FOUNDATIONS OF COMPUTING AND DECISION SCIENCES
Vol. 51 (2026) No. 1
ISSN 0867-6356
DOI: 10.2478/fcds-2026-0005 e-ISSN 2300-3405

sciendo

MOPOA: A New Multi-Objective Pufferfish Optimization Algorithm

Mohaddethe Nasrabadi* ,Mahdi khazaiepoor•, Mahdi Kherad+

Abstract. Multi-objective optimization problems (MOPs) pose significant challenges due to the presence of multiple conflicting objectives. This paper introduces MOPOA, a novel Multi-Objective Pufferfish Optimization Algorithm inspired by the defensive behaviors of pufferfish in nature. MOPOA extends the original single-objective POA by incorporating Pareto dominance, an external archive for preserving non-dominated solutions, and a crowding distance mechanism to maintain solution diversity. The algorithm balances exploration and exploitation through biologically inspired phases simulating predator-prey interactions. To evaluate MOPOA's performance, it was benchmarked against several state-of-the-art algorithms, including NSGA-III, MOPSO, MODA, and MOFDO, on two well-known test suites: the ZDT and CEC-2019 multi-objective functions. Results indicate that MOPOA not only achieves superior convergence to the Pareto front but also maintains high diversity and robustness across diverse optimization scenarios. These findings position MOPOA as a powerful and adaptive tool for solving complex real-world multi-objective problems.

Keywords: Multi-objective optimization; Pufferfish Optimization Algorithm, Pareto front, metaheuristics, evolutionary algorithms, exploration and exploitation.

1. Introduction

In numerous real-world contexts, optimization challenges encompass a variety of opposing goals that demand simultaneous attention. These challenges, commonly termed multi-objective optimization problems, emerge across diverse fields such as engineering, environmental studies, transportation management, and economic analysis [50]. The intricacy of these issues arises from the fact that advancing one goal often undermines another, calling for precise compromises. The aim is not to pinpoint a single answer but to

* Phd Student, Department of Computer Engineering, Bi.C., Islamic Azad University, Birjand, Iran; Nasrabadi_mohaddethe@iau.ac.ir

• Assistant Professor, Department of Computer Engineering, Bi.C., Islamic Azad University, Birjand, Iran; Mkhazaiepoor@iau.ac.ir

+ Phd Student, Department of Computer Engineering and IT, University of Qom, Iran; m.kherad@stu.qom.ac.ir

uncover a group of Pareto-optimal solutions known as the Pareto front, where improving any one objective inevitably detracts from at least one other [42].

Multi-objective evolutionary algorithms (MOEAs) have long been the dominant approach for solving multi-objective optimization problems (MOPs), owing to their flexibility and ability to approximate the Pareto front. However, despite the success of classical algorithms such as Non-Dominated Sorting Genetic Algorithm II (NSGA-II) {Deb, 2002 #851}, Multi-Objective Particle Swarm Optimization (MOPSO) {Coello, 2004 #2525}, and Multi-Objective Differential Evolution (MODE) [54], more recent studies have revealed that these methods still face limitations in convergence speed, scalability, and diversity preservation.

In recent years, researchers have proposed several advanced MOEAs to overcome these limitations. For example, MOEA-Isa [55] introduced an interval-based initialization strategy for better population distribution; DK-MOEA {Liu, 2023 #2782} incorporated domain knowledge to guide the evolutionary process; QSLMOF {Wu, 2024 #2783} proposed a bidirectional vector-based sampling strategy for high-dimensional MOPs; and TS-MOEA {Liu, 2024 #2784} adopted a two-stage search framework to avoid premature convergence.

Similarly, LDS-AF {Gu, 2025 #2785} applied surrogate-assisted learning to handle large-scale problems efficiently, DMOEAs {Wang, 2024 #2786} combined decision-maker preferences with dynamic search, and RMOEA-SuR {Jiang, 2025 #2787} introduced a survival-rate objective to enhance robustness against noise. Furthermore, MRL-MOEA {Wang, 2025 #2788} integrated reinforcement learning to adapt evolutionary operators dynamically.

Despite these advancements, most of these algorithms still rely on mathematically complex or computationally intensive mechanisms to balance convergence and diversity. This motivates the development of simpler yet adaptive biologically inspired algorithms capable of efficiently maintaining this balance.

Several well-known algorithms have been proposed to address multi-objective problems, belonging to different families of metaheuristics.

- Evolutionary algorithms (EAs), such as NSGA-II {Deb, 2002 #851}, Strength Pareto Evolutionary Algorithm 2 (SPEA2) {Zitzler, 2001 #2791}, and MOEA/D {Zhang, 2007 #2792}, rely on genetic evolution principles like selection, crossover, and mutation.
- Swarm intelligence-based algorithms (SIs), such as MOPSO {Coello, 2004 #2525} and Multi-Objective Grey Wolf Optimizer (MOGWO) {Mirjalili, 2016 #1779}, use collective behavior and communication among particles or agents to guide the search process.
- Hybrid and adaptive MOEAs, including MRL-MOEA {Wang, 2025 #2788} and surrogate-assisted LDS-AF {Gu, 2025 #2785} methods, represent recent advancements combining optimization with intelligent learning mechanisms.

While these families differ in their underlying mechanisms, all share the same objective: maintaining a well-distributed set of Pareto-optimal solutions. However, they still struggle to simultaneously achieve fast convergence and strong diversity maintenance, motivating the biologically inspired approach proposed in MOPOA.

A new bio-inspired metaheuristic algorithm, called the Pufferfish Optimization Algorithm (POA), which imitates the natural behavior between pufferfish and their predators in the sea, is introduced in [3]. The fundamental inspiration for POA is derived from the attacks of hungry predators on pufferfish and the defense mechanism of pufferfish against these attacks. To overcome the constraints of traditional multi-objective evolutionary approaches, this research introduces the Multi-Objective Pufferfish Optimization Algorithm (MOPOA), a creative evolution of the foundational POA. MOPOA is crafted to dynamically regulate the interplay between exploration and exploitation, resulting in superior Pareto front coverage and accelerated convergence rates. In the exploration phase, pufferfish search broadly across the decision space to identify promising regions, while in the exploitation phase, they focus on these discovered areas to find better solutions. High ability in exploration, exploitation, and the balance between them during the search process to provide effective solutions, MOPOA delivers a robust and adaptable framework for addressing multi-objective optimization tasks. The MOPOA algorithm employs an external archive to store non-dominated solutions. With each iteration, this archive is enhanced through mechanisms that update the non-dominated solutions, while simultaneously utilizing a crowding distance mechanism to maintain diversity along the Pareto front. During the search process, pufferfish use both the positions stored in the archive and the positions of current population members as guides to update their own locations. This approach allows the algorithm to strike a balance between exploration and exploitation, converging toward a diverse and high-quality set of Pareto solutions. By integrating innovative mechanisms inspired by pufferfish behavior, the MOPOA algorithm demonstrates a strong capability to identify optimal solutions for complex problems. It can be applied in various fields, such as engineering design, resource planning, network optimization, and many other domains involving multiple objective functions. The implementation of this algorithm, leveraging advanced techniques like archiving and crowding distance, enhances its performance and enables it to achieve higher-quality Pareto fronts. The main contributions of this work include:

1. The development of MOPOA as a multi-objective extension of the single-objective POA algorithm.
2. The balance of exploration and exploitation in MOPOA, achieved through the simulation of the predator’s attack on the pufferfish and the pufferfish’s defense mechanism against the predator, improves both coverage and convergence speed.
3. The use of an archive technique for non-dominant solutions ensures that high-quality solutions are stored effectively.
4. A crowding distance mechanism to prevent premature convergence and maintain diversity along the Pareto front.
5. The enhancement of algorithm performance through the integration of the concept of Pareto dominance, which provide a mechanism to determine "better" solutions during the predator attack phase and for updating the archive with non-dominated individuals.

