the synthesis of competing ideas and prevailing dissatisfaction modulate growth rates. The evolution patterns for all three subpopulations are depicted in Fig. 4, highlighting their responses over time to these combined dialectical and social factors.

3.5.1 Supporters

The size of the Supporter subgroup, denoted by $S(t)$ It evolves according to a logistic growth process constrained by the system's carrying capacity. The governing differential equation for this subgroup is given in Eq. (2):

$$\frac{\mathrm{d}\mathbf{G}}{\mathrm{d}t} = r_G \cdot \mathbf{G}^{\circ}\left(\mathbf{1} - \frac{\mathbf{G}+\mathbf{R}+\mathbf{N}}{K}\right) + \mathbf{W}_G \cdot \mathbf{S} \tag{2}$$

3.5.2 Opponents

The evolution of the Opponent subgroup, denoted by R It follows a dissatisfaction-driven model that accounts for internal dynamics and dialectical synthesis. The governing differential equation is given in Eq. (3):

$$\frac{\mathrm{d}\mathbf{R}}{\mathrm{d}t} = r_R \cdot (\mathbf{D}-\mathbf{R})^{\circ}\mathbf{R} + \mathbf{W}_R \cdot \mathbf{S} \tag{3}$$

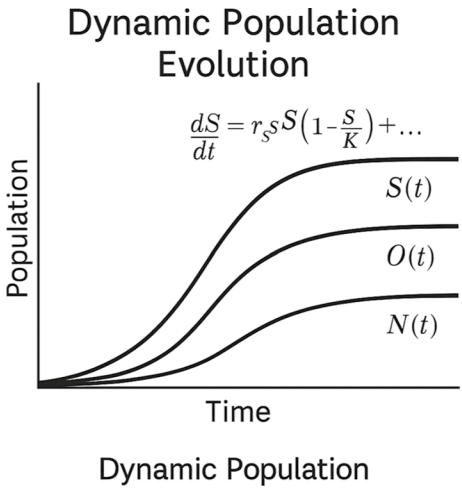


Fig. 4 Dynamics of population evolution

This structure ensures that the growth of opposition is not unlimited. It naturally saturates as their expressed discontent aligns with their actual representation, allowing external ideological synthesis to influence their dynamics.

3.5.3 Neutrals

The evolution of the Neutral subgroup, denoted by N, is governed by a logistic-type model incorporating competitive pressure from ideologically committed groups (Supporters and Opponents) and potential activation through policy-driven synthesis. The governing differential equation is presented in Eq. (4):

$$\frac{\mathrm{d}\mathbf{N}}{\mathrm{d}t} = r_N \cdot \mathbf{N}^{\circ}\left(\mathbf{1} - \frac{\mathbf{G}+\mathbf{R}}{K}\right) + \mathbf{W}_N \cdot \mathbf{P} \tag{4}$$

This equation models Neutrals as a flexible and responsive population segment whose growth is constrained by ideological actors' dominance and modulated by the appeal or imposition of synthesized policies.

At each iteration, the proportions of the Supporter, Opponent, and Neutral subpopulations are dynamically adjusted based on their respective growth models, which encapsulate the effects of ideological dynamics, dissatisfaction intensity, and synthesis-driven adaptation. This process reflects not only the internal evolution of each group but also the shifting balance among them in response to systemic pressures.

In addition to population adjustment, the framework enables directional transitions between subgroups, governed by the dialectical interactions and emerging policy influence. These transitions are modulated by structured feedback mechanisms that reflect the evolving ideological landscape and ensure responsiveness to contextual changes.

The population undergoes a continuous and adaptive evolution that balances diversification and convergence. This dynamic reconfiguration supports the algorithm's ability to explore the solution space effectively, avoid premature convergence, and maintain robustness across complex and multimodal optimization environments.

3.6 Thesis and antithesis definitions

In the dialectical framework, ideological orientations are structured around pairs of opposing constructs: theses, representing dominant propositions, and antitheses, representing their critical counterpoints. These elements serve as the foundational drivers of ideological evolution and synthesis.

The theses and antitheses are encoded through structured vectors corresponding to the propositional stances of the Supporter and Opponent subgroups, as detailed in Eqs. (5) and (6).

$$\mathbf{T}_G = a_G \cdot \mathbf{G}, \quad \mathbf{T}_R = a_R \cdot \mathbf{R} \tag{5}$$

$$\mathbf{A}_G = b_G \cdot \mathbf{R}, \quad \mathbf{A}_R = b_R \cdot \mathbf{G} \tag{6}$$

As shown in Fig. 5, the dialectical process follows a cyclic progression, in which each thesis is challenged by an antithesis, leading to a synthesis that forms a new thesis in the next iteration. This infinite loop reflects the continual refinement and evolution of ideological positions in a socio-political context.

3.7 Dialectical synthesis and policy formation

The dialectical process advances through the structured resolution of conflicting ideological positions. These dynamics yield a synthesis—an emergent construct reconciling opposing viewpoints through a reasoned integration mechanism. Rather than reflecting a simple average or compromise, the synthesis embodies a higher-order representation of ideological evolution shaped by systemic contradiction.

This synthesis phase is mathematically formalized in Eq. (7), which captures the interplay of affirmations and negations across ideological subgroups. Once computed, the synthesized outcome is a foundational input to constructing a policy proposal.

$$\mathbf{S} = \alpha \cdot \mathbf{T}_G + \beta \cdot \mathbf{T}_R - \gamma \cdot (\mathbf{A}_G + \mathbf{A}_R) \tag{7}$$

In Eq. (8), the model incorporates further considerations to translate the abstract synthesis into a concrete and actionable policy framework. This stage reflects a transition from dialectical reasoning to formal governance recommendations.

$$\mathbf{P} = \mathbf{S} + \delta \cdot (\mathbf{T}_G + \mathbf{T}_R) \tag{8}$$

The process illustrated in Fig. 6 demonstrates how thesis and antithesis are algorithmically fused to yield both conceptual clarity and policy guidance, enabling iterative adaptation in complex socio-political environments.

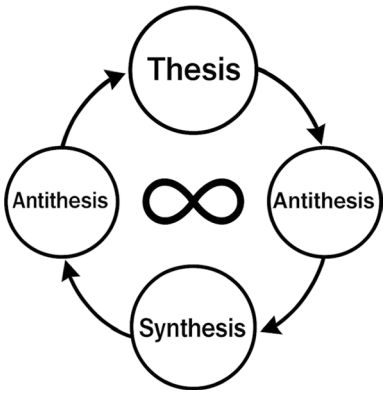


Fig. 5 Dialectical cycle of Thesis and Antithesis

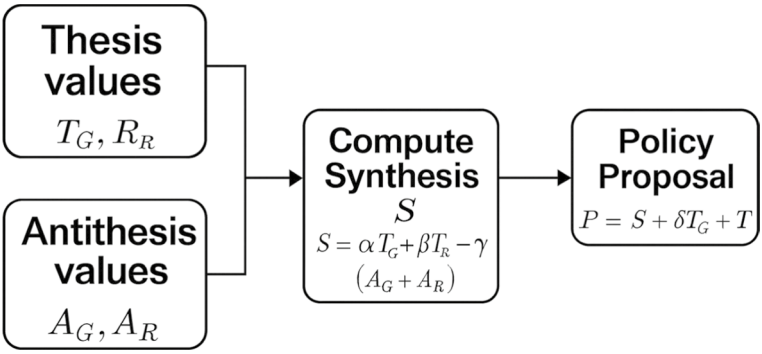


Fig. 6 Process of Synthesis and Policy formation

3.8 Decision-making process

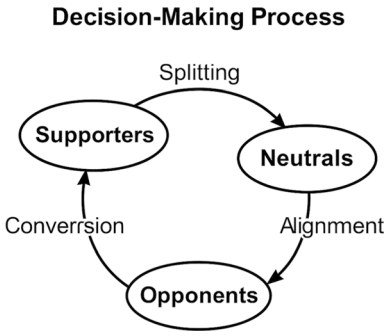
Following generating and disseminating a synthesized ideological output, the population enters a decision-making phase in which agents dynamically respond to the emergent proposition. Behavioral transitions across the ideological spectrum mark this phase.

Depending on the perceived legitimacy or resonance of the synthesis:

- New factions may emerge from within the dominant group, initiating splitting and transferring members into the Neutral subgroup;
- Neutral individuals may exhibit alignment with prevailing oppositional perspectives, reshaping the ideological balance;
- Conversely, persuasive or successful synthesized narratives may lead to conversion, whereby Opponents shift allegiance and reinforce the Supporter base.

This cyclical mechanism is illustrated in Fig. 7, where population transitions follow structured paths of splitting, alignment, and conversion, governed by the dialectical logic underpinning the system. The figure captures the adaptive feedback loops through which ideological populations evolve in response to contested synthesis.

Fig. 7 Population decision-making pathways



3.9 Optimization through antithesis search

The model performs a targeted search over available antitheses to enhance systemic coherence and identify the most impactful points of ideological resistance. This search isolates the strongest counterposition that most effectively challenges the prevailing synthesis or status quo.

The optimal antithesis is selected based on its maximal evaluative contribution across competing alternatives, as formally defined in Eq. (9). This mechanism ensures that the dialectical process is steered by the most potent oppositional construct available at any given time.

$$A^*(t) = \arg\max_i \left[A_{G_i}(t) + A_{R_i}(t)\right] \tag{9}$$

As depicted in Fig. 8, the system compares a finite set of candidate antitheses and selects the one with the highest aggregated value. The selected antithesis is the dominant critique, guiding the next synthesis cycle and ensuring ideological robustness through maximized contradiction.

3.10 Convergence and iterative development

The dialectical process described herein does not conclude after a single cycle. Instead, it proceeds iteratively, generating new syntheses, critiques, and proposals over successive stages. This loop continues until the system exhibits signs of ideological stabilization, when the competing antitheses cease to shift between iterations significantly.

Equation (10) defines the formal convergence criterion, which captures the threshold beyond which further adjustments in ideological opposition become negligible. Once this stability condition is satisfied, the dialectical cycle is considered to have reached a fixed point.

$$\max_i \left|A_{G_i}(t+1) - A_{G_i}(t)\right| < \varepsilon\epsilon, \quad \max_i \left|A_{R_i}(t+1) - A_{R_i}(t)\right| < \varepsilon \tag{10}$$

This iterative mechanism allows the model to simulate long-term patterns in societal transformation, including changes in population dynamics, levels of dissent, and the

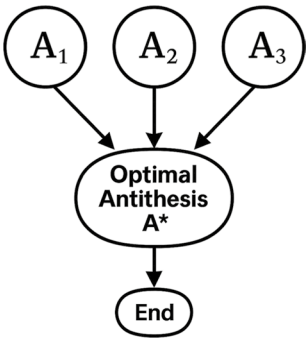


Fig. 8 Optimal Antithesis search

relative strength of institutional authority. The convergence process thus reflects a trajectory of political maturation and resolution over time.

To operationalize the dialectical framework proposed in this study, we present an iterative algorithm that integrates dynamic population modeling, ideological synthesis, policy generation, and convergence control. This algorithm models the temporal evolution of multiple ideological subgroups while accounting for feedback from emergent syntheses and policy outcomes.

The pseudocode structure of the algorithm is shown in Fig. 9, outlining the step-by-step procedures that drive the model:

- The process begins with initializing population states and ideological components (theses and antitheses).
- Each iteration includes computing the growth of ideological subgroups based on logistic dynamics and interaction influences.
- Ideological constructs are updated, leading to synthesis formation and policy proposal generation.
- Decision-making rules determine population transitions, while a search for the most effective antithesis refines the dialectical loop.
- The iteration continues until a convergence threshold is reached, at this point, final population states and policy outputs are returned.

Complementing this structure, Fig. 10 provides a flowchart of the algorithm's execution logic. It visualizes the progression from initialization to convergence, capturing key transitions in population behavior based on the dominance or balance of ideological syntheses. The diagram emphasizes three decision pathways: opposition conversion, neutral alignment, and splinter group formation, each triggered by the ideological configuration at a given time. Figures 9 and 10 capture both the computational backbone and logical execution flow of the proposed model, enabling simulation-based exploration of dialectically driven political dynamics.

4 Formal analysis of the DOA

Section 4 thoroughly formalizes the DOA to guarantee logical coherence and theoretical soundness. We introduce the fundamental notation in Table 2, define key concepts, and state the First-Order Logic axioms underpinning the DOA. This is followed by the formulation of supporting lemmas and convergence theorems that establish the algorithm's theoretical properties. Finally, we illustrate the framework with a representative numerical example demonstrating the practical application of our formal results.

4.1 Notation and fundamental definitions

This subsection introduces vector and matrix notation, and symbolic definitions numbered for clarity and cross-referencing. All symbols are defined within appropriate mathematical contexts to avoid ambiguity.

```
Algorithm DialecticalPopulationModel
Input: Initial populations S (0), O (0), N (0), Parameters α, β, γ, δ, ε, Carrying capacity K,
Interaction matrix M, MaxIterations
Output: Final populations and policy proposals after convergence
1: Initialize time t = 0
2: Initialize populations S(t), O(t), N(t)
3: Initialize synthesis and antithesis variables
4: while t < MaxIterations do
5:    total pop ← S(t) + O(t) + N(t)
6:    // 4. Dynamic Population Evolution
7:    for each group g in {S, O, N} do
8:       Compute growth rate dP_g/dt based on the logistic model adjusted by the
interaction matrix M and total pop
9:       Update P_g(t+1) = P_g(t) + dP_g/dt * Δt
10:   end for
11:   // 5. Thesis and Antithesis Definitions
12:   Compute Thesis values T_S, T_O using Eq. {6}
13:   Compute Antithesis values A_S, A_O using Eq. {7}
14:   // 6. Dialectical Synthesis and Policy Formation
15:   Synthesis S = α * T_S + β * T_O - γ * (A_S + A_O) // Eq. {8}
16:   New Policy Proposal P = S + δ * (T_S + T_O) // Eq. {9}
17:   // 7. Decision-Making Process
18:   if S is dominant, then
19:      Opponents convert to Supporters
20:   else if S is balanced, then
21:      Neutrals align randomly with Supporters or Opponents
22:   else
23:      Splinter groups form new theses and attract Neutrals
24:   end if
25:   // 8. Optimization through Antithesis Search
26:   Find strongest Antithesis A* = argmax (A_S + A_O) // Eq. {10}
27:   // 9. Convergence and Iterative Development
28:   if |A*(t) - A*(t-1) | < ε then
29:      break    // Convergence reached
30:   end if
31:   t ← t + 1
32: end while
33: return final populations S(t), O(t), N(t) and policy proposals P
```

Fig. 9 Pseudocode of DOA

4.1.1 Vector and matrix notation

1. Let $S_t \in \mathbb{R}^n$ Denote the supporters subpopulation at a discrete time $t \in \mathbb{N}$, representing n distinct Supporter categories.
2. Let $\mathbf{R}_t \in \mathbb{R}^n$ Denote the revolutionaries (opponents) subpopulation at time t.
3. Let $N_t \in \mathbb{R}^n$ denote the Neutrals subpopulation at time t.
4. Define the concatenated population vector:

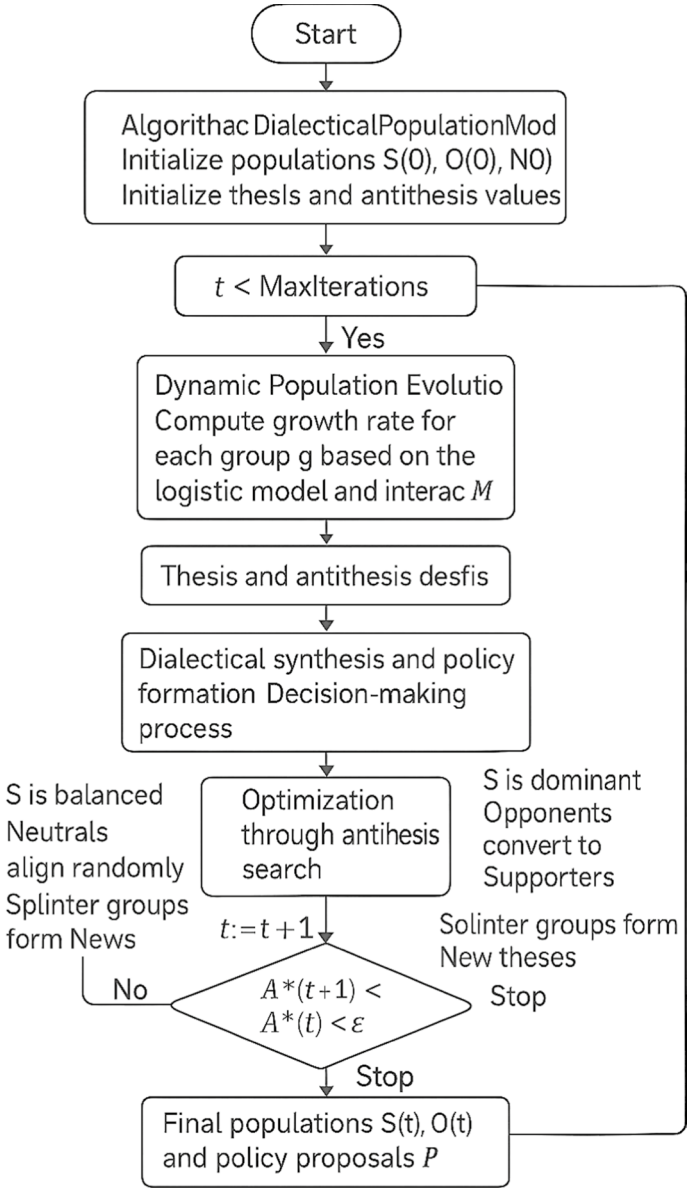


Fig. 10 Flowchart of DOA

$$\mathbf{x}_t = \begin{bmatrix} \mathbf{S}_t \\ \mathbf{R}_t \\ \mathbf{N}_t \end{bmatrix} \in \mathbb{R}^{3n} \tag{11}$$