This paper assesses the performance of MOPOA using widely recognized benchmark datasets, including the ZDT functions and the IEEE Congress on Evolutionary Computation (CEC) 2019 multi-modal benchmarks. Comparative evaluations against state-of-the-art multi-objective evolutionary algorithms (MOEAs) demonstrate MOANA’s advantages in terms of convergence speed and its ability to effectively cover the Pareto front.

In summary, while recent MOEAs have improved performance through reinforcement learning, surrogate models, and adaptive operators, their computational overhead and parameter dependencies remain significant. The proposed MOPOA introduces a new biologically inspired paradigm that models pufferfish defense dynamics to adaptively control exploration and exploitation, offering a lightweight yet competitive solution to modern multi-objective optimization challenges.

The structure of this paper is as follows: Section 2 provides an extensive review of existing multi-objective optimization algorithms. Section 3 presents the theoretical background, covering key concepts, and MOPOA is comprehensively described in terms of mathematical and programmatic aspects. Section 4 presents a detailed performance analysis of MOPOA across various benchmark optimization problems, comparing its results with leading algorithms like NSGA-II, MOPSO, and MODA. Finally, Section 5 provides concluding remarks, summarizing key findings and outlining future research directions.

2. Literature review

The primary aim of multi-objective optimization algorithms—particularly a posteriori approaches—is to generate highly accurate approximations of the true Pareto-optimal set while maintaining a high level of solution diversity [57]. This diversity is crucial as it offers decision-makers a broad spectrum of viable design alternatives. Traditionally, multi-objective problems were solved by converting them into single-objective problems through a priori aggregation techniques. However, this approach suffers from two significant limitations [23, 29]:

First, assigning evenly distributed weights to objectives does not ensure that the resulting solutions are evenly distributed along the Pareto front. Second, these methods fail to identify non-convex regions of the Pareto front due to the requirement that all weights must be non-negative and sum to a constant. In the weighted-sum scalarization method, the objective functions are combined using predefined weights that are typically normalized such that their sum equals on , enabling the exploration of different trade-off regions on the Pareto front.Essentially, traditional aggregation methods rely on forming a convex combination of objectives, which limits their ability to fully explore the solution space.

Several studies have attempted to enhance this approach. For instance, Parsopoulos and Vrahatis [39] introduced two adaptive weighting schemes—one gradual and one abrupt—to dynamically modify weights during optimization. Despite these improvements, the fundamental limitations of aggregation methods remain. Moreover, because each run of these methods typically yields only a single optimal solution, they must be executed multiple times to approximate the entire Pareto front.

As Deb [11] points out, using metaheuristic techniques in multi-objective optimization helps navigate numerous challenges, including infeasible regions, fragmented or local Pareto fronts, maintaining solution diversity, and avoiding premature convergence. A priori methods must confront all these issues independently in each execution. On the other hand, retaining the multi-objective nature of a problem offers significant benefits—most notably, the cooperative exchange of information among search agents, which facilitates more efficient exploration and quicker convergence toward the true Pareto front.

Second, multi-objective methods offer the advantage of approximating the entire true Pareto front in a single execution. Moreover, preserving the multi-objective structure of a problem enables a deeper understanding of how design parameters and operating conditions influence performance across a range of objectives. Despite these advantages, a key limitation of a priori methods is the increased complexity they introduce—often requiring the use of sophisticated metaheuristic strategies to effectively handle conflicting goals.

Nevertheless, many of the most prominent heuristic algorithms have been successfully adapted for multi-objective optimization.

Among these, the Non-dominated Sorting Genetic Algorithm II (NSGA-II) has emerged as one of the most widely used and effective multi-objective metaheuristics [13]. NSGA-II builds upon the foundation of the classic Genetic Algorithm (GA) [15] and addresses several shortcomings of its predecessor, the original NSGA [48]. The initial version struggled with high computational overhead in non-dominated sorting, lacked an elitism mechanism, and relied on a fixed sharing parameter. NSGA-II overcomes these issues by introducing a more efficient non-dominated sorting algorithm, incorporating elitism to preserve high-quality solutions, and using a parameter-less crowding distance mechanism to maintain diversity without requiring predefined parameters.

The second most widely adopted multi-objective metaheuristic is Multi-Objective Particle Swarm Optimization (MOPSO), initially introduced by Coello et al. {Coello, 2004 #2525}. Building upon the fundamental principles of the original PSO {Kennedy, 1995 #288}, MOPSO utilizes a swarm of particles that navigate the search space in pursuit of optimal solutions. Each particle updates its position based on its own historical best (pbest) and a globally shared best solution (gbest) within the swarm, as originally described by Shi & Eberhart [46]. However, unlike single-objective PSO, MOPSO does not rely on a unique global best, since multiple optimal solutions exist in multi-objective scenarios. Therefore, particles must consider non-dominated solutions from both their own search history and the swarm's collective findings.

To manage this, MOPSO commonly uses an external archive to store discovered Pareto-optimal solutions. This archive serves as a memory to preserve high-quality, non-dominated results throughout the optimization process. To further enhance solution diversity and exploration, a mutation strategy—referred to as turbulence—is occasionally applied, introducing randomness and helping particles escape local optima. A detailed review of PSO-based multi-objective algorithms is available in the work by Reyes-Sierra and Coello [43].

The external archive used in MOPSO resembles the adaptive grid structure employed in the Pareto Archived Evolution Strategy (PAES) proposed by Knowles and Corne [24]. This archive system consists of two main components: an archive controller and a grid-based selection mechanism. The archive controller determines whether a new solution should be retained. If a new solution is dominated by any current archive member, it is discarded. Conversely, if it is non-dominated, it is added to the archive, and any previously stored solutions that it dominates are removed. When the archive reaches its capacity, the adaptive grid mechanism is activated to manage space and maintain diversity by organizing and selecting solutions based on their distribution within the objective space.

A variety of multi-objective optimization algorithms have emerged, showcasing the adaptability of metaheuristics to complex multi-criteria problems. Notable examples include Multi-Objective Cat Swarm Optimization (MOCSO) [40], Multi-Objective Ant Colony Optimization [36], Multi-Objective Teaching–Learning-Based Optimization [27], Multi-Objective Artificial Bee Colony [16], Multi-Objective Differential Evolution [52], and the Multi-Objective Gravitational Search Algorithm (MOGSA) [17]. These developments demonstrate the growing effectiveness of metaheuristic frameworks in tackling multi-objective problems.

Despite their success in approximating the Pareto front, none of these algorithms can universally solve every optimization task. According to the No Free Lunch (NFL) theorem, no single algorithm performs best across all problem domains. As a result, new algorithms may outperform existing ones in certain scenarios, offering solutions where traditional methods fall short. This underscores the importance of continued exploration and development of novel optimization strategies tailored to specific problem characteristics.

The body of research on multi-objective evolutionary algorithms (MOEAs) encompasses a broad spectrum of strategies aimed at solving multi-objective optimization problems (MOPs). These strategies are commonly classified into three primary categories: indicator-based approaches (e.g., IBEAs) [21], decomposition-based methods [28], and dominance-based techniques [44]. Each category employs different mechanisms to approximate the Pareto-optimal front. For instance, dominance-based algorithms like NSGA-II [18] and SPEA2 {Zitzler, 2001 #2791} utilize non-dominated sorting to rank individuals within a population, whereas decomposition-based methods [53] transform the MOP into multiple scalar subproblems using weighted aggregations to estimate the Pareto front. NSGA-II, in particular, is renowned for its ability to maintain diversity via a crowding distance measure and has been further adapted for more complex scenarios, as seen in NSGA-III [20]. Similarly, MOPSO {Coello, 2004 #2525}, another widely adopted approach, employs an archive-based grid system to preserve diversity without compromising efficiency.

Within the broader domain of population-based metaheuristics [2], algorithms such as Differential Evolution (DE) {Storn, 1997 #2796} and Covariance Matrix Adaptation Evolution Strategy (CMA-ES) {Hansen, 2001 #2795} have shown strong performance in continuous, single-objective settings. While DE is effective at maintaining diversity, CMA-ES is sometimes prone to stagnation in local optima [19]. To mitigate this, the IR-CMA-ES hybrid method integrates DE with CMA-ES to enhance global exploration and avoid premature convergence. Although this hybrid improves diversity and search performance, it also introduces challenges, such as increased computational complexity and sensitivity to parameter tuning. Additionally, it may experience scalability issues and slower convergence in high-dimensional or complex problem spaces [8].

Over the last decade, a variety of novel MOEAs have emerged, drawing inspiration from biological and physical systems. Examples include the Whale Optimization Algorithm [33], Ant Lion Optimizer [32], Grey Wolf Optimizer {Mirjalili, 2016 #1779}, Moth Flame Optimization [31], Multi-Objective Cat Swarm Optimization [6], Dragonfly Algorithm [30], and learner performance-based behavioral models [41], as well as fitness-dependent optimization techniques [1]. These approaches aim to overcome the limitations of traditional MOEAs by introducing innovative mechanisms for balancing convergence and diversity. Indicator-based methods, in particular, have gained traction due to their theoretical robustness and use of performance metrics like hypervolume and the epsilon-indicator to guide the search process [4, 7].

Over the past three years, a number of innovative multi-objective evolutionary algorithms have emerged, focusing on scalability, adaptivity, and computational efficiency. For instance, MOEA-ISa {Xue, 2022 #2781} and DK-MOEA {Liu, 2023 #2782} enhanced initialization and mutation using knowledge-based mechanisms; QSLMOF {Wu, 2024 #2783} introduced a directed search guided by convergence gradients; and LDS-AF {Gu, 2025 #2785} leveraged surrogate models to drastically reduce the number of fitness evaluations for problems with up to 1000 decision variables.

DMOEAs {Wang, 2024 #2786} and DVA-TPCEA {Li, 2024 #2789} extended MOEAs to dynamic environments and decision-variable analysis, while MOEA/TS {Zhao, 2024 #2790} proposed a three-state adaptive framework for many-objective problems.

Most recently, RMOEA-SuR {Jiang, 2025 #2787} and MRL-MOEA {Wang, 2025 #2788} introduced robustness and reinforcement learning into MOEAs, respectively—marking a shift toward intelligent, self-adaptive multi-objective frameworks.

These developments demonstrate the continuous evolution of the MOEA research landscape. Nevertheless, maintaining a balance between convergence speed and solution diversity remains an open challenge—especially in high-dimensional and complex multi-modal spaces. To address this, the proposed MOPOA algorithm integrates the natural defense mechanism of pufferfish with Pareto-based selection and adaptive archiving to achieve robust and efficient optimization.

Despite substantial progress in this field, the design of MOEAs still faces the fundamental challenge of balancing two key aspects: the accuracy of the obtained Pareto-optimal solutions and the diversity across the solution set. Most algorithms employ metaheuristic strategies that begin with a random population and iteratively refine candidate solutions [22]. However, achieving an optimal trade-off between exploration (searching new regions) and exploitation (intensifying search around promising areas) remains a critical issue, prompting ongoing research into adaptive mechanisms and hybrid approaches.

For example, Mostaghim and Teich [35] proposed a sigma-based strategy for selecting local leaders within the population, whereas Pulido and Coello [51] incorporated clustering mechanisms to enhance the diversity of solutions along the Pareto front. Meanwhile, Zitzler [58] emphasized the role of elitism by applying crossover and mutation operators not only to individuals from the current population but also to those stored in the external archive [22]. Similarly, Laumanns et al. [25] introduced the concept of ε-dominance to simultaneously improve both convergence and diversity. A range of hybrid algorithms have also demonstrated success in tackling domain-specific challenges—for instance, the integration of Grey Wolf Optimization for industrial applications or modified genetic algorithms for optimizing ship routing and scheduling problems [10,49]. Furthermore, methods like Pareto Entropy-based MOPSO, which selects global leaders based on the density of individual solutions, have been effectively used in complex engineering design tasks [37]. The MOLPB technique [41] has been applied to a variety of engineering design problems, including pressure vessel design, speed reducer mechanisms, 4-bar linkages, coil compression springs, and vehicle side-impact structures.

In the context of cultural algorithms, five distinct categories of knowledge contribute to the learning and decision-making process [1]: historical, topographical, normative, domain-specific, and situational knowledge.

To overcome the shortcomings of conventional Multi-Objective Evolutionary Algorithms (MOEAs), the Multi-Objective Pufferfish Optimization Algorithm (MOPOA) was introduced as an innovative framework designed to achieve a more adaptive balance between exploration and exploitation. MOPOA enhances the foundational Pufferfish Optimization Algorithm by the simulation of the predator's attack on the pufferfish and the pufferfish's defense mechanism against the predator to more effectively guide the search process, while also employing crowding distance mechanism to preserve population diversity.

3. Methodology

This section presents a summary of the essential concepts and key definitions associated with multi-objective optimization. It also reviews the concepts of POA and then proposes the MOPOA algorithm. In nature, pufferfish exhibit a distinctive defense behavior: when threatened, they rapidly inflate their bodies by ingesting water or air, increasing their size severalfold to deter predators. Once the threat subsides, they gradually deflate to resume normal activity.

This dual-phase behavior (expansion (defensive inflation) and contraction (relaxation)) can be analogized to the exploration–exploitation trade-off in optimization. Expansion corresponds to exploring a wider search space, while contraction corresponds to refining search within promising regions.