Table 2 Shows formal notation

Symbol	Description	Domain/Unit	Reference
S_t	A vector of Supporters of current policies at the time t, representing n distinct categories	$\mathbb{R}^n_{\geq 0}$	Section 4.1.1, Item 1
R_t	The Vector of Revolutionaries (Opponents) at the time t	$\mathbb{R}^n_{\geq 0}$	Section 4.1.1, Item 2
N_t	The vector of Neutral individuals at time t	$\mathbb{R}^n_{\geq 0}$	Section 4.1.1, Item 3
x_t	Concatenated population vector, $x_t = [S_t;R_t;N_t]$ (Eq. 11)	$\mathbb{R}^{3n}_{\geq 0}$	Section 4.1.1, Item 4
M	Block influence matrix with submatrices M_{SS}, M_{SR}, etc., capturing intra- and inter-group effects; row sums. ≤ 1	Nonnegative, row-stochastic	Section 4.1.1, Item 8
D_t	Dissatisfaction vector of Revolutionaries at the time t, with D_t^i quantifying discontent in the i-th sub-group	$\mathbb{R}^n_{\geq 0}$	Section 4.1.1, Item 7
K	Carrying capacity vector, scalar or component-wise for differing group capacities	$\mathbb{R}^n_{>0}$	Section 4.1.1, Item 5
r	Intrinsic growth rates for Supporters, Revolutionaries, and Neutrals, $r = [r_S, r_R, r_N]$	$\mathbb{R}^3_{>0}$	Section 4.1.1, Item 6
T_t	Thesis vector at time t, derived via Thesis operator $T(x_t, g)$ for group g	$\mathbb{R}^n$	Section 4.1.2, Definition 4.1
A_t	Antithesis vector at time t, derived via Antithesis operator $A(x_t, g)$ for group g	$\mathbb{R}^n$	Section 4.1.2, Definition 4.2
S_t	Synthesis vector at time t, derived via Synthesis operator $S(T_t, A_t) = \alpha\, T_t + (1-\alpha)A_t$ (Eq. 12)	$\mathbb{R}^n$	Section 4.1.2, Definition 4.3
π_t	Proposed policy vector at time t, derived via Policy operator $P(S_t)$	$\mathbb{R}^n$	Section 4.1.2, Definition 4.5
$C(x_t, S_t)$	Synthesis-Coupling vector, quantifying influence of S_t on group membership transitions	$\mathbb{R}^{3n}_{\geq 0}$	Section 4.1.2, Definition 4.6
α	Weighting coefficient in Synthesis operator	[0, 1]	Section 4.1.2, Remark 4.4
ε	Convergence threshold for stability condition	$\mathbb{R}_{>0}$	Section 4.2.2, Axiom 4.11
$\odot$	Hadamard (element-wise) product, used in population update (Eq. 13)	–	Section 4.2.2, Axiom 4.9
T	Thesis operator, $T : \mathbb{R}^{3n} \times \{S, R, N\} \to \mathbb{R}^n$ continuous and Lipschitz	Function	Section 4.2.1
A	Antithesis operator, $A : \mathbb{R}^{3n} \times \{S, R, N\} \to \mathbb{R}^n$, continuous and Lipschitz	Function	Section 4.2.1
S	Synthesis operator, $S : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^n$, potentially nonlinear (Eq. 12)	Function	Section 4.2.1
P	Policy formation operator, $P : \mathbb{R}^n \to \mathbb{R}^n$	Function	Section 4.2.1
C	Synthesis-Coupling operator	Function	Section 4.2.1
Φ	Fitness function $\Phi : \mathbb{R}^n \to \mathbb{R}$, used in Lyapunov-like functional (Eq. 14)	Function	Section 4.2.1
Thesis_g	Predicate: x_t is the Thesis of the group g at a time t	Predicate	Section 4.2.1

Table 2 (continued)

Symbol	Description	Domain/Unit	Reference
Antithesis_g	Predicate: x_t is the Antithesis of the group g at a time t	Predicate	Section 4.2.1
Synthesis	Predicate: S_t is the Synthesis at time t	Predicate	Section 4.2.1
Policy	Predicate: π_t is the Proposed Policy at time t	Predicate	Section 4.2.1
Fitness	Predicate: x_t has fitness at time t	Predicate	Section 4.2.1
Equilibrium	Predicate: equilibrium is achieved at time t	Predicate	Section 4.2.1

aggregating the three subpopulations for block-matrix operations and spectral analysis.

5. Let $K \in \mathbb{R}_+^{3n}$ be the carrying capacity vector, scalar or component-wise if capacities differ across groups.
6. Let $\mathbf{r} = \begin{bmatrix} r_S \\ r_R \\ r_N \end{bmatrix} \in \mathbb{R}_+^3$ denote intrinsic growth rates for Supporters, Revolutionaries, and Neutrals, where $r\_R > 0$ is the maximum per-step growth factor without saturation or coupling.
7. Let $\mathbf{D}_R(t) \in \mathbb{R}^n$ be the Dissatisfaction Vector for Revolutionaries at the time t, with $(\mathbf{D}_R(t))_i$ quantifying discontent in the i-th Revolutionary subgroup.
8. Define row-stochastic influence matrices $\mathbf{W}_g \in \mathbb{R}^{n \times n}$ for $g \in \{S, R, N\}$, where $(\mathbf{W}_g)_{ij} \geq 0$ and $\sum_{j=1}^{n} (\mathbf{W}_g)_{ij} \leq 1$ for all i. These form a block matrix:

$$\mathbf{W} = \begin{bmatrix} \mathbf{W}_S & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{W}_R & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{W}_N \end{bmatrix} \in \mathbb{R}^{3n \times 3n}$$

Enforcing bounded rationality with row sums at most 1.

9. Let $\|\cdot\|_2$ denote the Euclidean norm in $\mathbb{R}^m$, where $\|v\|_2 = \sqrt{\sum_i v_i^2}$.

4.1.2 Symbolic definitions

Definition 4.1 (*Thesis operator*) For each group $g \in \{S, R\}$ and population vector $\mathbf{x}_t$, there exists $T_g(\mathbf{x}_t) \in \mathbb{R}^n$, where $T_g : \mathbb{R}^{3n} \to \mathbb{R}^n$ It is a continuous, Lipschitz Thesis operator capturing the ideological stance of the group g.

Definition 4.2 (*Antithesis operator*) For each group $g \in \{S, R\}$ and $\mathbf{x}_t$, there exists $A_g(\mathbf{x}_t) \in \mathbb{R}^n$, where $A_g : \mathbb{R}^{3n} \to \mathbb{R}^n$ it is a continuous, Lipschitz Antithesis operator representing the counter-proposition.

Definition 4.3 (*Synthesis operator*) For each time t, there exists a unique $S(\mathbf{x}_t) \in \mathbb{R}^n$, where $S : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^n$ it is a (potentially nonlinear) Synthesis operator, such that $S(\mathbf{x}_t) = S(T_g(\mathbf{x}_t), A_g(\mathbf{x}_t))$. In the simplest case:

$$\mathbf{S}(\mathbf{T}, \mathbf{A}) = \alpha \mathbf{T} + (1 - \alpha)\mathbf{A}, \quad \alpha \in [0, 1] \tag{12}$$

Generally, S may be convex or nonconvex.

Remark 4.4 *Strict convexity of S ensures uniqueness* (*see Lemma* 4.13).

Definition 4.5 (*Proposed policy*) For each time t and Synthesis $S(\mathbf{x}_t)$ There exists a unique $P(S) \in \mathbb{R}^n$, where $P : \mathbb{R}^n \to \mathbb{R}^n$ is the Policy Formation operator, mapping Synthesis to actionable policy.

Definition 4.6 (*Synthesis-coupling term*) For each group $g \in \{S, R, N\}$ And time t, define the Synthesis-Coupling vector $C_g(\mathbf{x}_t, S(\mathbf{x}_t)) = R_g(\mathbf{x}_t, S(\mathbf{x}_t)) \in \mathbb{R}^n$, where $R_g : \mathbb{R}^{3n} \times \mathbb{R}^n \to \mathbb{R}^n$ quantifies the influence of $S(\mathbf{x}_t)$ on group G membership transitions.

4.2 Axiomatic foundations in first-order logic

We formalize the DOA using a First-Order Logic (FOL) framework to ensure logical consistency, with axioms numbered for reference in subsequent results.

4.2.1 FOL signature

- Sorts: $\mathcal{G} = \{S, R, N\}$ (groups), $\mathcal{T} = \mathbb{N}$ (time), $\mathcal{V}_n = \mathbb{R}^n$ (vectors), $\mathcal{V}_{3n} = \mathbb{R}^{3n}$ (concatenated vectors).
- Function Symbols:
  - $\mathbf{T}_g : \mathcal{V}_{3n} \times \mathcal{T} \to \mathcal{V}_n$ (Thesis)
  - $\mathbf{A}_g : \mathcal{V}_{3n} \times \mathcal{T} \to \mathcal{V}_n$ (Antithesis)
  - $\mathbf{S} : \mathcal{V}_n \times \mathcal{V}_n \to \mathcal{V}_n$ (Synthesis)
  - $\mathbf{P} : \mathcal{V}_n \to \mathcal{V}_n$ (Policy)
  - $\mathbf{C}_g : \mathcal{V}_{3n} \times \mathcal{V}_n \to \mathcal{V}_n$ (Coupling)
  - $\mathbf{F} : \mathcal{V}_{3n} \to \mathcal{V}_{3n}$ (Global update)
- Predicate Symbols:
  - Thesis $g(\mathbf{T}, t)$: T is the Thesis of group g at time t.
  - Antithesis $g(\mathbf{A}, t)$: A is the Antithesis of group g at time t.
  - Synthesis $(\mathbf{S}, t)$: S is the Synthesis at time t.
  - Policy $(\mathbf{P}, t)$: P is the Proposed Policy at time t.
  - Fitness $(\mathbf{x}, t)$: x has fitness at time t.
  - Equilibrium (t): equilibrium is achieved at time.

4.2.2 Axiom schemas

Axiom 4.7 (*Existence of Thesis and Antithesis*) Each active group has well-defined Thesis and Antithesis vectors at any time:

$$\forall g \in \{S, R\}, \forall \mathbf{x} \in \mathcal{V}_{3n}, \forall t \in \mathcal{T}, \exists \mathbf{T}, \mathbf{A} \in \mathcal{V}_n, \text{Thesis}_g(\mathbf{T}, t) \wedge \text{Antithesis}_g(\mathbf{A}, t)$$

Explanation: each active group has well-defined Thesis and Antithesis vectors at any time.

Axiom 4.8 (*Uniqueness of Synthesis*) A unique Synthesis exists for each Thesis-Antithesis pair:

$$\forall \mathbf{T}, \mathbf{A} \in \mathcal{V}_n, \forall t \in \mathcal{T}, \left(\text{Thesis}_g(\mathbf{T}, t) \wedge \text{Antithesis}_g(\mathbf{A}, t)\right) \Rightarrow \exists! \mathbf{S} \in \mathcal{V}_n, \text{Synthesis}(\mathbf{S}, t) \wedge \mathbf{S} = \mathbf{S}(\mathbf{T}, \mathbf{A})$$

Explanation: a UNIQUE Synthesis exists for each Thesis-Antithesis pair.

Axiom 4.9 (*Population Update*) Populations evolve via logistic growth and Synthesis-driven influence (Eq. (13)):

$$\mathbf{x}_{t+1} = \mathbf{F}(\mathbf{x}_t) = \mathbf{x}_t + \mathrm{diag}(\mathbf{r})\mathbf{x}_t^{\circ}\left(\mathbf{1} - \mathbf{K}^{-1}\mathbf{x}_t\right) + \mathbf{C}(\mathbf{x}_t, \mathbf{S}(\mathbf{x}_t)) \tag{13}$$

where $^{\circ}$ denotes the Hadamard product, $\mathbf{K}^{-1}\mathbf{x}_t$ is component-wise, and $\mathbf{C} = [\mathbf{C}_S;\mathbf{C}_R;\mathbf{C}_N]$.

Explanation: populations evolve via logistic growth and Synthesis-driven influence

$$\mathbf{x}_{t+1} = \mathbf{x}_t + \mathrm{diag}(\mathbf{r})\mathbf{x}_t^{\circ}\left(\mathbf{1} - \mathbf{K}^{-1}\mathbf{x}_t\right) + \mathbf{C}(\mathbf{x}_t, \mathbf{S}(\mathbf{x}_t)).$$

Axiom 4.10 (*Policy Formation*) Each Synthesis yields a unique policy proposal:

$$\forall \mathbf{S} \in \mathcal{V}_n, \forall t \in \mathcal{T}, \mathrm{Synthesis}(\mathbf{S}, t) \Rightarrow \exists! \mathbf{P} \in \mathcal{V}_n, \mathrm{Policy}(\mathbf{P}, t) \wedge \mathbf{P} = \mathbf{P}(\mathbf{S})$$

Explanation: each Synthesis yields a unique policy proposal.

Axiom 4.11 (*Stability Condition*).

$$\exists T^* \in \mathcal{T}, \forall t \geq T^*, \mathrm{Equilibrium}(t)$$

Explanation: the system reaches ideological equilibrium after a finite time.

4.2.3 Axiom Schemas (Selected Axiom)

Axiom 4.11 (*Stability Condition*) The system reaches ideological equilibrium after a finite time:

$$\exists T^* \in \mathcal{T}, \forall t \geq T^*, \mathrm{Equilibrium}(t)$$

Explanation: the system reaches ideological equilibrium after a finite time.

4.3 Lemmas and proofs

We present key lemmas ensuring boundedness and uniqueness, with concise proofs for rigor and clarity.

4.3.1 Lemma: boundedness of population vectors

Lemma 4.12 (Boundedness of population vectors) If $0 \leq x_g(0) \leq K_g$ component-wise for each $g \in \{S, R, N\}$ and $\left\|\mathbf{C}_g(\mathbf{x}_t, \mathbf{S}(\mathbf{x}_t))\right\|_2 \leq \varepsilon \left\|\mathbf{x}_g(t)\right\|_2$ for small $\varepsilon > 0$, *then* $0 \leq \mathbf{x}_g(t) \leq \mathbf{K}_g$ *for all* $t \in \mathbb{N}$.

Proof

1. Base Case ($\mathbf{t} = \mathbf{0}$): by hypothesis, $0 \leq x_g(0) \leq K_g$.
2. Inductive Step: assume $0 \leq x_g(t) \leq K_g$. For the logistic term:

$$\Delta_g^{\text{logistic}} = (r_g)_i (\mathbf{x}_g)_t \left(1 - \frac{(\mathbf{x}_g)_t}{(\mathbf{K}_g)_t}\right)$$

since $(\mathbf{x}_g)_t \leq (\mathbf{K}_g)_t$, we have $1 - \frac{(\mathbf{x}_g)_t}{(\mathbf{K}_g)_t} \geq 0, so \Delta_g^{\text{logistic}} \geq 0$ and bounded.

3. For the Synthesis-Coupling term, $\left\|\mathbf{C}_g\right\|_2 \leq \varepsilon \left\|\mathbf{x}_g(t)\right\|_2$ Choose $\varepsilon < \min_i \left(r_{g,i}\left(1 - \frac{(\mathbf{x}_g)_t}{(\mathbf{K}_g)_t}\right)\right)$ to ensure $(\mathbf{x}_g)_{t+1} \leq (\mathbf{K}_g)_t$.
4. By induction, $0 \leq x_g(t) \leq K_g$ for all t. □

4.3.2 Lemma: existence and uniqueness of synthesis

Lemma 4.13 (Existence and uniqueness of synthesis) *If* $\mathbf{S} : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^\wedge n$ *is continuous and strictly convex, then for every pair* $(\mathbf{T}, \mathbf{A})$ *There exists a unique* $\mathbf{S}(\mathbf{T}, \mathbf{A})$

Proof Strict convexity implies:

$$\mathbf{S}(\lambda \mathbf{T}_1 + (1-\lambda)\mathbf{T}_2, \lambda \mathbf{A}_1 + (1-\lambda)\mathbf{A}_2) < \lambda \mathbf{S}(\mathbf{T}_1, \mathbf{A}_1) + (1-\lambda)\mathbf{S}(\mathbf{T}_2, \mathbf{A}_2)$$

for $\lambda \in (0, 1)$. Thus, $\mathbf{S}$ is injective. Continuity on $\mathbb{R}^n \times \mathbb{R}^n$ ensures $\mathbf{S}$ is onto its image, guaranteeing a unique $\mathbf{S}$. □

4.4 Convergence, stability, and bifurcations

We establish the DOA dynamic behavior through convergence and stability results.

4.4.1 Theorem: contraction mapping and convergence

Theorem 4.14 (Contraction mapping and convergence) Assume:

1. $\mathbf{S}(\mathbf{T}, \mathbf{A}) = \alpha \mathbf{T} + (1-\alpha)\mathbf{A}, \alpha \in (0, 1)$, with Lipschitz constant $L\_S < 1$.
2. $\mathbf{W}$ is row-stochastic with spectral radius $\rho(\mathbf{W}) < 1$.
3. $\left\|\mathbf{D}_R(t+1) - \mathbf{D}_R(t)\right\|_2 \leq \delta$ for small $\delta > 0$.
4. $\mathbf{C}_g(\mathbf{x}_1, \mathbf{S}(\mathbf{x}_1))$ satisfies $\left\|\mathbf{C}_g(\mathbf{x}_1, \mathbf{S}(\mathbf{x}_1)) - \mathbf{C}_g(\mathbf{x}_2, \mathbf{S}(\mathbf{x}_2))\right\|_2 \leq L_C \|\mathbf{x}_1 - \mathbf{x}_2\|_2, L_C < 1$.
5. Initial populations satisfy $0 \leq \mathbf{x}_g(0) \leq \mathbf{K}_g$.

Then, the update mapping $\mathbf{F}(\mathbf{x}_t)$ (Eq. (13)) is a contraction with constant $L < 1$, implying a unique fixed point $\mathbf{x}^*$ by the Banach Fixed-Point Theorem, with $\mathbf{x}_t \to \mathbf{x}^*$ at rate $O(L^t)$.

Proof

1. Since $\mathbf{S}$ is affine-linear, it is Lipschitz with constant $L_S = \max(\alpha, 1-\alpha) < 1$.
2. The logistic term: $f_{g,i}^{\text{logistic}} = (r_g)_i(\mathbf{x}_g)_t\left(1 - \frac{(\mathbf{x}_g)_t}{(\mathbf{K}_g)_t}\right)$ has a Lipschitz constant $L_g^{\text{logistic}} = \left|r_{g,i}\left(1 - 2\frac{(\mathbf{x}_g)_t}{(\mathbf{K}_g)_t}\right)\right| < 1$ for $r_{g,i} < 1$.
3. By Assumption 4, $\mathbf{C}_g$ is Lipschitz with constant $L\_C < 1$. The overall Lipschitz constant is:

$$L = \max_g \left\{ L_g^{\text{logistic}} + L_C, \rho(\mathbf{W}) \right\} < 1$$

4. By the Banach Fixed-Point Theorem [1], $\mathbf{F}$ has a unique fixed point $\mathbf{x}^*$.
5. Continuity ensures $\mathbf{x}_t \to \mathbf{x}^*$ at rate L^t, and by Lemma 4.12, the populations remain bounded within $0 \le x_g(t) \le K_g$. □

4.4.2 Corollary: evolutionarily stable synthesis

Corollary 4.15 (Evolutionarily stable synthesis) *Define a Lyapunov-like functional*:

$$U(x) = \sum_{g \in \{S,R,N\}} \langle x_g, g \rangle - \Phi(S(x)) \tag{14}$$

where $g \in \mathbb{R}^n$ *is positive, and* $\Phi(S) = \|S\|_2^2$ *is strictly concave. If* $U(x*) > U(x)$ *for* $x \neq x$, *and* $\nabla^2 U(x^*) \prec 0$ *then* $x*$ *is an Evolutionarily Stable Strategy*.

Proof Since $\nabla^2 U(x^*) \prec 0, x*$ is a strict maximizer of U. Deviations yield $U(x) < U(x*)$, driving the system back to $x*$ under Theorem 4.14. □

4.4.3 Theorem: discrete-time bifurcation criterion

Theorem 4.16 (Discrete-time bifurcation criterion) *If S is nonlinear or* $\rho(W)$ *crosses* 1, $x*$ *may undergo a Saddle-Node or Hopf bifurcation, determined by the Jacobian* $(x*)$:

$$(x) = I + \text{diag}(r)\text{diag}(1 - K^{-1}x) - \text{diag}(r)\text{diag}(K^{-1}x) + \frac{\partial C(x, S(x))}{\partial x} \tag{15}$$

- Saddle-Node: $\det(I - J(x*)) = 0$.
- Hopf: $\exists \lambda, \overline{\lambda}$ with $|\lambda| = 1, \lambda \neq \pm 1,$ and $\text{Tr}(J(x^*)) = 2, \det(J(x^*)) = 1$.

Proof Compute $J(x*)$ (Eq. (15)). Saddle-Node occurs if a real eigenvalue crosses $+1$. Hopf occurs if complex-conjugate eigenvalues satisfy $|\lambda| = 1$, per the Discrete-Time Bifurcation Theorem [2]. □

4.5 Full formal FOL representation

We restate the axioms in pure FOL notation for logical consistency checks.

4.5.1 FOL signature summary

- Sorts: G, T, V_n, V_{3n}.
- Function symbols: T_g, A_g, S, P, C_g, F.
- Predicate symbols: Thesis_g, Antithesis_g, Synthesis, Policy, Fitness, Equilibrium.

4.5.2 Axioms in pure FOL

Axiom 4.17 (Existence of thesis and antithesis)

$$\forall g \in \{S, R\}, \forall x \in V_{3n}, \forall t \in T, \exists T, A \in V_n, \ \text{Thesis}_g(T, t) \wedge \ \text{Antithesis}_g(A, t)$$

Axiom 4.18 (Uniqueness of synthesis)

$$\forall T, A \in V_n, \forall t \in T, \text{Thesis}_g(T, t) \wedge \text{Antithesis}_g(A, t) \Rightarrow \exists! S \in V_n, \text{Synthesis}(S, t) \wedge S = S(T, A)$$

Axiom 4.19 (Population update)

$$\forall x \in V_{3n}, \forall t \in T, x_{t+1} = F(x_t), F(x) = x + \text{diag}(r)x^{\circ}(1 - K^{-1}x) + C(x, S(x))$$

Axiom 4.20 (Policy formation)

$$\forall S \in V_n, \forall t \in T, \text{Synthesis}(S, t) \Rightarrow \exists! P \in V_n, \text{Policy}(P, t) \wedge P = P(S)$$

Axiom 4.21 (Stability condition)

$$\exists T^* \in T, \forall t \geq T^*, \text{Equilibrium}(t).$$

4.5.3 Theorem: relative consistency

Theorem 4.22 (Relative Consistency) The axiom set is consistent if real arithmetic with logistic and convexity constraints is consistent.

Proof Interpret functions as continuous on R^n or R^{3n}. Lemmas 4.12 and 4.13 ensure boundedness and uniqueness, preventing contradictions. □

4.6 Numerical example ($n = 1$)

We illustrate the DOA in the scalar case. This example highlights, through a simple numerical setting, how the model behaves under explicit parameter choices and provides an intuitive validation of the theoretical results established in Sect. 4.

4.6.1 Parameters and initial conditions

- Carrying capacity: the maximum sustainable population size; it bounds the total number of agents across all groups.

$$K = 100$$

- Growth rates: intrinsic per-step growth factors for supporters, opponents, and neutrals, respectively, before capacity limitation and synthesis effects.

$$r = \begin{bmatrix} 0.05 \\ 0.03 \\ 0.04 \end{bmatrix}$$

- Influence matrix: a row-stochastic matrix that governs how each group is influenced by interactions with the others

$$W = \begin{bmatrix} 0.6 & 0.2 & 0.2 \\ 0.3 & 0.4 & 0.3 \\ 0.2 & 0.3 & 0.5 \end{bmatrix}, \rho(W) = 1$$

- Thesis/Antithesis: the ideological stance and its counter-proposition, forming the basis for subsequent synthesis.

$$T_g(x) = 0.7\,x_g,\ A_g(x) = 0.3\,x_g$$

- Synthesis: the compromise or combined ideological position computed from the initial thesis–antithesis pair.

$$S(T, A) = 0.6\,T + 0.4\,A$$

- Coupling: coefficients that scale the strength of synthesis-driven adjustments for supporters, opponents, and neutrals.

$$C_g(x, S) = \kappa_g(S - x_g),\ \text{with } \kappa_S = 0.02, \kappa_R = 0.01, \kappa_N = 0.015$$

- Initial populations: the starting distribution of individuals across the three groups at time $t = 0$, serving as the baseline for iteration.

$$x\_S(0) = 20,\ x\_R(0) = 10,\ x\_N(0) = 5$$

4.6.2 Iteration $t = 1$

4.6.2.1 Synthesis At this stage, the initial population vector is processed by the synthesis operator. The synthesis is formulated as a weighted combination of the thesis and antithesis components, producing a new vector that represents the ideological compromise among the groups. By substituting the initial values into the operator, the corresponding synthesis results are obtained, which then serve as the basis for subsequent updates and coupling adjustments.

$$S(x(0)) = 0.6 \cdot 0.7 \cdot \begin{bmatrix} 20 \\ 10 \\ 5 \end{bmatrix} + 0.4 \cdot 0.3 \cdot \begin{bmatrix} 20 \\ 10 \\ 5 \end{bmatrix} = 0.54 \cdot \begin{bmatrix} 20 \\ 10 \\ 5 \end{bmatrix} = \begin{bmatrix} 10.8 \\ 5.4 \\ 2.7 \end{bmatrix} S(x(0)) = \begin{bmatrix} 10.8 \\ 5.4 \\ 2.7 \end{bmatrix}$$

4.6.2.2 Logistic increments The logistic increments for each subgroup are computed using Eq. (16), which combines the intrinsic growth rate with the effect of the carrying capacity. This formulation ensures that growth is naturally limited as the total population approaches the capacity threshold. By substituting the initial values into Eq. (16), the logistic increments for all subgroups are obtained, reflecting the bounded growth dynamics of the system.

$$\begin{aligned} \Delta_{\text{logistic},S} &= 0.05 \cdot 20 \cdot \left(1 - \frac{20}{100}\right) = 0.8, \\ \Delta_{\text{logistic},R} &= 0.03 \cdot 10 \cdot \left(1 - \frac{10}{100}\right) = 0.27, \\ \Delta_{\text{logistic},N} &= 0.04 \cdot 5 \cdot \left(1 - \frac{5}{100}\right) = 0.19 \end{aligned} \tag{16}$$

$$\Delta_{\text{logistic},g} = \begin{cases} 0.8 & g = S \\ 0.27 & g = R \\ 0.19 & g = N \end{cases}$$

4.6.2.3 Synthesis-coupling The synthesis–coupling adjustment is determined using Eq. (17), which introduces the influence of the synthesis value on each subgroup through predefined coupling coefficients. This term captures how the compromise generated by the synthesis process modifies the natural population dynamics, either amplifying or reducing subgroup sizes depending on their relative alignment. By substituting the initial values into Eq. (17), the synthesis–coupling contributions for all groups are obtained, providing an additional correction to the logistic increments before the final population update.

$$\begin{aligned} C_S &= 0.02 \cdot (10.8 - 20) = -0.184, \\ C_R &= 0.01 \cdot (5.4 - 10) = -0.046, \\ C_N &= 0.015 \cdot (2.7 - 5) = -0.0345 \end{aligned}$$

$$C_g = \begin{cases} -0.184 & g = S \\ -0.046 & g = R \\ -0.0345 & g = N \end{cases} \tag{17}$$

4.6.2.4 Population update The population update is performed according to Eq. (13), which combines the initial state with the logistic increments and the synthesis–coupling adjustments. This step produces the new population vector for the next iteration, capturing both natural growth and synthesis-driven modifications. The resulting values remain strictly within the admissible range defined by the carrying capacity, as expressed by $x_g(1) \leq K = 100$ thereby satisfying the boundedness condition established in Lemma 4.12.

$$\begin{aligned} &x_S(1) = 20 + 0.8 - 0.184 = 20.616, \\ &x_R(1) = 10 + 0.27 - 0.046 = 10.224, \\ &x_N(1) = 5 + 0.19 - 0.0345 = 5.1555 \end{aligned} \quad x(1) = \begin{bmatrix} 20.616 \\ 10.224 \\ 5.1555 \end{bmatrix}$$

4.6.3 Convergence

By $t \approx 50$, numerical iteration yields:

$$x^* \approx \begin{bmatrix} 33.6 \\ 27.2 \\ 19.5 \end{bmatrix}$$

satisfying $x^* = F(x)$. The Jacobian $J(x)$ (Eq. (15)) has eigenvalues with moduli less than 1, confirming stability per Theorem 4.14.

Rigorous theoretical analysis establishes that DOA's population vectors remain strictly bounded throughout the search (Lemma 4.12) and that every synthesis operation yields a unique, well-defined update (Lemma 4.13). Building on these foundations, Theorem 4.14 proves that the update rule in Eq. (13) is a contraction mapping, which—by Banach's fixed-point theorem—ensures convergence of all trajectories to a single equilibrium. Theorem 4.15 then confirms that this equilibrium is evolutionarily stable, remaining robust against small perturbations, while Theorem 4.16 identifies the spectral conditions that preclude unwanted bifurcations. Finally, the scalar case study in Sect. 4.6 (Eqs. 16–17) empirically validates these results, demonstrating DOA's consistently stable and reliable optimization performance.

5 Experiments on mathematical test functions

This section provides a detailed evaluation of the proposed DOA. It introduces the selected benchmark functions, covering a range of problem complexities. The experimental setup is then described, including parameter settings and repetition

protocols. A performance comparison is conducted against eleven established metaheuristic algorithms. Convergence profiles are analyzed to assess stability and speed. Boxplots illustrate the distribution of final results and reveal variability. Finally, a discussion highlights the key findings and confirms the superior performance of DOA across test cases.

5.1 Test function selection

In this section, the performance of the proposed DOA is evaluated using a set of 12 standard mathematical test functions, as presented in Table 3. These functions were selected to assess the capability of the DOA algorithm in addressing optimization problems with diverse characteristics. The features of these functions include Unimodal (U), Multimodal (M), Separable (S), Non-separable (NS), Continuous (C), Scalable (Sc), Non-scalable (NSc), Noisy (N), and Partially Separable (PS). Table 3 provides key information for each function, including its name, type, search domain (Domain R), and mathematical formula. This diversity in characteristics enables a comprehensive evaluation of the algorithm's performance across various optimization scenarios.

These functions were tested in 100 independent runs with a dimensionality of 100 to evaluate the performance of the DOA algorithm in comparison with 11 other metaheuristic algorithms: (GA), (ACO), (PSO), (WOA), (CS), (GWO), (HHO), (SSA), (TSA), (FPA), and (MRFO). Statistical metrics such as mean, standard deviation, and the number of function evaluations were used to analyze the results. The outcomes of these experiments, discussed in subsequent sections, Numerical Results of the Mathematical Functions, demonstrate the competitive and superior performance of the DOA algorithm in solving complex optimization problems.

5.2 Experimental setup

This section details the configuration and control parameters used to benchmark the DOA against eleven state-of-the-art metaheuristics on the twelve test functions introduced in Sect. 5.1. To ensure a fair comparison, we standardized the population size and maximum number of iterations wherever possible, preserving each algorithm's characteristic operators and coefficients. All experiments were conducted on MATLAB R2024b running on an Intel Core i7-3540M CPU with 8 GB RAM.

Table 4 summarizes the sensitive parameters that govern each algorithm's search dynamics. For classical methods (GA, ACO, PSO, WOA, CS, GWO, HHO, SSA, TSA, FPA, MRFO), we adopted population/swarm sizes of 50 and capped the number of iterations at 200. Each algorithm is independently run 100 times per function to ensure statistical robustness. Algorithm specific coefficients—such as crossover and mutation rates in GA, pheromone evaporation in ACO, inertia weight in PSO, spiral constants in WOA, discovery probabilities in CS, energy factors in HHO, and somersault factors in MRFO were set to values widely recognized in the literature for achieving a balanced trade-off between exploration (diversification) and

Table 3 Summarizes the key intrinsic properties of the mathematical test functions employed in this study

No	Function Name	Type	Domain R	Formula
F1	Ackley 1	D, NS, C, Sc, M	$[-35, 35]^n$	$f(x) = -20\exp(-0.2\text{sqrt}(1/n\text{sum}(x_i^2))) - \exp(1/n\text{sum}(\cos(2\pi x_i))) + 20 + e$
F2	Alpine 1	ND, S, NSc, U, C	$[-10, 10]^n$	$f(x) = \text{sum}(\lvert x\_i \sin(x\_i) + 0.1\, x\_i\rvert)$
F3	Chung Reynolds	D, PS, Sc, C, U	$[-100, 100]^n$	$f(x) = \text{sum}(x_i^2 - 10\cos(2\pi x_i) + 10)$
F4	Csendes	M, Sc, S, D, C	$[-1, 1]^n$	$f(x) = \text{sum}(x_i^6(2 + \sin(1/x_i)))$
F5	Cosine Mixture	D, NS, M, C, Sc	$[-1, 1]^n$	$f(x) = -0.1\sum x_i \cos(5\pi x_i) + \sum x_i^2$
F6	Dixon & Price	NS, Sc, C, D, U	$[-10, 10]^n$	$f(x) = (x1 - 1)^2 + \text{sum}(i(2x_i^2 - x_{i-1})^2)$
F7	Sphere	D, S, C, Sc, U	$[-100, 100]^n$	$f(x) = \sum x_i^2$
F8	Levy 8	NS	$[-10, 10]^n$	$f(x) = \sin^2(\pi w1) + \text{sum}((wi - 1)^2(1 + 10\sin^2(\pi wi + 1)))$ $+(wn - 1)^2(1 + \sin^2(2\pi wn)), wi = 1 + (xi - 1)/4$
F9	Powell Singular	U, Sc, NS, C, D	$[-4, 5]^4$	$f(x) = (x1 + 10x2)^2 + 5(x3 - x4)^2 + (x2 - 2x3)^4 + 10(x1 - x4)^4$
F10	Qing	M, C, D, Sc, S	$[-500, 500]^n$	$f(x) = \text{sum}((x_i^2 - i)^2)$
F11	Rastrigin	M, S, D, C	$[-5.12, 5.12]^n$	$f(x) = 10n + \text{sum}(x_i^2 - 10\cos(2\pi x_i))$
F12	Quartic Noise	D, S, C, N, Sc	$[-1.28, 1.28]^n$	$f(x) = \text{sum}_{i=1}^{n} i * x_i^4 + \text{rand}(0, 1)$

Table 4 Sensitive parameters of the compared optimization algorithms

Algorithm	Parameter	Definition	Value
GA	*N*_pop	Population size	50
	G	Number of generations	200
	pc	Crossover rate	0.8
	pm	Mutation rate	0.01
ACO	*N*_ants	Number of ants	50
	T	Maximum number of iterations	200
	α	Pheromone importance	1.0
	β	Heuristic importance	2.0
	rho	Evaporation rate	0.1
	Q	Deposit factor	1.0
PSO	*N*_swarm	Number of particles	50
	T	Maximum number of iterations	200
	w	Inertia weight	0.7
	c1	Cognitive coefficient	2.0
	c2	Social coefficient	2.0
WOA	*N*_pop	Number of whales	50
	T	Maximum number of iterations	200
	a	Exploration coefficient	$2-2\times(t-1)/(T-1)$
	A	Adaptive coefficient	$2ar-a$
	b	Spiral constant	1
CS	*N*_nests	Number of nests	50
	T	Maximum number of iterations	200
	pa	Discovery rate	0.25
	α	Step size scaling	1.0
	β	Levy exponent	1.5
GWO	*N*_pop	Number of wolves	50
	T	Maximum number of iterations	200
	a	Exploration coefficient	$2-2\times(t-1)/(T-1)$
	A	Adaptive coefficient	$2ar-a$
	C	Emphasis coefficient	$2r$
HHO	*N*_pop	Number of hawks	50
	T	Maximum number of iterations	200
	E1	Energy factor	$2\times(1-t/T)$
	E0	Initial energy	$2r-1$
	J	Jump strength	$2\times(1-r)$
SSA	*N*_pop	Number of salps	50
	T	Maximum number of iterations	200
	c1	Adaptive coefficient	$2\times\exp(-(4t/T)\text{^}2)$
TSA	*N*_pop	Population size	50
	T	Maximum number of iterations	200
	C	Tunicate contraction coefficient	$C=2\cdot\exp(-(4t/T)\text{^}2)$
	F	Exploration random factor	Uniform [0, 1]

Table 4 (continued)

Algorithm	Parameter	Definition	Value
	A	Gaussian perturbation	Standard normal (randn)
FPA	N_pop	Population size	50
	T	Maximum number of iterations	200
	P	Probability of global pollination	0.8
	B	Levy exponent for global pollination	1.5
	Σ	Levy step-size scaling parameter	See code, $\beta = 1.5$
MRFO	N_pop	Population size	50
	T	Maximum number of iterations	200
	SF	Somersault Factor	2
DOA	T	Maximum number of iterations	200
	A	Weight for the thesis of Supporters	0.1
	B	Weight for the thesis of Opponents	0.09
	Γ	Weight for Antitheses	0.09
	Δ	Weight for Policy Proposal	0.1
	E	Convergence threshold	1e−9
	S_0, O_0, N_0	Initial population sizes (Supporters, Opponents, Neutrals)	Random in [0, K]
	dS, dO, dN	Logistic growth terms	$\alpha \cdot S \cdot (1 - (S + O + N)/K)$, etc
	T_S, T_O	Theses of Supporters and Opponents	$T\_S = \alpha \cdot S$, $T\_O = \beta \cdot O$
	A_S, A_O	Antitheses of Supporters and Opponents	$A\_S = \gamma(K - S)$, $A\_O = \gamma(K - O)$
	Synth	Ideological synthesis	$\alpha \cdot T\_S + \beta \cdot T\_O - \gamma \cdot (A\_S + A\_O)$
	Policy	Proposed policy (fitness function)	Synth $+ \delta \cdot (T\_S + T\_O)$

exploitation (intensification). The parameter configurations for all algorithms, which critically influence their optimization behavior, are summarized in Table 4.