In the proposed algorithm, these two behavioral modes are mathematically represented by adaptive operators that control the step size and direction of population movement according to the current level of population diversity and dominance pressure. Thus, the model draws inspiration from pufferfish behavior but is formalized through well-defined mathematical mechanisms.

3.1. Multi-objective optimization

As outlined in the introduction, multi-objective optimization involves solving problems that contain multiple objective functions. In general terms, such problems can be represented as a maximization task, described as follows:

$$\begin{aligned} &\text{Maximize: } F(\vec{x}) = f_1(\vec{x}), f_2(\vec{x}), \ldots, f_O(\vec{x}) \qquad (1)\\ &\text{Subject to: } g_j(\vec{x}) \geq 0,\ \ j = 1,2,\ \ldots,q \\ &\qquad h_k(\vec{x}) = 0,\ \ k = 1,2,\ \ldots,p \\ &\qquad lb_d \leq\ x_d\ \leq\ ub_d,\ \ d = 1,2,\ \ldots,m \end{aligned}$$

where m is the number of variables, O is the number of objective functions, q is the number of inequality constraints, p is the number of equality constraints, g_j is the jth inequality constraints, h_k indicates the kth equality constraints, and $[lb_d, ub_d]$ are the boundaries of dth variable.

In single-objective optimization, evaluating and comparing solutions is straightforward because only one objective function is involved. For maximization problems, solution X is considered better than solution Y if X > Y. However, in multi-objective optimization, such direct comparisons are not feasible due to the presence of multiple, often conflicting, criteria. Instead, a solution is said to dominate another if it performs at least as well on all objectives and strictly better on at least one. This concept of solution comparison in multi-objective contexts was initially introduced by Francis Ysidro Edgeworth [14] and later expanded upon by Vilfredo Pareto [38].

3.2. Proposed Methodology

This section introduces the proposed Multi-Objective Pufferfish Optimization Algorithm (MOPOA), inspired by the biological behavior of pufferfish. The method builds upon the foundation of the original Pufferfish Optimization Algorithm (POA), adapting its principles to address multi-objective optimization problems. The structure of this section is as follows:

Subsection 3.2.1 presents the core mechanism of the original POA, describing its biological inspiration, mathematical formulation, and behavioral analogy. Subsection 3.2.2 then extends these principles to the multi-objective context, introducing dominance relations, archive management, and diversity-preserving mechanisms that define the MOPOA.

The decision to utilize POA over other methods, such as Differential Evolution (DE) and Covariance Matrix Adaptation Evolution Strategy (CMA-ES), stems from its exploration and exploitation mechanisms. These mechanisms strike an optimal balance between exploring uncharted solution areas and refining known solutions. In contrast to DE and CMA-ES, which employ mutation and recombination techniques to probe the solution space, POA modeled in two phases, (i) exploration based on the simulation of the predator attack on pufferfish and (ii) exploitation based on the simulation of the escape of the predator from the spiny spherical pufferfish. These phases enhance the Multi-Objective Pufferfish Optimization Algorithm's (MOPOA) ability to comprehensively map the Pareto front and pinpoint both global and local optima with greater efficiency—an essential trait for tackling engineering optimization challenges like designing welded beams.

3.2.1. Pufferfish Optimization Algorithm (POA)

The POA algorithm was proposed by S. Mirjalili et al. [3]. Pufferfish are a primarily marine and estuarine fish of the family Tetraodontidae and order Tetraodontiformes. Among the natural behaviors of pufferfish, conflicts between this fish and predators and the use of the defense mechanism of turning into a ball of pointed spines against the attacks of predators are much more significant. The modeling of these natural processes, which consists of (i) a predator's attack on pufferfish and (ii) a pufferfish's defense mechanism against predator attacks, is employed in the design of the POA approach. Each member of the POA represents a potential solution to the problem and can be mathematically expressed as a vector, with each component corresponding to a specific decision variable. Collectively, these members constitute the algorithm's population. Mathematically, the entire set of solution vectors can be represented as a matrix, as shown in Equation (2). The initial positions of all POA members at the start of the algorithm are determined using Equation (3).

$$X = \begin{bmatrix} X_1 \\ \vdots \\ \\ \vdots \\ X_N \end{bmatrix}_{N\times m} = \begin{bmatrix} x_{1,1} & \cdots & x_{1,d} & \cdots & x_{1,m} \\ \vdots & \ddots & \vdots & \iddots & \vdots \\ x_{i,1} & \cdots & x_{i,d} & \cdots & x_{i,m} \\ \vdots & \iddots & \vdots & \ddots & \vdots \\ x_{N,1} & \cdots & x_{N,d} & \cdots & x_{N,m} \end{bmatrix}_{N\times m} \tag{2}$$

$$x_{i,d} = lb_d + r.(ub_d - lb_d) \tag{3}$$

Here, X is the POA population matrix, X_i is the ith POA member (candidate solution), $x_{i,d}$ is its dth dimension in the search space (decision variable), N is the number of population members, m is the number of decision variables, r is a random number in the interval [0, 1], and lb_d and ub_d are the lower bound and upper bound of the dth decision variable, respectively. The set of evaluated values for the objective function of the problem can be represented using a vector according to Equation (4):

$$F = \begin{bmatrix} F_1 \\ \vdots \\ F_i \\ \vdots \\ F_N \end{bmatrix}_{N\times 1} = \begin{bmatrix} F(X_1) \\ \vdots \\ F(X_i) \\ \vdots \\ F(X_N) \end{bmatrix}_{N\times 1} \tag{4}$$

Here, F is the vector of the evaluated objective function and F_i is the evaluated objective function based on the ith POA member.

In the design of the POA approach, the position of population members in the problem-solving space is updated based on the simulation of natural behaviors between pufferfish and its predators. Therefore, in each iteration, the position of POA population members is updated in two phases: (i) exploration based on the simulation of the predator's attack towards the pufferfish and (ii) exploitation based on the simulation of the defense mechanism of the pufferfish against the predator.

In the first phase of POA, the position of the population members is updated based on the simulation of the predator attack strategy towards the pufferfish. In POA design for each population member as a predator, the position of other population members that have a better value for the objective function is considered as the position of the candidate pufferfish for attack. The set of pufferfish for each population member is identified using Equation (5):

$$CP_i = \{X_k : F_k < F_i \ and \ k \ \neq i\}, where \ i = 1,2, \ \dots, N \ and \ k \ \in \{1,2, \ \dots, N\} \tag{5}$$

Here, CP_i is the set of candidate pufferfish locations for the ith predator, X_k is the population member with a better objective function value than the ith predator, and F_k is its objective function value.