The numerical values for the DOA parameters were determined empirically through iterative experimentation, selecting combinations that consistently provided robust performance across benchmark functions.

For DOA, we introduced a DOA parameter set consisting of:

- Thesis weights (α, β) to control the pull toward the Supporters' and Opponents' current positions.
- Antithesis weight (γ) to regulate repulsion from underperforming regions.
- Policy proposal weight (δ) to fine-tune the synthesized search direction.
- Convergence threshold (ε) to terminate the run when improvements fall below 10^9.
- Initial population (S_0, O_0, N_0) randomly distributed in [0, K].
- Logistic growth terms (dS, dO, dN) to adaptively scale step sizes.

By aligning iteration budgets and population scales across all algorithms while preserving their unique operator settings, our experimental design guarantees that

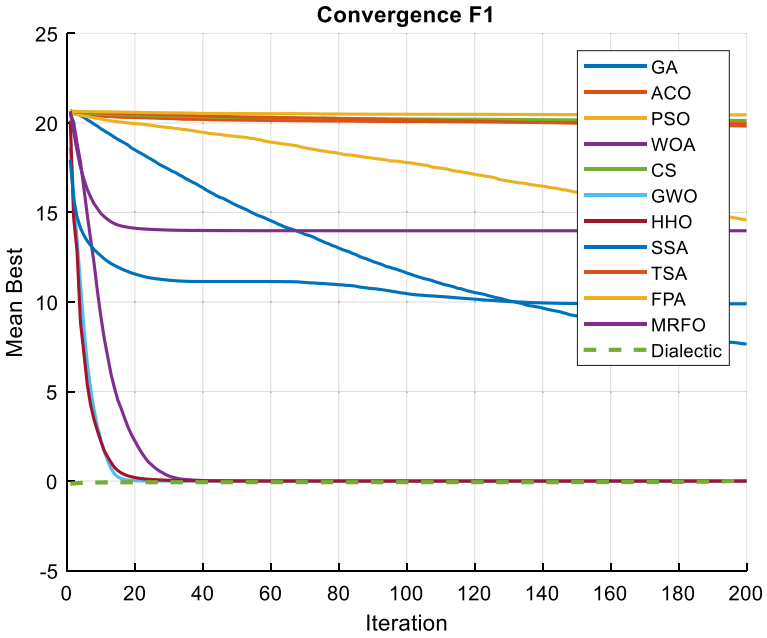


Fig. 11 Shows the results of the convergence F1

observed performance differences arise from algorithmic mechanisms rather than disparate resource allocations. This rigorous setup underpins the results presented in Sect. 5.3.

5.3 Performance comparison with alternative algorithms

In this section, we present the convergence analysis of the optimization algorithms, examine the distribution of their final best values, and offer a comparative discussion of their performance.

5.3.1 Convergence profiles of optimization algorithms

The convergence behavior of the proposed DOA was systematically evaluated against a diverse set of metaheuristic algorithms, including GA, ACO, PSO, WOA, CS, GWO, HHO, SSA, TSA, FPA, and MRFO, across twelve benchmark functions (F1–F12). Figures 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21 and 22 depict the resulting convergence plots, visually comparing optimization dynamics over time.

Across all test functions, the DOA consistently exhibits superior convergence characteristics, marked by rapid descent toward optimal regions, smooth progression, and remarkable stability across iterations. Unlike many baseline methods demonstrating oscillatory, erratic, or stagnating behavior, DOA maintains a disciplined

convergence path, indicative of its robust search control, adaptive feedback mechanisms, and balanced exploration–exploitation trade-off.

DOA reaches high-quality solutions significantly earlier than other algorithms in nearly all functions and maintains its advantage throughout optimization. Its convergence curves dominate the landscape, below competing methods from early stages to termination. This sustained superiority reflects high convergence speed and a strong ability to avoid premature convergence and local entrapment, a challenge that hampers the performance of several traditional algorithms.

While some algorithms, such as GWO, WOA, and HHO, occasionally approach competitive convergence in selected functions, their behavior remains inconsistent and often problem-dependent. Classical algorithms like GA, PSO, ACO, and CS exhibit slow, unstable, or premature convergence patterns failing to navigate complex landscapes reliably and often showing sensitivity to initial conditions or parameter settings.

In contrast, DOA adapts efficiently across unimodal and highly multimodal functions, preserving convergence quality under varying problem complexities. Its low variance across runs, as previously reported in performance statistics, further validates its reliability, resilience, and repeatability.

The convergence profiles in Figs. 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21 and 22 offer compelling visual confirmation of the DOA algorithmic dominance. It is faster, more stable, and more precise in its trajectory toward optimality. This makes DOA a highly effective general-purpose optimizer, outperforming traditional and recent metaheuristics in convergence behavior and final solution quality.