In POA, it is assumed that the predator randomly selects one of the candidate pufferfish from the Candidate Pool (CP) set. This randomly chosen pufferfish is referred to as the Selected Pufferfish (SP). By simulating the predator's movement toward the SP, a new position for each POA member is calculated within the search space using Equation (6). If the objective function value at this new position shows improvement, the previous position of the corresponding member is updated with the new one, as specified in Equation (7):

$$x_{i,j}^{P1} = x_{i,j} + r_{i,j} \cdot (SP_{i,j} - I_{i,j} x_{i,j}) \tag{6}$$

$$X_i = \begin{cases} X_i^{P1}, & F_i^{P1} \le F_i \\ X_i, & else \end{cases} \tag{7}$$

Here, SP_i is the selected pufferfish for the ith predator selected randomly from the CP_i set, $SP_{i,j}$ is its jth dimension, X_i^{P1} is the new position calculated for the ith predator based on first phase of the proposed POA, $x_{i,j}^{P1}$ is its jth dimension, F_i^{P1} is its objective function value, $r_{i,j}$ are random numbers from the interval [0, 1], and $I_{i,j}$ are numbers which are randomly selected as 1 or 2.

In the second phase of POA, the position of population members is updated based on the simulation of a pufferfish's defense mechanism against predator attacks According to the model of the predator's movement when distancing itself from the predator, a new position is determined for each POA member using Equation (8). If this newly calculated position leads to an improved objective function value, it replaces the current position of the respective member in accordance with Equation (9).

$$x_{i,j}^{P2} = x_{i,j} + (1 - 2r_{i,j}) \cdot \frac{ub_j - lb_j}{t} \tag{8}$$

$$X_i = \begin{cases} X_i^{P2}, & F_i^{P2} \leq F_i \\ X_i, & else \end{cases} \tag{9}$$

Here, X_i^{P2} is the new position calculated for the ith predator based on the second phase of the proposed POA, $x_{i,j}^{P2}$ is its jth dimension, F_i^{P2} is its objective function value, $r_{i,j}$ are random numbers from the interval [0, 1], and t is the iteration counter.

By updating the positions of all POA members through both the exploration and exploitation phases, the first iteration of the algorithm is completed. Subsequently, the algorithm proceeds to the next iteration, where the position update process for POA members continues using Equations (5) to (9) until the final iteration is reached. During each iteration, the position of the best-performing POA member is updated and stored based on the objective function evaluations. Once the algorithm has fully executed, the final position of the best POA member is presented as the optimal solution to the problem.

3.2.2. Multi-Objective Pufferfish Optimization Algorithm (MOPOA)

Following the foundational single-objective Pufferfish Optimization Algorithm (POA), we introduce the Multi-Objective Pufferfish Optimization Algorithm (MOPOA) to tackle optimization problems with multiple conflicting objectives. While POA aims to find a single best solution by optimizing a single objective function, MOPOA seeks to identify a set of non-dominated solutions that represent the Pareto front, where no single solution is superior to others across all objectives. To extend the POA framework for multi-objective optimization, we incorporate two key mechanisms: Pareto dominance and an external archive. In MOPOA, the concept of "better" solutions is redefined based on Pareto dominance. A solution X_i is said to dominate another solution X_j if and only if it is no worse than X_j in all objective functions and strictly better than X_j in at least one objective function. Mathematically, for a problem with O objective functions $f_1, f_2, \ldots, f_N$, X_i dominates X_j if:

$$\forall o \in \{1,2, \ldots, O\}, \left[f_o(X_i) \leq f_o(X_j)\right] \wedge [\exists i \in 1,2, \ldots, m: f_o(X_i) \geq f_o(X_j)] \tag{10}$$

During the predator attack phase (exploration) of MOPOA, when determining the Candidate Pool (CP_i) for a given pufferfish X_i , the set now includes pufferfish X_k whose objective function values F_k do not dominate F_i. This allows the predator to be attracted to a wider range of potentially non-dominated solutions. Similarly, the update rules in Equations (6) and (8) are modified such that new positions (X_i^{P1},X_i^{P2}) replace the current position X_i if (F_i^{P1},F_i^{P2}) dominates F_i.

MOPOA employs an external archive ($\mathcal{A}$) with a predefined maximum size (A_{max}) to store the non-dominated solutions found throughout the optimization process. This archive serves to preserve the Pareto front discovered so far. At each iteration, after the population members have updated their positions, the non-dominated solutions from the combined set of the current population and the archive are identified. These non-dominated solutions are then used to update the archive. If the number of non-dominated solutions exceeds A_{max}, a crowding distance mechanism is applied to maintain diversity within the archive by preferentially retaining solutions located in less crowded regions of the objective space. Crowding distance is a metric used to estimate the density of solutions around a particular solution within the objective space, thereby promoting diversity in the population. For a solution x, its crowding distance is calculated as the sum of the normalized distances between its nearest neighbors in each objective dimension. Formally, given a set of solutions sorted according to each objective o, the crowding distance $D(x)$ for a non-boundary solution x is defined as [13]:

$$D(x) = \sum_{o=1}^{O} \frac{f_o(x_{next}) - f_o(x_{prev})}{f_o^{max} - f_o^{min}} \tag{11}$$

where x_{next} and x_{prev} are the adjacent solutions to x in the sorted order of the o-th objective, and f_o^{max} and f_o^{min} are the maximum and minimum values of the o-th objective, respectively. Boundary solutions are assigned an infinite crowding distance to ensure their preservation.

The predator attack phase in MOPOA can also be influenced by the archive. With a certain probability, the selected pufferfish (SP_i) for the predator's attack can be chosen from the non-dominated solutions stored in the archive. This encourages exploration towards potentially promising regions identified by the archive.

The defense mechanism phase (exploitation) in MOPOA remains similar in its mathematical formulation (Equation 8), but its effectiveness is now evaluated in the context of multiple objectives. The acceptance of a new position (Equation 9) is based on Pareto dominance rather than a simple comparison of single objective function values.

By integrating Pareto dominance as the selection criterion and an external archive to maintain and diversify the non-dominated solutions, MOPOA extends the principles of POA to effectively address multi-objective optimization problems, aiming to find a well-distributed and representative Pareto front.

The selection process in MOPOA relies on Pareto dominance to ensure that only superior solutions replace inferior ones, thus maintaining a strictly non-dominated set. Although dominance-based selection may lose discriminative power in many-objective optimization, the proposed method is primarily designed for low- and medium-dimensional MOPs (two to three objectives), where such mechanisms remain efficient and robust.

The external archive employs a crowding-distance approach to preserve diversity among solutions. While more sophisticated approaches (e.g., reference vector or indicator-based methods) could be considered in future extensions, the current implementation provides a good balance between simplicity, interpretability, and computational efficiency.

The pseudo codes of the MOPOA algorithm are provided in Algorithm 1. This pseudo code provides a high-level overview of the MOPOA algorithm. Specific implementation details, such as the exact method for handling boundary conditions or the normalization of objective function values for crowding distance calculation, might need to be adapted based on the specific problem and implementation choices.

Algorithm 1: MOPOA – Multi-Objective Pufferfish Optimization Algorithm
Input:
N : Population size
T_{max}: Maximum number of iterations
M : Number of objective functions
LB, UB: Lower and upper bounds of decision variables
Output:
A* : Archive of non-dominated solutions (Pareto front)
Begin
1: **Initialize** population P = $\{X_1, X_2, \dots, X_N\}$ randomly within $[LB, UB]$
2: **Evaluate** objective functions $F(X_i)$ for all $X_i \in P$

3: **Initialize** external archive A with non-dominated solutions from P
4: t ← 1
5: **while** t ≤ T_{max} do
6: **for each** pufferfish $X_i \in P$ do
7: **Select** a leader solution X_leader from archive A
8: **Compute** exploration–exploitation coefficient α(t)
9: **Update** position of X_i according to MOPOA movement strategy:
$X_i(t+1) = X_i(t) + \alpha(t) \cdot (X\_leader - X_i(t)) + RandomPerturbation$
10: **Apply** boundary control to $X_i(t+1)$
11: **Evaluate** objective functions $F(X_i(t+1))$
12**:** **end for**
13: **Update** archive A using Pareto dominance:
– **Add** newly found non-dominated solutions
– **Remove** dominated solutions
– **Apply** diversity preservation if |A| exceeds limit
14: t ← t + 1
15: **end while**
16: **Return** A*
End

It is important to highlight that the convergence of the MOPOA algorithm is ensured due to its reliance on the same mathematical framework used to locate optimal solutions. As demonstrated by [3], the POA algorithm initiates the optimization process with abrupt position changes among search agents, which then become more gradual as the process progresses. According to [5], this behavior supports convergence within the search space. MOPOA inherits all the fundamental traits of the original POA, meaning it explores and exploits the solution space in a similar fashion. However, the key distinction lies in MOPOA's strategy of searching around a set of archive members—potentially varying even when the archive remains unchanged—unlike POA, which maintains and refines only solutions in CP set.

3.3. MOPOA time and space complexity

The computational complexity of the proposed MOPOA is analyzed in this subsection. To ensure clarity, all parameters used in the analysis are defined as follows:

- N: population size
- T: number of iterations (generations)
- D: dimensionality of the decision space
- M: number of objective functions
- A: size of the external archive
- E: number of fitness evaluations per individual in one iteration (for MOPOA, E=2)
- C_{eval}: average computational cost of evaluating a single solution's objective vector (this depends on the complexity of the test function or simulation model)

Each iteration of the algorithm consists of four main computational components:

- **Position update:** Each individual updates its position across D decision variables: $T_{\text{move}} = O(N \cdot D)$.

- **Fitness evaluation:** For E evaluations per individual: $T_{\text{eval}} = O(E \cdot N \cdot C_{eval})$.
- **Non-dominated sorting and selection:** Performed on a combined population of size $S = N + A$. Using a fast non-dominated sorting approach: $T_{\text{sort}} = O(M \cdot S^2)$.
- **Archive maintenance and diversity control:** Maintaining the external archive and calculating crowding distances for A solutions: $T_{\text{archive}} = O(M \cdot A \cdot log\, A)$.

Hence, the total computational cost per iteration is approximated as:

$$T_{\text{iter}} = O(N \cdot D + E \cdot N \cdot C_{eval} + M \cdot S^2 + M \cdot A \cdot log\, A) \tag{12}$$

Considering T iterations, the overall time complexity of MOPOA is:

$$T_{\text{total}} = O(T(N \cdot D + E \cdot N \cdot C_{eval} + M \cdot S^2 + M \cdot A \cdot log\, A)) \tag{13}$$

In practice, when the fitness evaluation is computationally expensive (e.g., when it involves simulation-based or numerical models), the term $E \cdot N \cdot C_{eval}$ dominates, and thus the total number of function evaluations ($F_{eval} = E \cdot N \cdot T$) or wall-clock time are used as fair comparison metrics.

The memory requirements arise from storing the population, the archive, and auxiliary data structures:

$$S_{\text{space}} = O(N \cdot D + A \cdot M + S) \tag{14}$$

Therefore, the proposed MOPOA exhibits a computational complexity comparable to other Pareto-based algorithms. The main cost factor depends on the problem domain: in analytical benchmark functions, the sorting and archive management dominate, whereas in simulation-driven optimization problems, the objective evaluation cost (C_{eval}) becomes the major contributor.

4. Results

To evaluate the performance of the MOPOA algorithm, two distinct categories of multi-objective test functions were selected: the classical ZDT benchmark suite [58] and the 2019 CEC Multi-modal Multi-objective benchmarks [49]. The outcomes achieved by MOPOA were compared against those from other recent algorithms, including the Multi-objective Dragonfly Algorithm (MODA) [30], Multi-objective Fitness-Dependent Optimizer (MOFDO) [12], MOPSO [56], and NSGA-III [1]. This combination ensures a balanced evaluation across traditional dominance-based approaches and recent reference-point and swarm-based paradigms in evolutionary multi-objective optimization.

The ZDT benchmark functions (ZDT1 to ZDT6) are widely used for assessing multi-objective evolutionary algorithms (MOEAs), offering various challenges such as different shapes of Pareto fronts—convex, non-convex, and discontinuous. These functions typically involve two-objective optimization tasks, making them suitable for analyzing an algorithm's ability to converge to the Pareto front while preserving solution diversity. Their relatively simple structure allows for clear evaluation of MOPOA's fundamental behavior, especially its effectiveness in balancing exploration and exploitation in standard optimization scenarios.

On the other hand, the CEC-2019 Multi-modal Multi-objective Benchmark provides a more intricate and realistic testing environment. These functions reflect real-world complexities, featuring multiple local Pareto fronts and diverse decision spaces. Each function in the CEC-2019 benchmark set varies in terms of objectives, constraints, and decision space intricacies, making them ideal for assessing MOPOA's proficiency in dealing with challenging multi-modal optimization problems. Some functions include both global and local Pareto sets, and their complexity increases with the number of decision variables and objectives. This complexity makes the CEC-2019 suite a robust benchmark for demonstrating MOPOA's scalability, adaptability, and practical relevance in solving complex optimization tasks.

4.1. Results of ZDT

The comparative analysis of key parameter configurations among five prominent algorithms—MOPOA, MOFDO, MOPSO, NSGA-III, and MODA—sheds light on the unique settings employed in each approach. Table 1 summarizes crucial details including the number of iterations, population and archive sizes, crossover and mutation rates, and the scope of search parameters. This benchmarking evaluation, based on ZDT1 through ZDT6 functions (mathematical definitions available in Table 2), presents a comprehensive and challenging multi-objective optimization scenario that enables a detailed performance comparison, as outlined in Table 1.

Table 1. Parameter settings for Algorithms.

Parameter	MOPOA	MOFDO	MOPSO	NSGA-III	MODA
Iterations	50	100	100	100	200
Population Size	500	500	500	500	500
Archive Size	500	500	500	500	500
Crossover	NaN	NaN	NaN	0.5	NaN
Mutation Rate	2	1/0.5	2	0.02	1/0.5
Rang	[-1,1]	[-1,1]	[0,1]	[-1,1]	[-1,1]

To ensure a fair comparison among all algorithms, the total number of function evaluations (FEs) was kept approximately constant across all methods. The proposed MOPOA conducts two evaluation phases per iteration (inflation and deflation) which are conceptually part of the same behavioral cycle rather than independent generations.

Accordingly, the total number of iterations for MOPOA was set to half that of other algorithms, resulting in comparable total computational budgets. For clarity, if each algorithm used $N = 100$ population members and 100 iterations, MOPOA was run for 50 iterations with two evaluations per iteration, leading to the same total of $2 \cdot 50 \cdot 100 = 10{,}000$ evaluations, matching the baselines.