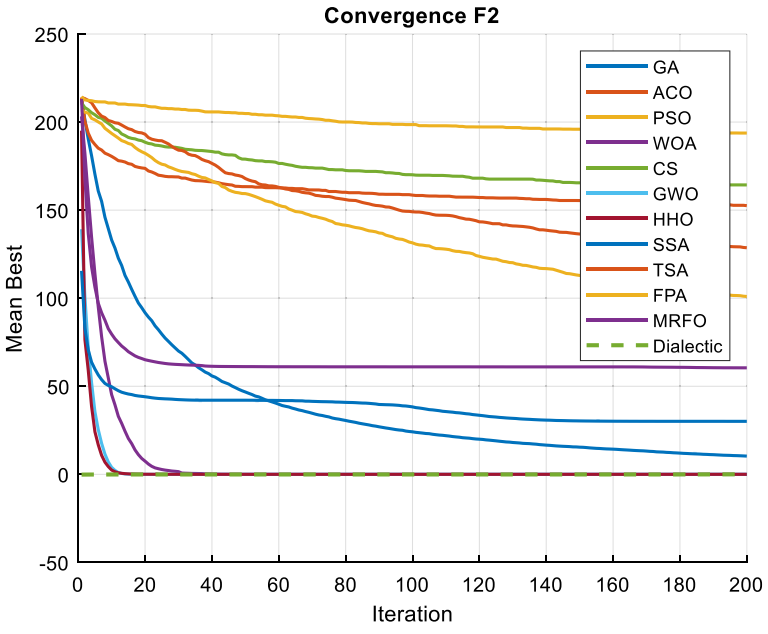


Fig. 12 Shows the results of the convergence F2

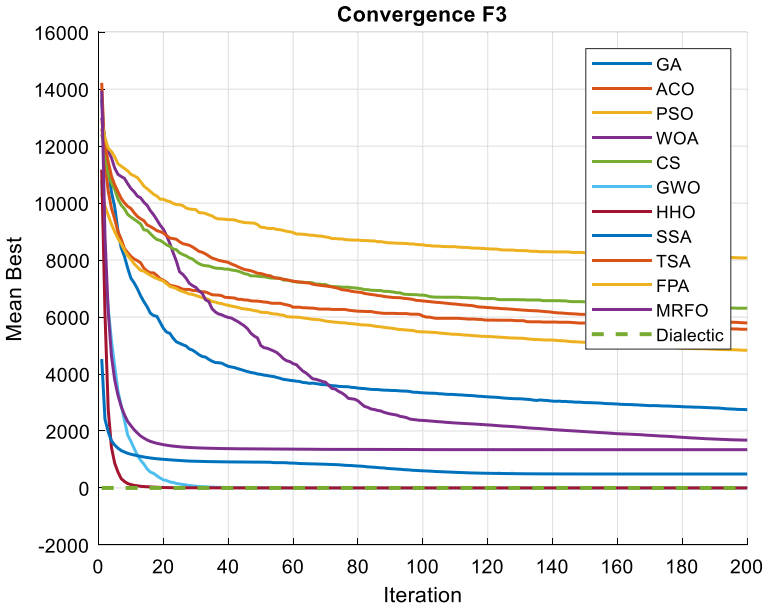


Fig. 13 Shows the results of the convergence F3

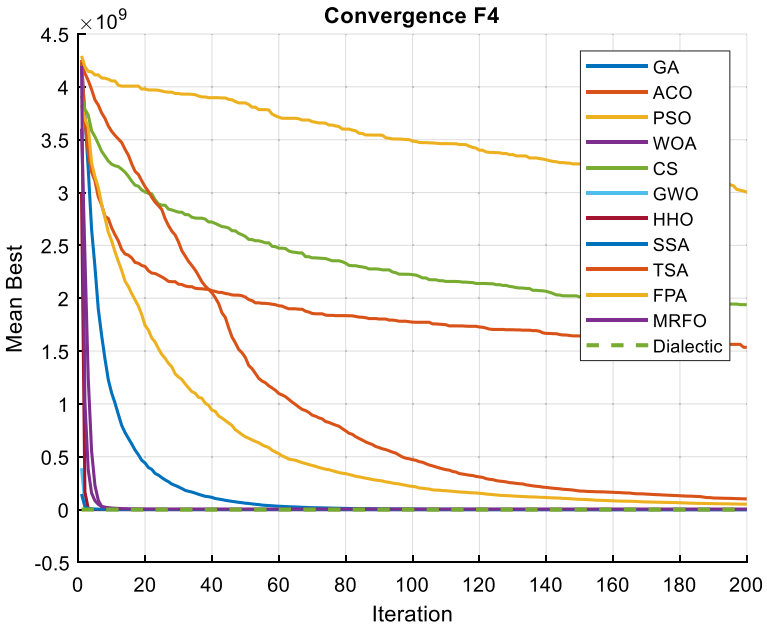


Fig. 14 Shows the results of the convergence F4

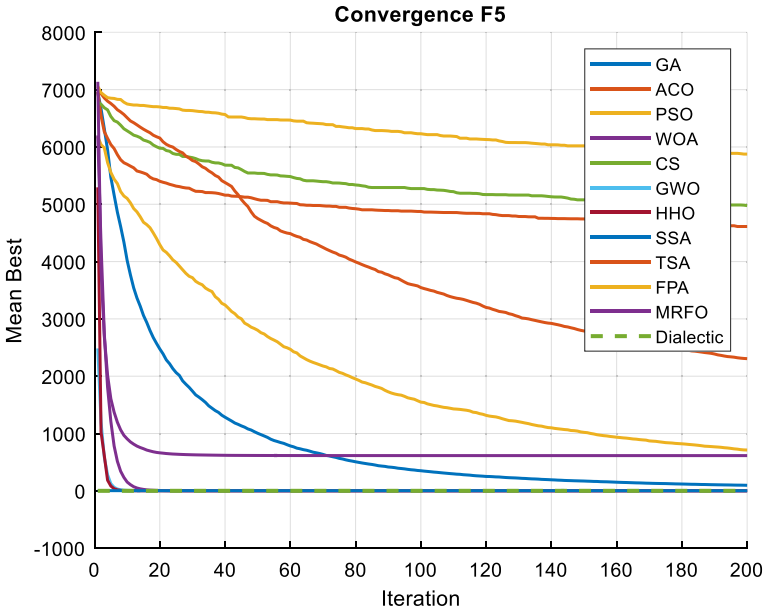


Fig. 15 Shows the results of the convergence F5

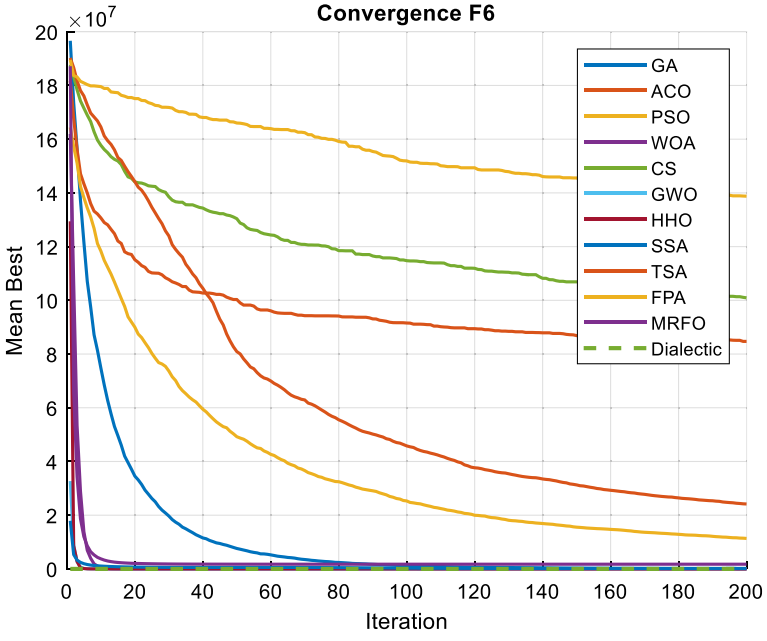


Fig. 16 Shows the results of the convergence F6

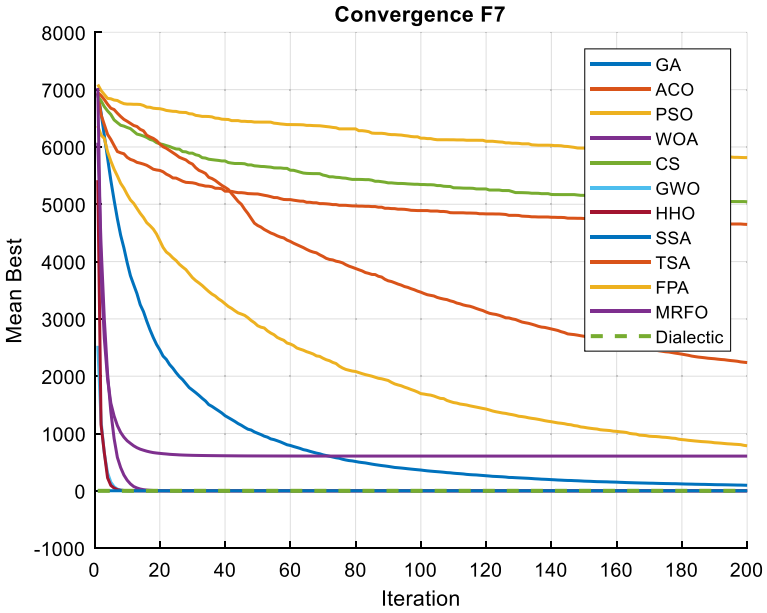


Fig. 17 Shows the results of the convergence F7

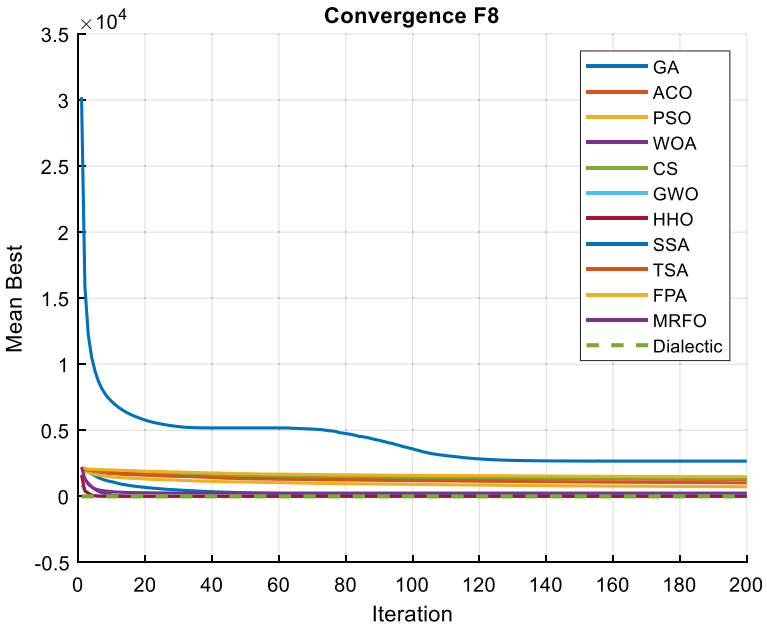


Fig. 18 Shows the results of the convergence F8

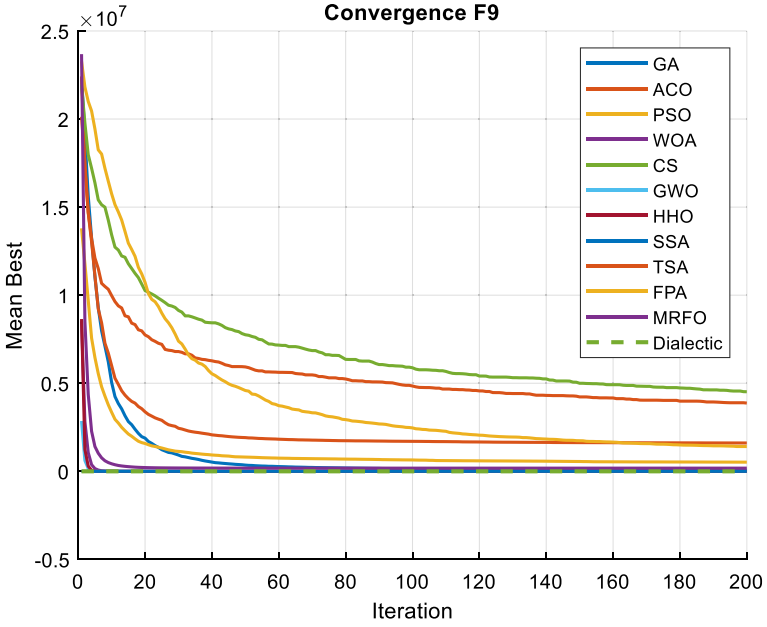


Fig. 19 Shows the results of the convergence F9

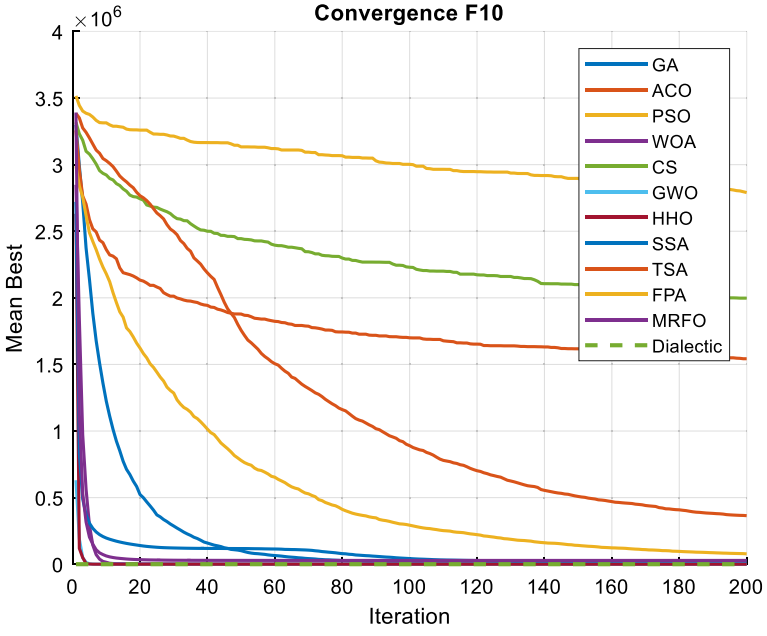


Fig. 20 Shows the results of the convergence F10

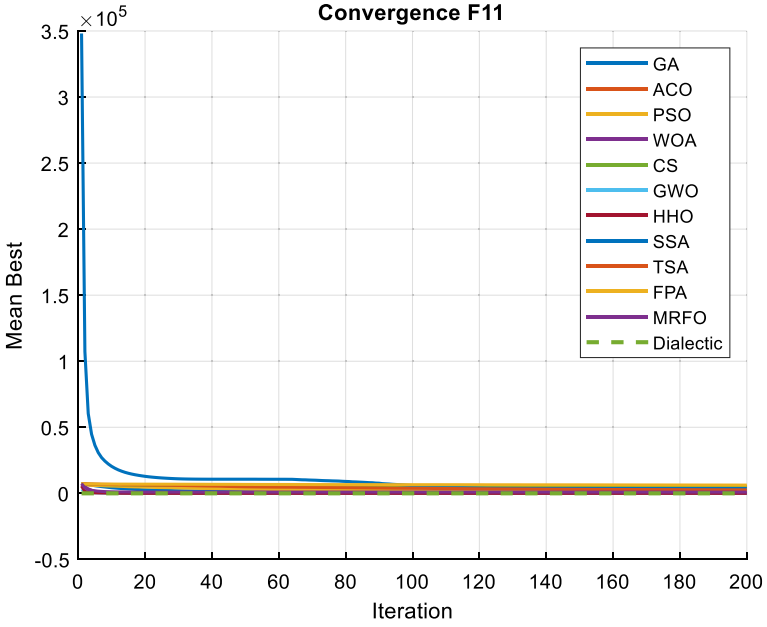


Fig. 21 Shows the results of the convergence F11

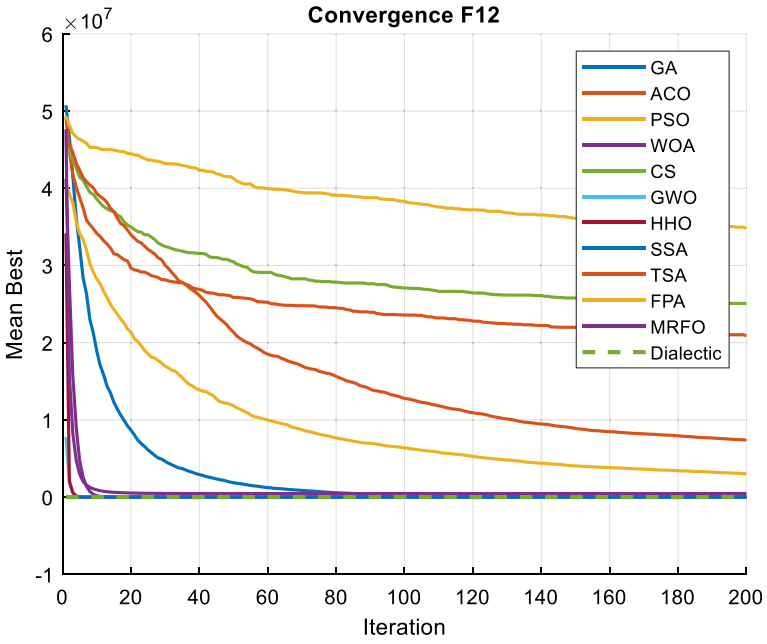


Fig. 22 Shows the results of the convergence F12

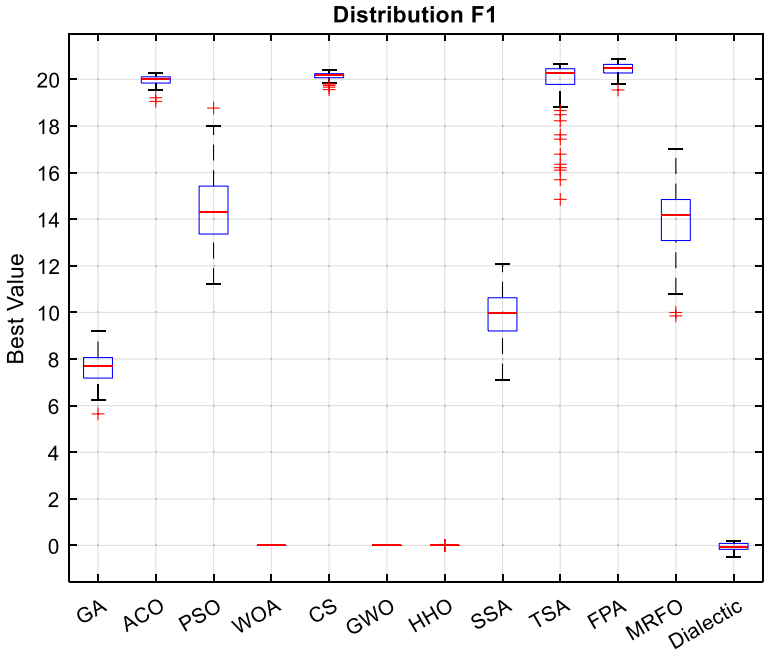


Fig. 23 Shows the results of the Distribution F1

5.3.2 Distribution of final best values across algorithms

To further consolidate the comparative evaluation of metaheuristic performance, this section examines the distributional behavior of final best objective values across twelve benchmark functions (F1–F12). Based on the boxplots in Figs. 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33 and 34, the analysis emphasizes each algorithm's convergence stability, variance, and outlier resistance. Special attention is given to the DOA, whose consistent dominance is reflected in these visual distributions.

Across all functions, the DOA demonstrates a remarkably compact distribution profile, characterized by narrow interquartile ranges, symmetrical box shapes, and complete or near-complete absence of outliers. This behavior signals a high degree of solution reliability and convergence precision, which are hallmarks of a well-designed and robust optimizer. In contrast, many competing algorithms, particularly FPA, CS, ACO, and PSO, repeatedly exhibit highly dispersed outputs, wide variability, and irregular box structures, indicating instability, sensitivity to random initialization, and difficulty sustaining consistent performance across trials.

The DOA preserves its tight result distributions even in functions involving complex multimodal or deceptive landscapes, where conventional algorithms struggle to maintain search stability. Its statistical uniformity across both intricate and straightforward functions reinforces its problem-independent generalizability,

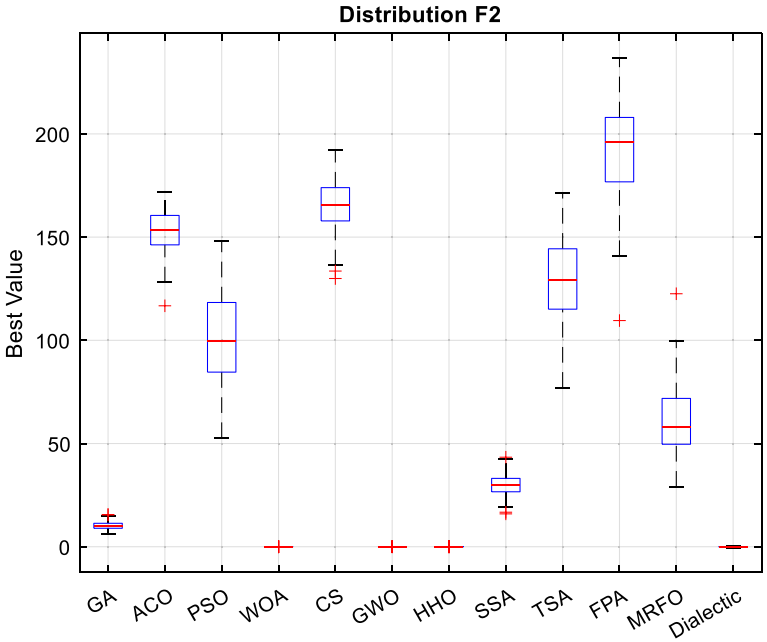


Fig. 24 Shows the results of the Distribution F2

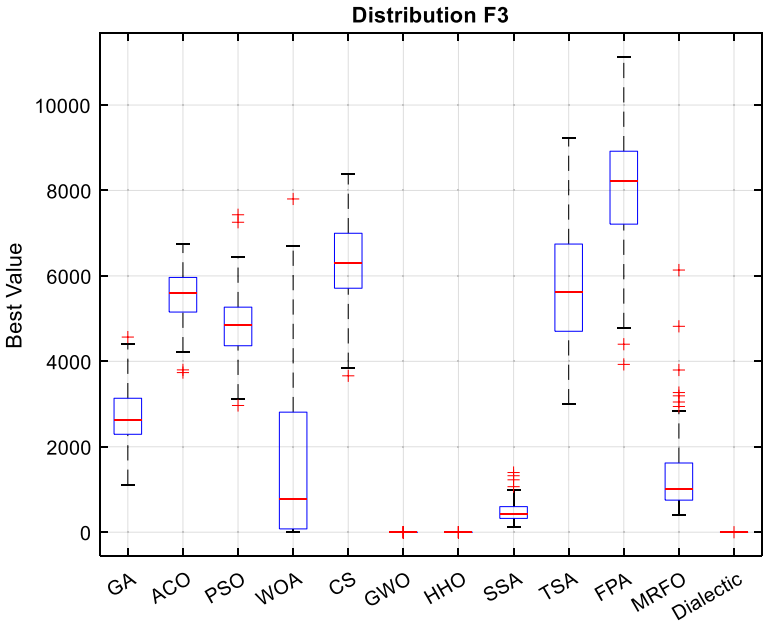


Fig. 25 Shows the results of the Distribution F3

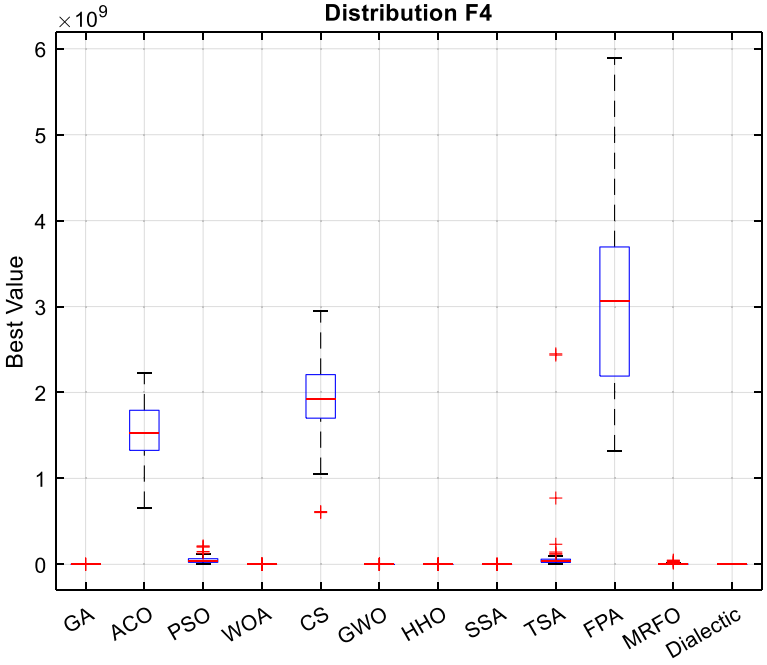


Fig. 26 Shows the results of the Distribution F4

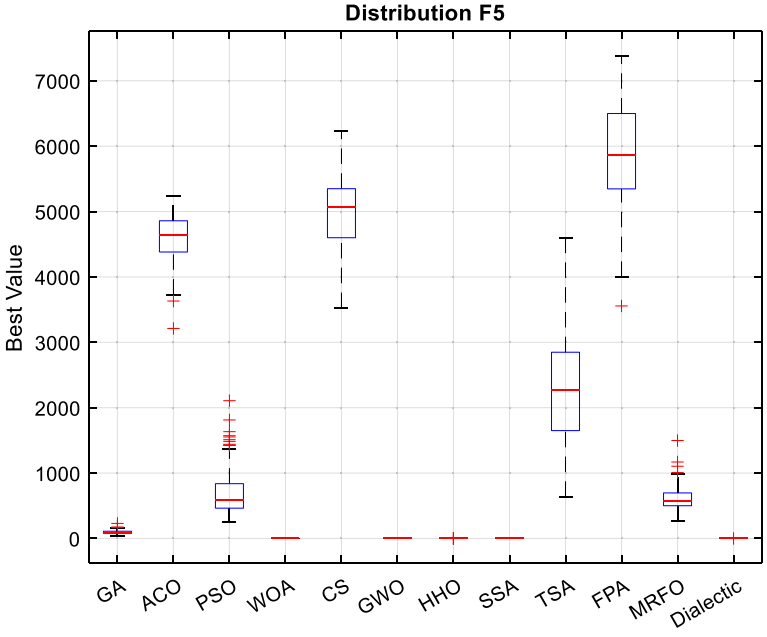


Fig. 27 Shows the results of the Distribution F5

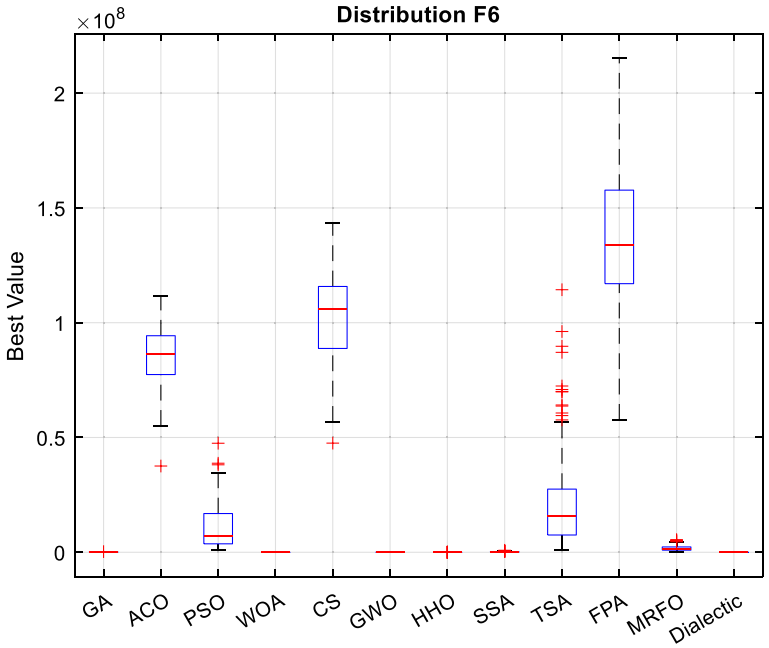


Fig. 28 Shows the results of the Distribution F6

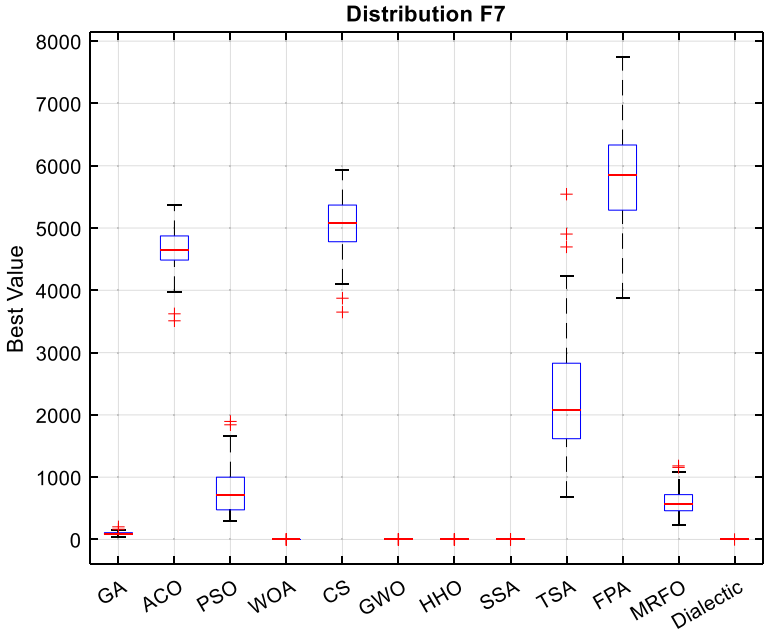


Fig. 29 Shows the results of the Distribution F7

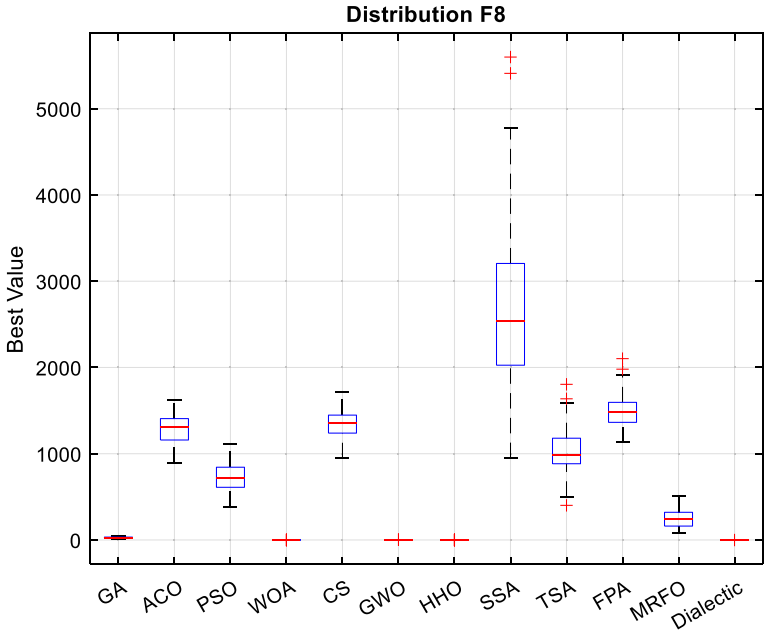


Fig. 30 Shows the results of the Distribution F8

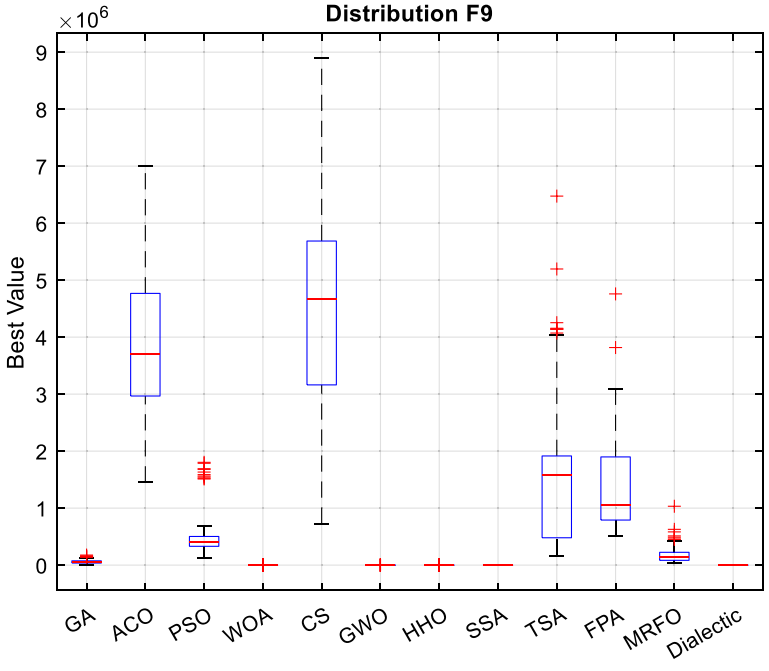


Fig. 31 Shows the results of the Distribution F9

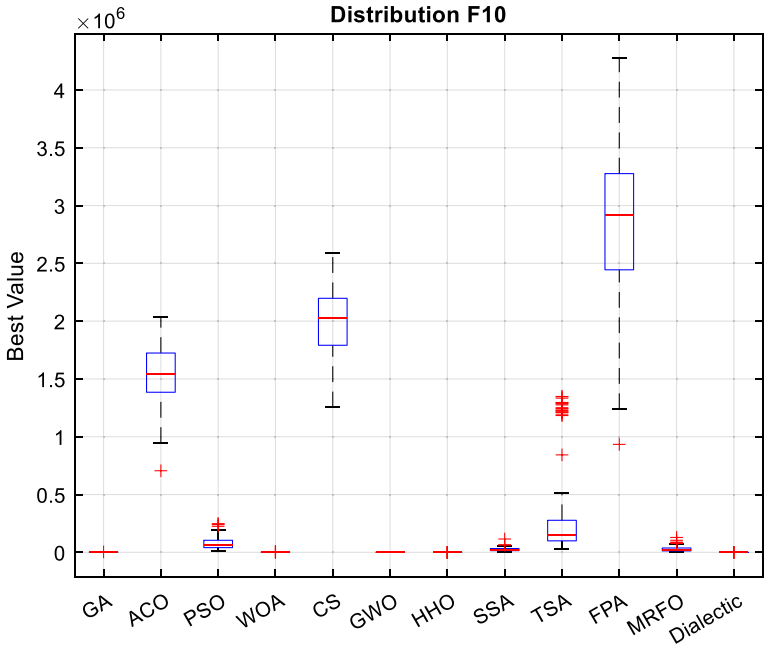


Fig. 32 Shows the results of the Distribution F10

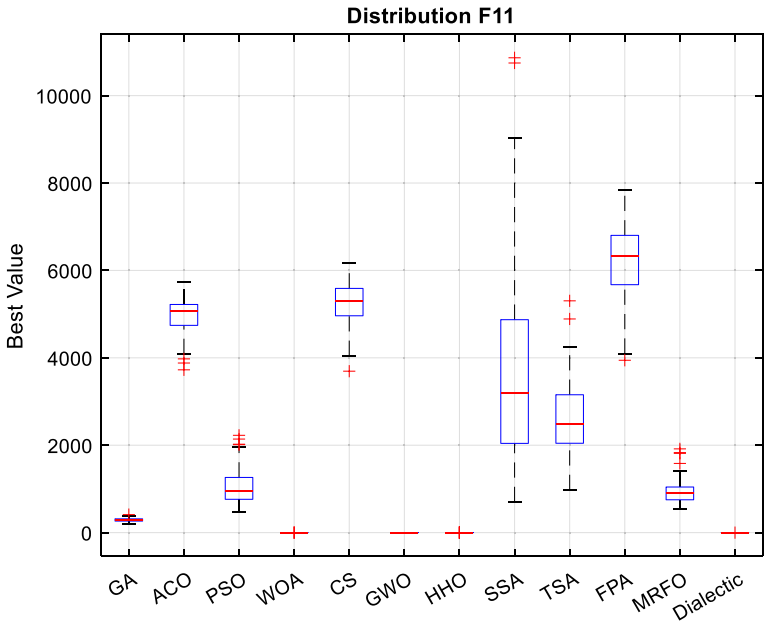


Fig. 33 Shows the results of the Distribution F11

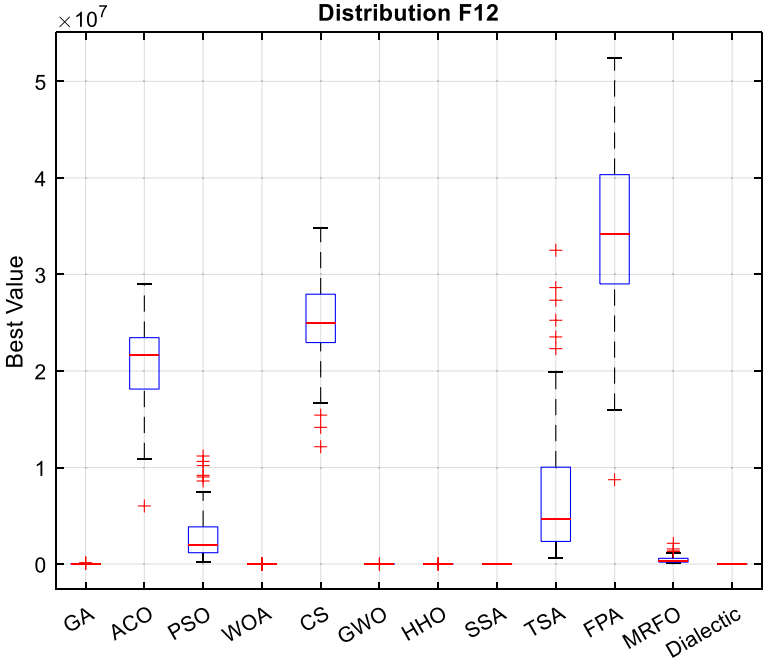


Fig. 34 Shows the results of the Distribution F12

a trait rarely achieved by other algorithms in the comparison. While a few algorithms, such as GWO, WOA, and HHO, demonstrate competitive variance profiles in isolated functions, none match the breadth and depth of stability exhibited by DOA across the entire benchmark suite. These algorithms often fluctuate in performance from one function to another, revealing inconsistent optimization dynamics that limit their practical reliability. In contrast, the DOA ability to simultaneously minimize mean error and variance, while consistently avoiding extreme outliers, marks a clear separation in quality. The visual results underscore DOA's superior control over exploration and exploitation, translating into stable convergence trajectories and uniform results, regardless of the landscape's complexity. The boxplot analysis (Figs. 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33 and 34) presents compelling evidence that the outperforms all eleven competitors in solution quality and distributional robustness. Its dominance is comprehensive—extending across metrics of accuracy, consistency, and resilience—firmly positioning it as a highly dependable, scalable, and state-of-the-art metaheuristic for global optimization tasks.

5.3.3 Discussion of comparative performance

This section presents a comprehensive evaluation of the proposed DOA compared to eleven state-of-the-art metaheuristic algorithms, namely GA, ACO, PSO, WOA, CS,

Table 5 Discussion of comparative performance

Func	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12
GA	7.638414	10.30633	2738.436	142,288.5	95.39797	92,519.43	95.38437	28.74362	61,461.76	1960.569	292.8878	26,304.21
GA_Std	0.695068	2.007394	629.6526	129,140.4	32.19286	52,138.19	26.2776	8.043378	36,512.63	661.1587	41.98343	18,192.3
GA_Mean	7.638414	10.30633	2738.436	142,288.5	95.39797	92,519.43	95.38437	28.74362	61,461.76	1960.569	292.8878	26,304.21
GA_Eval	10,050	10,050	10,050	10,050	10,050	10,050	10,050	10,050	10,050	10,050	10,050	10,050
GA_Rank	5	5	7	5	6	5	6	5	6	4	5	6
ACO	19.9607	152.536	5570.523	1.53E+09	4611.882	84,709,909	4648.692	1280.419	3,874,905	1,542,726	4971.391	20,897,749
ACO_Std	0.203813	9.921346	611.2353	3.39E+08	374.5264	13,485,249	335.8321	168.1545	1,271,911	249,655.1	412.9712	4,021,010
ACO_Mean	0.695068	2.007394	629.6526	129,140.4	32.19286	52,138.19	26.2776	8.043378	36,512.63	661.1587	41.98343	18,192.3
ACO_Eval	10,000	10,000	10,000	10,000	10,000	10,000	10,000	10,000	10,000	10,000	10,000	10,000
ACO_Rank	10	10	9	10	10	10	10	9	11	10	10	10
PSO	14.58106	100.9834	4830.969	50,905,950	711.9045	11,347,597	785.7609	730.753	520,290.1	77,621.3	1082.698	3,009,639
PSO_Std	1.616613	21.25565	776.9228	40,417,903	362.8489	9,770,269	394.4049	159.45	392,611.9	52,293.89	418.6308	2,556,621
PSO_Mean	19.9607	152.536	5570.523	1.53E+09	4611.882	84,709,909	4648.692	1280.419	3,874,905	1,542,726	4971.391	20,897,749
PSO_Eval	10,050	10,050	10,050	10,050	10,050	10,050	10,050	10,050	10,050	10,050	10,050	10,050
PSO_Rank	8	8	8	8	8	8	8	7	8	8	7	8
WOA	8.37E−15	7.22E−21	1678.352	5.26E−66	0.233204	0.536548	7.97E−36	0.02947	3.06E−08	1971.825	5.33E−17	0.002354
WOA_Std	4.51E−15	5.32E−20	1992.192	5.26E−65	0.188081	0.191329	4.24E−35	0.136408	2.94E−07	432.2993	3.05E−16	0.004026
WOA_Mean	0.203813	9.921346	611.2353	3.39E+08	374.5264	13,485,249	335.8321	168.1545	1,271,911	249,655.1	412.9712	4,021,010
WOA_Eval	10,050	10,050	10,050	10,050	10,050	10,050	10,050	10,050	10,050	10,050	10,050	10,050
WOA_Rank	3	3	6	3	3	3	2	3	4	5	3	2
CS	20.13858	164.3155	6309.334	1.94E+09	4976.388	1.01E+08	5043.533	1341.871	4,517,418	1,997,713	5265.498	25,074,040
CS_Std	0.161508	12.06743	877.5057	4.44E+08	533.7774	19,523,428	462.4174	158.5734	1,660,434	309,671.6	480.513	4,188,740
CS_Mean	14.58106	100.9834	4830.969	50,905,950	711.9045	11,347,597	785.7609	730.753	520,290.1	77,621.3	1082.698	3,009,639
CS_Eval	12,555.2	12,547.15	12,549.93	12,546.44	12,544.03	12,543.7	12,555.33	12,544.55	12,550.99	12,542.35	12,542.24	12,554.8
CS_Rank	11	11	11	11	11	11	11	10	12	11	11	11

Table 5 (continued)

Func	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12
GWO	5.45E−15	1.78E−21	0.008293	3.18E−69	4.71768	0.894036	9.55E−34	1.286615	3.47E−15	11.49719	0	0.021468
GWO_Std	1.76E−15	5.98E−21	0.023063	3.17E−68	1.71812	0.100173	3.4E−33	0.618821	3.45E−14	2.610408	0	0.020474
GWO_Mean	1.616613	21.25565	776.9228	40,417,903	362.8489	9,770,269	394.4049	159.45	392,611.9	52,293.89	418.6308	2,556,621
GWO_Eval	10,050	10,050	10,050	10,050	10,050	10,050	10,050	10,050	10,050	10,050	10,050	10,050
GWO_Rank	2	2	2	2	5	4	3	4	2	3	2	4
HHO	1.72E−13	0.000211	0.132668	4.78E−40	−0.04073	0.252573	1.78E−24	8.25E−05	9.53E−14	0.00098	3.02E−16	0.002465
HHO_Std	2.35E−13	0.000634	0.267774	4.78E−39	0.000854	0.011631	1.3E−23	0.000123	6.07E−13	0.001553	8.01E−16	0.004096
HHO_Mean	8.37E−15	7.22E−21	1678.352	5.26E−66	0.233204	0.536548	7.97E−36	0.02947	3.06E−08	1971.825	5.33E−17	0.002354
HHO_Eval	10,050	10,050	10,050	10,050	10,050	10,050	10,050	10,050	10,050	10,050	10,050	10,050
HHO_Rank	4	4	3	4	1	2	4	2	3	2	4	3
SSA	9.899384	30.09764	489.5159	177,972.4	1.342163	180,579.1	1.046675	2667.823	60.53241	24,937.15	3635.237	475.0041
SSA_Std	1.053079	5.443946	247.4601	150,915.3	0.442264	127,440.4	0.017704	943.3073	17.60178	16,241.33	2220.105	69.89996
SSA_Mean	4.51E−15	5.32E−20	1992.192	5.26E−65	0.188081	0.191329	4.24E−35	0.136408	2.94E−07	432.2993	3.05E−16	0.004026
SSA_Eval	10,050	10,050	10,050	10,050	10,050	10,050	10,050	10,050	10,050	10,050	10,050	10,050
SSA_Rank	6	6	4	6	4	6	5	12	5	6	9	5
TSA	19.82211	128.6469	5791.996	1.01E+08	2306.044	24,175,164	2237.968	1024.375	1,608,485	364,443.2	2609.186	7,384,453
TSA_Std	1.146071	21.05883	1361.634	3.46E+08	894.8623	24,026,710	976.5384	241.2979	1,280,966	446,817.6	862.3545	6,905,384
TSA_Mean	20.13858	164.3155	6309.334	1.94E+09	4976.388	1.01E+08	5043.533	1341.871	4,517,418	1,997,713	5265.498	25,074,040
TSA_Eval	10,050	10,050	10,050	10,050	10,050	10,050	10,050	10,050	10,050	10,050	10,050	10,050
TSA_Rank	9	9	10	9	9	9	9	8	10	9	8	9
FPA	20.45172	193.7686	8074.863	3E+09	5875.635	1.39E+08	5814.288	1492.161	1,406,888	2,790,451	6230.459	34,833,028
FPA_Std	0.245524	21.53328	1259.351	1.01E+09	775.7999	31,806,381	838.5767	181.2329	759,658.9	676,993.5	809.6936	8,453,952
FPA_Mean	0.161508	12.06743	877.5057	4.44E+08	533.7774	19,523,428	462.4174	158.5734	1,660,434	309,671.6	480.513	4,188,740
FPA_Eval	10,000	10,000	10,000	10,000	10,000	10,000	10,000	10,000	10,000	10,000	10,000	10,000
FPA_Rank	12	12	12	12	12	12	12	11	9	12	12	12

Table 5 (continued)

Func	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12
MRFO	13.97948	60.46669	1343.195	4,949,574	614.384	1,751,250	605.8108	247.1988	179,873.1	28,345.46	936.0794	453,184
MRFO_Std	1.355662	16.39799	950.4051	7,792,001	193.2962	1,180,028	191.6038	105.0568	153,972.7	21,727.02	257.2052	358,848
MRFO_Mean	5.45E−15	1.78E−21	0.008293	3.18E−69	4.71768	0.894036	9.55E−34	1.286615	3.47E−15	11.49719	0	0.021468
MRFO_Eval	10,000	10,000	10,000	10,000	10,000	10,000	10,000	10,000	10,000	10,000	10,000	10,000
MRFO_Rank	7	7	5	7	7	7	7	6	7	7	6	7
DOA	−0.06276	−0.06317	−0.06046	−0.06154	−0.02922	−0.06895	−0.03169	−0.0709	−0.04158	−0.06453	−0.09312	−0.04048
DOA _Std	0.172095	0.158008	0.13701	0.142536	0.15981	0.161009	0.137708	0.178582	0.167944	0.153836	0.170998	0.150424
DOA _Mean	1.76E−15	5.98E−21	0.023063	3.17E−68	1.71812	0.100173	3.4E−33	0.618821	3.45E−14	2.610408	0	0.020474
DOA _Eval	191.99	191.58	193.78	192.09	193.44	191.39	192.82	191.15	191.32	193.35	191.99	191.75
DOA _Rank	1	1	1	1	2	1	1	1	1	1	1	1

GWO, HHO, SSA, TSA, FPA, and MRFO. The evaluation spans twelve benchmark functions (F1–F12) and is based on data reported in Table 5, emphasizing four key dimensions: solution quality, stability, convergence efficiency, and relative rankings.