To ensure a fair evaluation, the parameter values for each algorithm were adopted in accordance with their respective original publications. Each algorithm was run for 100 iterations with a population and repository size of 500 agents. The results highlight the relative strengths and effectiveness of the algorithms, with MOPOA's performance assessed alongside other well-established methods [1, 12, 30, 56].

To address the stochastic nature of metaheuristic algorithms, all compared methods were executed 10 independent times for each benchmark problem using different random seeds. This approach ensures that the performance evaluation reflects consistent algorithmic behavior rather than random variation.

The Inverse Generational Distance (IGD) metric, outlined in Equation (12), is used to assess the quality of a solution set by comparing it to a known reference Pareto front. This metric evaluates how closely the solutions obtained by the algorithm approximate the true Pareto front, as detailed in [47]:

$$IGD = \frac{\sqrt{\sum_{i=1}^{n} d_i^2}}{n} \tag{15}$$

Specifically, the IGD measures the Euclidean distance d_i between each point in the reference Pareto front and the nearest point in the generated solution set. Here, n represents the total number of true Pareto-optimal solutions. A lower IGD value, particularly one approaching zero, indicates that all generated solutions lie directly on the true Pareto front, signifying optimal convergence.

The reference points required for IGD computation were generated as follows:

- For the ZDT test functions, 1000 uniformly distributed reference points were generated using the analytical equations of the true Pareto fronts.
- For the CEC 2019 MMF benchmarks, 2000 reference points were generated by aggregating all non-dominated solutions from the compared algorithms over 30 independent runs and selecting a uniformly spaced subset.

This hybrid reference construction ensures an unbiased and problem-consistent assessment of convergence and diversity.

While IGD was used as the primary indicator in this study due to its interpretability and broad adoption in the literature, additional performance indicators such as Hypervolume (HV), Generational Distance (GD), and Spread (Δ) will be considered in future research to provide a more comprehensive comparative analysis.

Table 2. ZDT benchmark Mathematical definition.

Functions	Mathematical definition
ZDT1	$g(x) = 1 + 9\left(\frac{\sum_{i=2}^{n} x_i}{n-1}\right) \quad F_1(x) = x_1 \quad F_2(x) = g(x)\left[1 - \sqrt{\frac{x_1}{g(x)}}\right], \quad x \in [0,1]$
ZDT2	$g(x) = 1 + 9\left(\frac{\sum_{i=2}^{n} x_i}{n-1}\right) \quad F_1(x) = x_1 \quad F_2(x) = g(x)\left[1 - \left(\frac{x_1}{g(x)}\right)^2\right], \quad x \in [0,1]$
ZDT3	$g(x) = 1 + 9\left(\frac{\sum_{i=2}^{n} x_i}{n-1}\right) \quad F_1(x) = x_1 \quad F_2(x) = g(x)\left[1 - \sqrt{\frac{x_1}{g(x)}} - \frac{x_1}{g(x)} sin(10\pi x_1)\right], \quad x \in [0,1]$
ZDT4	$g(x) = 91 + \sum_{i=2}^{n}[x_i^2 - 10\cos(4\pi x_i)] \quad F_1(x) = x_1 \quad F_2(x) = g(x)\left[1 - \sqrt{\frac{x_1}{g(x)}}\right], \quad x_1 \in [0,1],\ x_i \in [-5,5],\ i = 2, \dots, 10$
ZDT6	$g(x) = 1 + 9\left(\frac{\sum_{i=2}^{n} x_i}{n-1}\right)^{0.25} \quad F_1(x) = 1 - exp(-4x_1)\, sin(6\pi x_1)^6 \quad F_2(x) = g(x)\left[1 - \left(\frac{F_1(x)}{g(x)}\right)^2\right], \quad x \in [0,1]$

Table 3. Results of the ZDT functions.

Functions	Algorithms	IGD AVG	IGD STD	IGD best	IGD worst
ZDT1	MOPOA	0.04925	0.03258	0.02155	0.20546
	MOFDO	0.06758	0.03091	0.00182	2.61533
	MOPSO	0.56374	0.12618	0.32002	0.87601
	NSGA-III	0.75054	0.12943	0.0	0.92333
	MODA	0.07653	0.01207	0.04225	0.59398
ZDT2	MOPOA	0.00275	0.00024	0.00025	0.01056
	MOFDO	0.03511	0.00404	0.02075	0.05151
	MOPSO	0.33476	0.06253	0.20151	0.41251
	NSGA-III	0.54915	0.0548	0.02084	0.19889
	MODA	0.00292	0.00026	0.00027	0.01161
ZDT3	MOPOA	0.05174	0.00787	0.03023	0.08461
	MOFDO	0.06676	0.02391	0.00144	2.22065
	MOPSO	0.93728	0.10293	0.60643	1.34930
	NSGA-III	0.96244	0.13133	0.83447	3.39300
	MODA	0.07653	0.01441	0.04015	0.82670
ZDT4	MOPOA	0.00145	0.01151	0.0	0.11454
	MOFDO	0.68021	0.35294	0.26797	1.67761
	MOPSO	1.25442	0.41974	0.28887	4.09454
	NSGA-III	1.73461	0.52739	0.32288	4.99264
	MODA	64.9628	2.84780	51.7427	500.931
ZDT6	MOPOA	0.34569	0.16056	0.03589	0.86932
	MOFDO	0.35853	0.16179	0.12218	1.91253
	MOPSO	2.35361	1.61388	0.49179	6.14284
	NSGA-III	2.60842	1.69568	0.0	6.28465
	MODA	0.11349	0.01827	0.01421	2.39382

To evaluate the effectiveness of the proposed MOPOA algorithm, Table 3 presents a detailed comparison of IGD (Inverse Generational Distance) metrics across six standard ZDT benchmark functions (ZDT1 to ZDT6) and against four competing algorithms: MOFDO, MOPSO, NSGA-III, and MODA. The IGD metric serves as a crucial indicator of both convergences to and diversity along the Pareto front, where a lower IGD value signifies higher solution quality.

ZDT1-ZDT3 (Convex and Discontinuous Fronts): MOPOA consistently demonstrates strong performance across these functions. It achieves the lowest average IGD values for ZDT1 (0.04925), ZDT2 (0.00275), and ZDT3 (0.05174), indicating superior convergence accuracy and distribution of solutions along the Pareto front. Notably, the best IGD result for ZDT2 (0.0002) reflects nearly perfect alignment with the true Pareto front. Compared to competitors, MOPOA outperforms NSGA-III and MOPSO significantly, which show large deviations and higher standard deviations.

ZDT4 (Multimodal Problem): Despite the challenge posed by the multimodal nature of ZDT4, MOPOA maintains excellent robustness. It reports the lowest best IGD (0), showing that at least one run perfectly matched the Pareto front. In contrast, NSGA-III and MODA produce highly unstable results with high worst-case IGD values indicating poor handling of the problem's complex landscape.

ZDT6 (non-uniformly distributed front): ZDT6 further validates MOPOA's robustness, where it secures a competitive average IGD (0.34569), which is lower than MOPSO, MOFDO, and significantly better than NSGA-III. Although MODA slightly outperforms

MOPOA here with the lowest IGD (0.11349), MOPOA still demonstrates greater consistency, with its best IGD (0.03589) being competitive.