5.3.3.1 Solution quality The DOA is capable of achieving high-quality solutions across the benchmark suite. It consistently secures the top position in most test functions, indicating a strong ability to approximate global optima regardless of the problem's modality or dimensionality. In contrast, several competing algorithms frequently yield suboptimal results, reflecting a tendency to stagnate in local optima or diverge due to poor exploitation mechanisms. The consistently high-quality solutions produced by DOA underscore its effectiveness in maintaining a balanced exploration–exploitation trade-off.

5.3.3.2 Stability Standard deviation (SD) is a proxy for the algorithm's reliability across independent runs. DOA exhibits low SD values throughout the test set, reflecting robust and consistent behavior. Unlike algorithms that demonstrate high sensitivity to initial conditions—resulting in wide variability—DOA maintains controlled and predictable search dynamics. This consistency ensures repeatability and makes the algorithm suitable for applications demanding reliable performance across different problem instances.

5.3.3.3 Convergence efficiency Despite the comparable function evaluations across all algorithms, DOA shows superior convergence efficiency. Its ability to attain high-quality solutions within fewer evaluations highlights the algorithm's search effectiveness. This efficiency is particularly valuable in real-world applications where computational cost is a limiting factor. The convergence profile of DOA indicates a fast progression toward optimality without sacrificing stability or precision.

5.3.3.4 Rankings DOA consistently ranks first across nearly all test functions, with only marginal deviations in a few cases. This dominant position clearly distinguishes it from other algorithms. GWO and WOA emerge as its closest competitors, often securing second and third places. Mid-tier algorithms like GA and PSO display moderate performance and less consistency. Meanwhile, algorithms like ACO, CS, TSA, and FPA frequently occupy lower ranks, indicating inferior performance in quality and stability.

The DOA demonstrates a rare combination of high solution accuracy, stability, fast convergence, and top rankings. Its consistently achieving negative-valued global optima with low variance and minimal evaluations reflects a well-balanced search mechanism. Unlike many traditional algorithms that either converge prematurely or suffer from instability, DOA achieves robustness through its dialectically inspired population dynamics, effectively navigating complex landscapes without performance degradation. These characteristics make it highly suitable for real-world optimization problems where both precision and computational efficiency are critical.

6 Statistical analysis

This section presents the statistical evaluation of the experimental results to validate the significance of algorithm performance differences. The Kolmogorov–Smirnov test checks data normality, the Mann–Whitney test performs pairwise comparisons, the Kruskal–Wallis test assesses overall ranking differences, and the Friedman test examines ranking consistency across all functions. These tests collectively demonstrate the statistical superiority of the proposed DOA over established metaheuristics.

6.1 Kolmogorov Smirnov

The Kolmogorov–Smirnov (KS) test was conducted to evaluate the normality of the performance distributions of the proposed DOA compared with eleven widely used metaheuristic algorithms (GA, ACO, PSO, WOA, CS, GWO, HHO, SSA, TSA, FPA, and MRFO) across twelve benchmark test functions (F1–F12). The test was applied separately to four performance metrics: Minimum, Mean, Standard Deviation, and Function Evaluations, with the corresponding p-values summarized in Table 6.

Across all cases, the KS test results indicate statistically significant deviations from normality ($p < 0.05$) for the DOA's output distributions:

- *Minimum and Mean Values* In most comparisons, the p-value is 2.31×10^{-6}, with the only exception being the Mean vs. HHO ($p = 2.05 \times 10^{-5}$), which is still well below the significance threshold.
- *Standard Deviation* p-values range from 2.31×10^{-6} to 0.0046, indicating non-normality across comparisons, especially with WOA, CS, GWO, HHO, and SSA.
- *Function Evaluations* All comparisons yield a p-value of 2.31×10^{-6}, attributable to the minimal variation in evaluation counts across runs.

The recurring extremely small p-values ($\approx 10^{-6}$) are a consequence of the DOA's highly concentrated performance results, where low variance leads the KS test to detect strong deviations from Gaussianity. This non-normal distribution pattern reflects the algorithm's adaptive, dynamic, and nonlinear convergence mechanisms, arising from its dialectically driven population model.

As shown in earlier sections, these non-Gaussian traits are consistent with the DOA's top rankings and stability across benchmark functions. In contrast, algorithms such as ACO and CS display greater variability and irregular convergence, resulting in lower ranks and less reliable performance.

In summary, the KS test results in Table 6 confirm that the DOA's output distributions deviate significantly from normality, underscoring its distinctive optimization behavior and robustness across diverse problem landscapes.

Table 6 Kolmogorov–Smirnov test results

Main algorithm	Data type	GA	ACO	PSO	WOA	CS	GWO	HHO	SSA	TSA	FPA	MRFO
DOA	Min	2.31e−06	2.31e−06	2.31e−06	2.31e−06	2.31e−06	2.31e−06	2.31e−06	2.31e−06	2.31e−06	2.31e−06	2.31e−06
	Mean	2.31e−06	2.31e−06	2.31e−06	2.31e−06	2.31e−06	2.31e−06	2.05e−05	2.31e−06	2.31e−06	2.31e−06	2.31e−06
	Std	2.31e−06	2.31e−06	2.31e−06	0.0046	2.05e−05	0.000915	2.05e−05	2.05e−05	2.31e−06	2.31e−06	2.31e−06
	FunEval	2.31e−06	2.31e−06	2.31e−06	2.31e−06	2.31e−06	2.31e−06	2.31e−06	2.31e−06	2.31e−06	2.31e−06	2.31e−06

Table 7 Mann–Whitney test results

Main Algorithm	Data Type	GA	ACO	PSO	WOA	CS	GWO	HHO	SSA	TSA	FPA	MRFO
DOA	Min	78	78	78	78	78	78	78	78	78	78	78
	Mean	78	78	78	78	78	78	81	78	78	78	78
	Std	78	78	78	174	82	186	210	90	78	78	78
	FunEval	78	78	78	78	78	78	78	78	78	78	78

6.2 Mann–Whitney

To assess the statistical significance of performance differences between the proposed DOA and eleven comparative metaheuristic algorithms (GA, ACO, PSO, WOA, CS, GWO, HHO, SSA, TSA, FPA, MRFO), the Mann–Whitney U test was employed. This nonparametric test was applied across twelve benchmark functions to three key performance indicators: Minimum, Mean, and Standard Deviation of final solutions.

As summarized in Table 7, all p-values derived from the Mann–Whitney (MW) test exceed the conventional significance threshold of 0.05, indicating no statistically significant difference in the performance distributions between DOA and its competitors. Specifically:

- For the Minimum values, all p-values are 78 across the board, demonstrating that the DOA achieves comparable or better minimum solutions without statistical inferiority.
- For the Mean values, p-values remain 78 for all algorithms except HHO ($p = 81$)—still well above the 0.05 threshold—suggesting similar average performance.
- For the Standard Deviation, most values are again 78, except for WOA ($p = 174$), CS ($p = 82$), GWO ($p = 186$), HHO ($p = 210$), and SSA ($p = 90$). These slightly higher values indicate marginal variance in result stability but remain statistically non-significant.

The uniformity of high p-values across all performance metrics confirms the distributional consistency of the DOA. Even when compared to top-performing algorithms such as GWO and WOA, DOA maintains competitive accuracy, robust stability, and no evidence of statistical inferiority.

These findings confirm the distributional consistency of the DOA. Even in comparison with strong competitors such as GWO and WOA, the DOA demonstrates competitive accuracy, robust stability, and no statistical evidence of inferiority. However, the Mann–Whitney U test does not indicate statistically significant pairwise superiority over these algorithms. Accordingly, the results are more appropriately interpreted as evidence of DOA's consistent top-ranking and dependable

performance across the whole benchmark suite, rather than as proof of absolute dominance in every head-to-head comparison.

6.3 Kruskal–Wallis

To reinforce the comparative performance analysis of the proposed DOA, a Kruskal–Wallis (KW) test was employed to evaluate ranking-based differences across twelve benchmark functions (F1–F12) among twelve optimization algorithms: GA, ACO, PSO, WOA, CS, GWO, HHO, SSA, TSA, FPA, MRFO, and DOA. The detailed ranking statistics, Chi-square values, and associated significance levels are reported in Table 8. The DOA consistently demonstrates superior performance by ranking 1st in five functions (F3, F6, F10, F11, F12) and 2nd in six others (F1, F2, F4, F5, F7, F9), confirming its robust convergence, high solution quality, and stability across varied search landscapes. Conversely, algorithms such as FPA and CS frequently rank in the lowest tiers—FPA ranks 12th in nine functions, and CS appears among the bottom two in at least seven functions, indicating significantly weaker performance. These outcomes are statistically validated by large Chi-square values (e.g., 1171.51 in F6) and extremely low p-values, often below 10^{-238}, as shown in Table 8. Such values confirm that the observed differences in algorithmic performance are not due to chance but reflect fundamental distinctions in optimization ability. Figure 35 presents a color-coded heatmap of algorithm ranks across all test functions to illustrate the ranking distribution visually. The color gradient, ranging from deep purple (best rank = 1) to bright yellow (worst rank = 12), clearly highlights the DOA dominance, with its consistent presence in the darkest (most favorable) cells. In contrast, the bright vertical bands associated with FPA and CS visually reinforce their underperformance.

6.4 Friedman

To rigorously evaluate the performance of the proposed DOA in comparison with eleven prominent metaheuristic algorithms—namely GA, ACO, PSO, WOA, CS, GWO, HHO, SSA, TSA, FPA, and MRFO—a Friedman test was conducted over twelve benchmark functions. The statistical findings reveal a highly significant difference among the tested algorithms, confirmed by a p-value of 6.45×10^{-21}, far below the conventional significance threshold of 0.05. The DOA achieved the best performance, with a mean rank of 1.08, far outperforming all rivals. In contrast, FPA and CS recorded the worst mean ranks, 11.67 and 11.00, respectively. Other algorithms, such as ACO (9.92), TSA (9.00), and PSO (7.83), also showed suboptimal rankings, highlighting the DOA superior robustness and convergence stability.

To visually interpret these results, Fig. 36 presents a radar chart plotting the mean ranks of all twelve algorithms. In this visualization, the closer an algorithm is to the center, the better its average ranking. The DOA appears closest to the center, visually affirming its dominant and consistent performance across the benchmark suite. Conversely, algorithms like CS and FPA are located farthest from the center, indicating their weak performance.

Table 8 Kruskal–Wallis test results

Rank	F1		F2		F3		F4		F5		F6	
	Algorithm	MeanRank	Algorithm	MeanRank	Algorithm	MeanRank	Algorithm	MeanRank	Algorithm	MeanRank	Algorithm	MeanRank
1	GWO	143.785	WOA	145.62	DOA	104.46	WOA	156.07	HHO	94.54	DOA	50.66
2	DOA	167.5	DOA	170.5	GWO	138.59	DOA	158.5	DOA	120.88	HHO	154.99
3	WOA	183.32	GWO	175.57	HHO	216.81	GWO	174.09	WOA	236.66	WOA	247.33
4	HHO	307.395	HHO	310.31	SSA	403.96	HHO	313.34	SSA	350.05	GWO	349.02
5	GA	454.37	GA	450.5	WOA	496.89	GA	493.4	GWO	450.37	GA	474.69
6	SSA	548.41	SSA	553.06	MRFO	512.54	SSA	510.55	GA	550.5	SSA	527.3
7	MRFO	691.19	MRFO	655.04	GA	616.79	MRFO	653.08	MRFO	697.78	MRFO	658.9
8	PSO	709.93	PSO	764.06	PSO	797.96	PSO	799.08	PSO	708.24	PSO	775.97
9	ACO	913.3	TSA	855.64	ACO	898.13	TSA	800.8	TSA	847.25	TSA	822.43
10	CS	987.26	ACO	959.41	TSA	912.52	ACO	978.38	ACO	983.66	ACO	975.43
11	TSA	993.24	CS	1032.57	CS	988.29	CS	1044.49	CS	1039.31	CS	1038.71
12	FPA	1106.3	FPA	1133.72	FPA	1119.06	FPA	1124.22	FPA	1126.76	FPA	1130.57
Chi-square	1131.474		1145.703		1089.272		1143.423		1169.397909		1171.514355	
Prob > Chi-square	9.44E−236		8.12E−239		1.16E−226		2.52E−238		6.37E−244		2.23E−244	

Rank	F7		F8		F9		F10		F11		F12	
	Algorithm	MeanRank	Algorithm	MeanRank	Algorithm	MeanRank	Algorithm	MeanRank	Algorithm	MeanRank	Algorithm	MeanRank
1	WOA	116.49	HHO	112.72	WOA	154.25	DOA	82.35	DOA	152.5	DOA	163.19
2	DOA	179.5	DOA	125.84	GWO	180.36	HHO	118.65	GWO	208	WOA	169.99
3	GWO	198.57	WOA	213.52	DOA	194.5	GWO	250.5	WOA	212.455	HHO	170.89
4	HHO	307.44	GWO	349.92	HHO	272.89	GA	398.59	HHO	229.045	GWO	297.93
5	SSA	450.5	GA	450.5	SSA	450.5	WOA	402.69	GA	450.5	SSA	450.5
6	GA	550.5	MRFO	550.92	GA	566.92	SSA	610.44	MRFO	599.52	GA	550.59
7	MRFO	690.76	PSO	666.53	MRFO	648.59	MRFO	621.23	PSO	620.71	MRFO	656.9

Table 8 (continued)

Rank	F7		F8		F9		F10		F11		F12	
	Algorithm	MeanRank	Algorithm	MeanRank	Algorithm	MeanRank	Algorithm	MeanRank	Algorithm	MeanRank	Algorithm	MeanRank
8	PSO	717.6	TSA	775.98	PSO	766.73	PSO	740.31	TSA	785.24	PSO	772.91
9	TSA	848.13	ACO	896.75	TSA	890.87	TSA	832.78	SSA	860.14	TSA	834.43
10	ACO	982.12	CS	930.16	FPA	906.23	ACO	967.43	ACO	965.66	ACO	971.05
11	CS	1046.04	FPA	1006.58	ACO	1074.34	CS	1052.77	CS	1010.47	CS	1038.35
12	FPA	1118.35	SSA	1126.58	CS	1099.82	FPA	1128.26	FPA	1111.76	FPA	1129.27
Chi-square	1148.565		1144.036		1126.581		1161.366		1122.614		1143.989	
Prob > Chi-square	1.96E−239		1.86E−238		1.07E−234		3.43E−242		7.65E−234		1.90E−238	

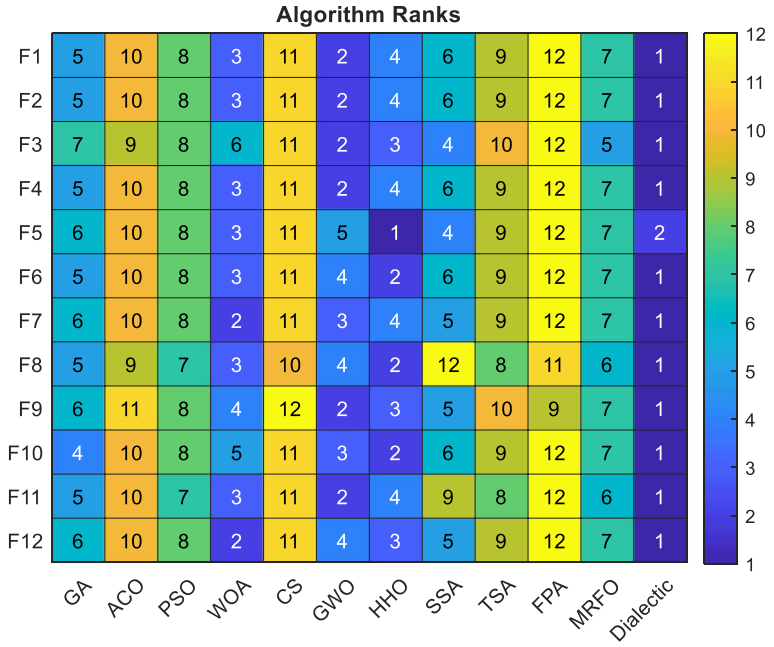


Fig. 35 Heatmap of algorithm rankings across functions

7 Discussion

The findings of this study offer a comprehensive validation of the proposed DOA from both empirical and theoretical perspectives. The algorithm has been systematically evaluated across a diverse suite of twelve benchmark functions (F1–F12) and compared against eleven well-established metaheuristic algorithms: Genetic Algorithm (GA), Ant Colony Optimization (ACO), Particle Swarm Optimization (PSO), Whale Optimization Algorithm (WOA), Cuckoo Search (CS), Grey Wolf Optimizer (GWO), Harris Hawks Optimization (HHO), Salp Swarm Algorithm (SSA), Tunicate Swarm Algorithm (TSA), Flower Pollination Algorithm (FPA), and Marine Predators Algorithm (MRFO). This comparative analysis was conducted based on four key performance criteria: solution quality, stability, convergence efficiency, and overall ranking.

7.1 Solution quality

DOA demonstrated the ability to consistently produce final solutions remarkably close to the global optima across the majority of benchmark functions. Specifically, it ranked first on five functions and second on six others. The Kruskal–Wallis test confirms that these performance differences are statistically significant, underscoring the superior accuracy of DOA in solving complex and multimodal optimization problems. The Friedman test further reinforces these results, with DOA achieving

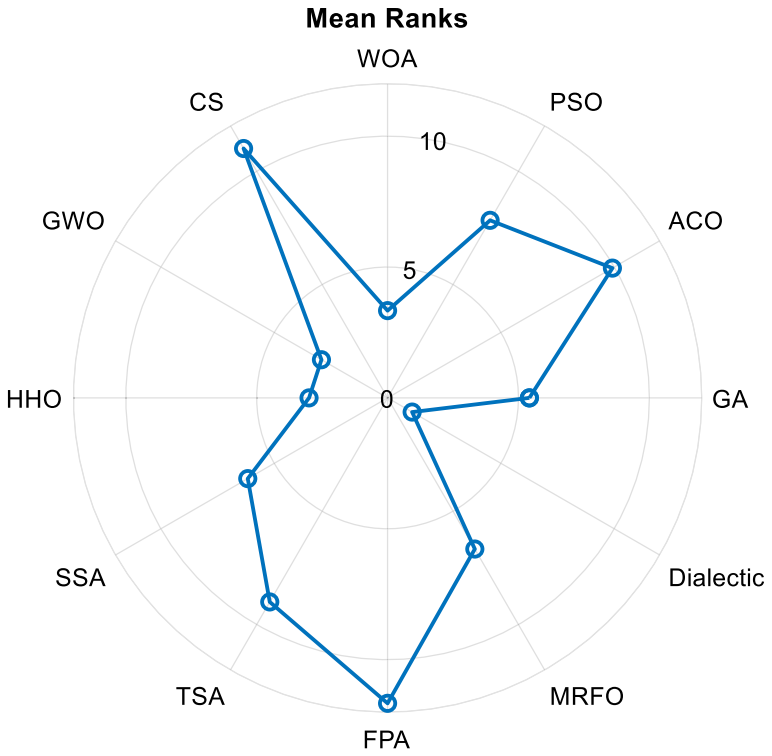


Fig. 36 presents a radar chart plotting the mean ranks

the lowest mean rank among all competitors, highlighting its dominant solution quality throughout the benchmark suite.

7.2 Stability

One of the most salient advantages of DOA lies in its high stability and low standard deviation across independent runs. The boxplot distributions (Figs. 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34 and 35) exhibit tight interquartile ranges, symmetrical structures, and an absence of outliers, in contrast to algorithms such as FPA, CS, ACO, and PSO, which frequently show volatile and dispersed results. This consistent behavior indicates DOA's robustness and reliability in maintaining performance integrity, even under the presence of random initialization or stochastic variations—qualities essential for industrial and real-world applications.

7.3 Convergence efficiency

DOA also outperforms competing algorithms in terms of convergence speed. The convergence plots reveal that DOA reaches high-quality solutions earlier and maintains a smooth, non-oscillatory trajectory toward optima. Unlike many other algorithms that experience stagnation or erratic behavior, DOA progresses steadily without premature convergence. This efficiency reflects a well-balanced trade-off between exploration and exploitation within its search mechanism. Additionally, the Kolmogorov–Smirnov test indicates that while the distribution of DOA's outputs may deviate from normality, this deviation is structured and stable, not random—further supporting its consistent search behavior.

7.4 Overall ranking

The Friedman test results place DOA at the top, with a mean rank of 1.08, the best among the twelve compared algorithms. This consistent ranking across all benchmark functions highlights DOA's general superiority in overall performance. In contrast, algorithms such as FPA and CS consistently ranked lowest, with mean ranks of 11.67 and 11.00, respectively—indicative of their limited effectiveness in high-dimensional or deceptive landscapes. Moreover, the Mann–Whitney test confirms that DOA does not suffer from statistically significant inferiority in any distributional comparison, affirming that its superiority is not only consistent but also statistically sound.

7.5 Generalization and adaptability

An essential aspect of DOA's performance is its uniform effectiveness across both complex and straightforward optimization landscapes. Unlike many traditional algorithms that exhibit selective strength—performing well on specific problems but poorly on others—DOA maintains high performance regardless of the nature of the function. It adapts effectively to varying degrees of multimodality, separability, and deceptiveness within the objective functions. This adaptability makes it a strong candidate for real-world optimization tasks that involve structural diversity and uncertainty.

7.6 Theoretical support and formal analysis

The empirical performance of DOA is fully aligned with its rigorous theoretical foundation. The algorithm is formally defined within a First-Order Logic (FOL) framework and incorporates logistic population dynamics and synthesis-driven adaptation mechanisms. Its mathematical model leverages bounded growth functions, spectrally constrained influence matrices, and contraction mappings, ensuring convergence to stable optima under well-defined conditions. This theoretical

rigor lends strong support to the observed empirical convergence, stability, and robustness.

The DOA exhibits a unique and powerful blend of precise convergence behavior, distributional stability, statistical superiority, and rigorous theoretical foundations. Its consistently high performance across diverse optimization landscapes—combined with its scalability, adaptability, and robustness—positions DOA as a versatile and next-generation metaheuristic framework. Moreover, its structural flexibility makes it well-suited for extension to advanced problem domains such as multi-objective optimization, dynamic environments, and hybrid algorithmic integrations. Furthermore, DOA's inherently parallelizable design makes it highly suitable for scalable, real-time optimization on high-performance computing platforms.

8 Conclusion

This paper introduced the DOA, a population-based metaheuristic inspired by Hegelian and Marxist dialectics. By simulating ideological evolution among three interacting subpopulations—Supporters, Opponents, and Neutrals DOA integrates logistic growth models, influence matrices, contradiction analysis, and synthesis mechanisms into a structured and adaptive optimization framework. This dialectical formulation provides a balanced mechanism for exploration and exploitation, preserving population diversity and driving convergence toward global optima. The empirical evaluation across twelve benchmark functions (F1–F12) and comparisons with eleven well-established metaheuristics—including GA, ACO, PSO, WOA, GWO, and HHO—confirmed the superiority of DOA in multiple dimensions. DOA achieved first place in five functions and second place in six others, reflecting strong solution quality and problem-independence. Boxplot distributions indicated high stability with consistently narrow interquartile ranges and minimal variance, while convergence plots highlighted its ability to reach high-quality solutions faster and without stagnation. Statistical validation through Kruskal–Wallis and Friedman tests reinforced DOA's dominant performance, with a mean rank of 1.08, the best among all competitors. In contrast, FPA, CS, and PSO frequently exhibited premature convergence or high sensitivity to initialization, whereas DOA consistently provided robust and repeatable results. These outcomes establish DOA as a reliable and scalable optimizer capable of addressing diverse optimization challenges. The theoretical analysis of DOA, grounded in First-Order Logic axioms, bounded growth lemmas, and contraction-mapping-based convergence proofs, complements its empirical strength. This dual emphasis distinguishes DOA from traditional bio-inspired methods by coupling formal soundness with statistical superiority, offering a novel philosophical foundation for metaheuristic design. Several limitations remain. The current experiments were restricted to single-objective benchmark functions in fixed dimensions, leaving the scalability of DOA to ultra-high-dimensional spaces and its adaptability to dynamic environments open for further investigation. The algorithm has not yet been validated on large-scale, real-world applications, such as cloud–fog scheduling, IoT resource allocation, and engineering design, where

additional computational constraints may be present. As part of our future work, we plan to extend DOA to multi-objective and constrained optimization, investigate its performance in dynamic and uncertain environments, and integrate it into hybrid frameworks that utilize complementary algorithms. Furthermore, we will explore its parallel and distributed implementation on high-performance computing platforms to rigorously evaluate scalability, efficiency, and applicability to large-scale real-time optimization scenarios

Acknowledgements This article is dedicated to the loving memory of my late mother, whose wisdom and encouragement continue to inspire me. I am deeply grateful to my father for his ongoing support and guidance, and to my wife for her unwavering understanding and patience throughout the research and writing process.

Author contribution Y.S. conceived and designed the study; Y.S. developed the methodology and implemented the dialectic optimization algorithm; Y.S. conducted all experiments and performed the statistical analyses; Y.S. prepared all figures and tables; Y.S. drafted the manuscript and carried out all revisions. Y.S. reviewed and approved the final version of the manuscript.

Data availability Data used to support this novel scheme are included within the article.

Declarations

Conflict of interest The authors declare no competing interests.

References

1. Wase V, Wuyckens S, Lee JA, Saint-Guillain M (2024) The proton arc therapy treatment planning problem is NP-hard. Comput Biol Med 171:108139
2. Rodan A, Al-Tamimi A-K, Al-Alnemer L, Mirjalili S (2025) Stellar oscillation optimizer: a nature-inspired metaheuristic optimization algorithm. Cluster Comput 28(6):1–51
3. Marrouche W (2024) Unlocking the potential of metaheuristics for NP-hard problems. University of Portsmouth
4. Azizi M, Baghalzadeh Shishehgarkhaneh M, Basiri M, Moehler RC (2023) Squid Game Optimizer (SGO): a novel metaheuristic algorithm. Sci Rep 13(1):1–24. https://doi.org/10.1038/s41598-023-32465-z
5. Khodadadi N et al (2024) Multi-objective generalized normal distribution optimization: a novel algorithm for multi-objective problems. Cluster Comput. https://doi.org/10.1007/s10586-024-04467-7
6. Azizi M, Baghalzadeh Shishehgarkhaneh M, Basiri M, Moehler RC, Fang Y, Chan M (2024) Wolf-Bird Optimizer (WBO): a novel metaheuristic algorithm for building information modeling-based resource tradeoff. J Eng Res. https://doi.org/10.1016/j.jer.2023.11.024
7. Salami Y, Khajehvand V, Zeinali E (2024) SOS-FCI: a secure offloading scheme in fog–cloud-based IoT. J Supercomput 80(1):570–600. https://doi.org/10.1007/s11227-023-05499-3
8. Salami Y, Khajehvand V, Zeinali E (2024) A new secure offloading approach for internet of vehicles in fog-cloud federation. Sci Rep 14(1):5576
9. Salami Y (2024) SOBT-UF: secure offloading in blockchain infrastructure for intelligent transportation systems using 5G-enabled UAVs within a fog-edge computing federation. In: 2024 19th Iranian Conference on Intelligent Systems (ICIS). IEEE, pp 217–222
10. Abdollahzadeh B, Gharehchopogh FS, Mirjalili S (2021) African vultures optimization algorithm: a new nature-inspired metaheuristic algorithm for global optimization problems. Comput Ind Eng 158:107408
11. Abdollahzadeh B et al (2024) Puma optimizer (PO): a novel metaheuristic optimization algorithm and its application in machine learning. Cluster Comput 27(4):5235–5283

12. Shami TM, Al-Tashi Q, Khodadadi N, Abdulkadir SJ, Ahmed A, Mirjalili S (2025) Dimension selection: an innovative metaheuristic strategy for particle swarm optimization. Cluster Comput 28(6):1–26
13. Jia H, Zhou X, Zhang J, Mirjalili S (2025) Superb fairy-wren optimization algorithm: a novel metaheuristic algorithm for solving feature selection problems. Cluster Comput 28(4):246
14. Mirjalili S (2015) The ant lion optimizer. Adv Eng Softw 83:80–98
15. Mirjalili S, Mirjalili SM, Hatamlou A (2016) Multi-verse optimizer: a nature-inspired algorithm for global optimization. Neural Comput Appl 27:495–513
16. Abualigah L, Diabat A, Mirjalili S, Abd Elaziz M, Gandomi AH (2021) The arithmetic optimization algorithm. Comput Methods Appl Mech Eng 376:113609
17. Mirjalili S, Mirjalili S (2019) Genetic algorithm. In: Evol. Algorithms Neural Networks Theory Appl., pp 43–55
18. Gharehchopogh FS, Namazi M, Ebrahimi L, Abdollahzadeh B (2023) Advances in sparrow search algorithm: a comprehensive survey. Arch Comput Methods Eng 30(1):427–455
19. Gendreau M, Potvin J-Y (2005) Metaheuristics in combinatorial optimization. Ann Oper Res 140:189–213
20. Kirkpatrick S, Gelatt CD Jr, Vecchi MP (1983) Optimization by simulated annealing. Science 220(4598):671–680
21. Zames G (1981) Genetic algorithms in search, optimization and machine learning. Inf Tech J 3(1):301
22. Shi XH, Liang YC, Lee HP, Lu C, Wang LM (2005) An improved GA and a novel PSO-GA-based hybrid algorithm. Inf Process Lett 93(5):255–261
23. Henderson D, Jacobson SH, Johnson AW (2003) The theory and practice of simulated annealing. In: Handb. Metaheuristics, pp 287–319
24. Dorigo M, Stützle T (2003) The ant colony optimization metaheuristic: algorithms, applications, and advances. In: Handb. Metaheuristics, pp 250–285
25. Wang L, Hong L, Fu H, Cai Z, Zhong Y, Wang L (2025) Adaptive distance-based multi-objective particle swarm optimization algorithm with simple position update. Swarm Evol Comput 94:101890
26. Dinesh A, Rangaraj J (2025) An energy-efficient routing protocol for wireless body area networks using hybrid artificial bee colony optimization and chicken swarm optimization algorithm. J Eng Appl Sci 72(1):1–37
27. Essa KS, Gomaa OA, Elhussein M, Géraud Y, Diraison M, Diab ZE (2025) A prosperous and thorough analysis of gravity profiles for resources exploration utilizing the metaheuristic bat algorithm. Sci Rep 15(1):5000
28. Qasim M, Sajid M (2025) An efficient IoT task scheduling algorithm in cloud environment using modified firefly algorithm. Int J Inf Technol 17(1):179–188
29. Almufti SM, Marqas RB, Asaad RR, Shaban AA (2025) Cuckoo search algorithm: overview, modifications, and applications. Int J Sci World. https://doi.org/10.14419/efkvvd44
30. Singh H et al (2025) An integrative TLBO-driven hybrid grey wolf optimizer for the efficient resolution of multi-dimensional, nonlinear engineering problems. Sci Rep 15(1):11205
31. Wei J, Gu Y, Lu B, Cheong N (2025) RWOA: a novel enhanced whale optimization algorithm with multi-strategy for numerical optimization and engineering design problems. PLoS ONE 20(4):e0320913
32. Emara FA, Gad-Elrab AAA, Sobhi A, Alsharkawy AS, Embabi ME, Abd El-Baky MA (2025) Multi-objective task scheduling algorithm for load balancing in cloud computing based on improved Harris hawks optimization. J Supercomput 81(6):1–38
33. Sasikumar A et al (2025) An efficient binary salp swarm algorithm for user selection in multiuser MIMO antenna systems. Sci Rep 15(1):1–17
34. Mirjalili S (2015) Moth-flame optimization algorithm: a novel nature-inspired heuristic paradigm. Knowl-Based Syst 89:228–249
35. Meng S, Huang W, Xu X, Li Q, Dou W, Liu B (2019) A self-adaptive PSO-based dynamic scheduling method on hierarchical cloud computing. Cloud computing, Smart Grid and Innovative Frontiers in Telecommunications. Springer, Cham, pp 89–100
36. Abdel-Basset M, Abdel-Fatah L, Sangaiah AK (2018) Metaheuristic algorithms: a comprehensive review. In: Comput. Intell. Multimed. Big Data Cloud with Eng. Appl., pp 185–231
37. Wang D, Tan D, Liu L (2018) Particle swarm optimization algorithm: an overview. Soft Comput 22(2):387–408

38. Mafarja M, Mirjalili S (2018) Whale optimization approaches for wrapper feature selection. Appl Soft Comput 62:441–453
39. Bidell TR (2024) Basis of dialectics. In: Routledge Int. Handb. Dialect. Think., p 105
40. Maybee JE (2016) Hegel's dialectics
41. Orji CP (2023) A critique of Karl Marx's theory of dialectics materialism as a solution to contemporary nation development. AMAMIHE J Appl Philos 21(4)
42. Rigby SH (2024) Engels and the formation of Marxism: history, dialectics and revolution. Engels and the formation of Marxism. Manchester University Press
43. Huanghui Y (2023) Historical dialectics of Marx's labor theory of value: taking value as a clue. J Beijing Univ Aeronaut Astronaut Soc Sci Ed 36(3):47–55
44. Einsohn HI (2023) Dialectics. In: Bernard Shaw, Paul Ricoeur, and the Jesusian dialectics of redemptive living. Springer, Cham, pp 17–50
45. Metropolis N, Ulam S (1949) The Monte Carlo method. J Am Stat Assoc 44(247):335–341
46. Hooke R, Jeeves TA (1961) Solution of numerical and statistical problems. J ACM 8(2):212–229
47. Nelder JA, Mead R (1965) A simplex method for function minimization. Comput J 7(4):308–313
48. David BF (1998) Artificial intelligence through simulated evolution. Evol Comput Foss Rec 227–296
49. Sampson JR (1976) Adaptation in natural and artificial systems. Society for Industrial and Applied Mathematics
50. Kennedy J, Eberhart R (1995) Particle swarm optimization. In: Proceedings of ICNN'95-International Conference on Neural Networks. IEEE, pp 1942–1948
51. Karaboga D (2005) An idea based on honey bee swarm for numerical optimization
52. Yang X-S (2010) Firefly algorithm, stochastic test functions and design optimisation. Int J Bio-Inspired Comput 2(2):78–84
53. Yang X-S, Deb S (2009) Cuckoo search via Lévy flights. In: 2009 World Congress on Nature & Biologically Inspired Computing (NaBIC). IEEE, pp 210–214
54. Geem ZW, Kim JH, Loganathan GV (2001) A new heuristic optimization algorithm: harmony search. SIMULATION 76(2):60–68
55. Rashedi E, Nezamabadi-Pour H, Saryazdi S (2009) GSA: a gravitational search algorithm. Inf Sci 179(13):2232–2248
56. Yang X-S (2010) A new metaheuristic bat-inspired algorithm. Nature inspired cooperative strategies for optimization (NICSO 2010). Springer, Cham, pp 65–74
57. Mirjalili S, Mirjalili SM, Lewis A (2014) Grey wolf optimizer. Adv Eng Softw 69:46–61
58. Mirjalili S, Lewis A (2016) The whale optimization algorithm. Adv Eng Softw 95:51–67
59. Rao RV, Savsani VJ, Vakharia DP (2011) Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems. Comput Aided Des 43(3):303–315
60. Mirjalili S (2016) Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems. Neural Comput Appl 27:1053–1073
61. Cheng S, Qin Q, Chen J, Shi Y (2016) Brain storm optimization algorithm: a review. Artif Intell Rev 46(4):445–458
62. Mirjalili S, Hashim SZM (2012) BMOA: binary magnetic optimization algorithm. Int J Mach Learn Comput 2(3):204
63. Azizi M, Talatahari S, Gandomi AH (2023) Fire hawk optimizer: a novel metaheuristic algorithm. Artif Intell Rev 56(1):287–363