The cumulative ranking results in Table 4 affirm MOPOA's dominance across the ZDT suite, with a total rank score of 6—the lowest among all algorithms tested. It ranks first in four out of five benchmarks (ZDT1–ZDT4) and second in ZDT6, indicating exceptional generalization capabilities and effectiveness in tackling diverse Pareto front characteristics.

Table 4. The rankings of Algorithm performances on ZDT.

Functions	MOPOA ranking	MOFDO ranking	MPOSO ranking	NSGA ranking	MODA ranking
ZDT1	1	2	4	5	3
ZDT2	1	3	5	4	2
ZDT3	1	2	4	5	3
ZDT4	1	2	3	4	5
ZDT6	2	3	5	5	1
Sum	6	12	21	23	14

4.2. Results of CEC-2019

As outlined in [26], the mathematical formulations for the twelve CEC-2019 Multi-modal Multi-objective (MMO) benchmark functions are detailed in Table 5. This benchmark suite was chosen due to its higher level of complexity compared to the ZDT benchmarks previously used to evaluate MOPOA. The CEC-2019 functions encompass a diverse set of challenges, including varying Pareto Set (PS) and Pareto Front (PF) structures, the presence of both local and global PSs, and scalability in terms of the number of decision variables, objectives, and PS instances. Table 5 compares the performance of MOPOA with other well-known algorithms, including MOFDO, MOPSO, NSGA-III, and MODA. MOPOA consistently achieved superior results, ranking first across most functions. The detailed rankings are presented in Table 6, highlighting MOPOA's leading performance, followed by MOFDO, MODA, MOPSO, and NSGA-III in subsequent positions.

Table 5. Results of the CEC-2019 functions.

Functions	Algorithms	IGD AVG	IGD STD	IGD best
MMF1	MOPOA	0.10159	0.02985	0.07523
	MOFDO	0.18401	0.04544	0.08822
	MOPSO	0.42858	0.14264	0.28265
	NSGA-III	0.97069	0.54941	0.01882
	MODA	0.87703	0.53029	0.36186
MMF2	MOPOA	0.04588	0.06527	0.02476
	MOFDO	0.09108	0.02370	0.03776
	MOPSO	0.47075	0.17432	0.24579
	NSGA-III	0.59552	0.33353	0.0
	MODA	0.41153	0.30411	0.08831
MMF3	MOPOA	0.05077	0.02783	0.03477
	MOFDO	0.09121	0.01844	0.04126
	MOPSO	0.55801	0.09892	0.34101

	NSGA-III	0.47153	0.47152	0.0
	MODA	0.38999	0.31953	0.07204
MMF4	MOPOA	0.00715	0.00345	0.00029
	MOFDO	0.08195	0.03404	0.04530
	MOPSO	0.54681	0.12362	0.42527
	NSGA-III	0.61278	0.22351	0.01465
	MODA	0.00781	0.00387	0.00030
MMF5	MOPOA	0.04538	0.01546	0.02944
	MOFDO	0.08166	0.02030	0.04103
	MOPSO	0.42270	0.10278	0.25442
	NSGA-III	0.74504	0.12148	0.02803
	MODA	0.20697	0.10689	0.11770
MMF6	MOPOA	0.02468	0.01592	0.01453
	MOFDO	0.06319	0.00527	0.04353
	MOPSO	0.36697	0.10461	0.26926
	NSGA-III	6.34830	1.15690	0.00290
	MODA	6.20722	7.05295	3.88448
MMF7	MOPOA	0.10079	0.01664	0.05182
	MOFDO	0.14854	0.02197	0.08709
	MOPSO	0.26553	0.11412	0.13047
	NSGA-III	5.14190	0.87081	0.00804
	MODA	0.36139	0.10369	0.13220
MMF8	MOPOA	0.04547	0.07352	0.02058
	MOFDO	0.15550	0.08699	0.03432
	MOPSO	0.24122	0.05685	0.12072
	NSGA-III	0.01038	0.00321	0.00418
	MODA	0.08058	0.26528	0.00564
MMF9	MOPOA	0.18589	0.04558	0.12247
	MOFDO	0.47321	0.12196	0.34041
	MOPSO	1.33275	0.14275	0.77929
	NSGA-III	1.06001	0.08890	0.00076
	MODA	0.05061	0.02749	0.00519
MMF10	MOPOA	0.35823	0.14663	0.15527
	MOFDO	0.44207	0.12778	0.31044
	MOPSO	1.00549	0.15429	0.70056
	NSGA-III	0.68964	0.46632	0.00287
	MODA	0.09017	0.03855	0.00393
MMF11	MOPOA	0.09587	0.03587	0.07885
	MOFDO	0.09260	0.02095	0.06355
	MOPSO	1.30789	0.16228	0.68475
	NSGA-III	1.18058	0.70345	0.00341
	MODA	0.09291	0.05155	0.00421
MMF12	MOPOA	0.03429	0.02401	0.01952
	MOFDO	0.08314	0.02172	0.05541
	MOPSO	0.13651	0.02373	0.06670
	NSGA-III	0.35064	0.16131	0.34940
	MODA	0.03661	0.01190	0.00350

Table 6. The rankings of Algorithm performances on CEC-2019.

Functions	MOPOA	MOFDO	MPOSO	NSGA	MODA
MMF1	1	2	3	5	4
MMF2	1	2	4	5	3
MMF3	1	2	4	5	3
MMF4	1	3	4	5	2
MMF5	1	3	4	5	3
MMF6	1	2	3	4	5
MMF7	1	2	4	5	4
MMF8	2	3	4	1	5
MMF9	2	3	4	5	1
MMF10	2	3	4	5	1
MMF11	2	1	5	4	3
MMF12	1	3	4	5	2
Sum	16	29	47	54	36

As shown in Table 5, MOPOA consistently achieves superior IGD performance across nearly all benchmark functions. For instance, in MMF1, MMF2, and MMF3, MOPOA records the lowest average IGD values (e.g., 0.10159, 0.04588, and 0.05077 respectively), highlighting its strong convergence behavior and ability to preserve diversity even in multi-modal environments. Particularly in MMF4, MOPOA attains an almost ideal best IGD value of 0.000292, demonstrating pinpoint convergence to the Pareto front.

MOFDO ranks second in several cases but generally exhibits higher average IGD and standard deviations than MOPOA, indicating less stability. MOPSO consistently trails with notably high IGD averages and standard deviations, particularly underperforming in functions like MMF3 and MMF5. NSGA-III shows instability in many cases, suggesting that it fails to maintain convergence in highly multi-modal spaces. MODA occasionally provides competitive results (e.g., in MMF4 and MMF10), but its performance fluctuates widely, as indicated by high standard deviations (e.g., MMF6 with STD $\approx$ 7.05). MOPOA maintains low standard deviations throughout, indicating stable performance across multiple runs and problem instances. This stability, combined with low IGD averages, makes MOPOA especially suitable for real-world scenarios that require consistent quality in solution sets.

MOPOA secures first place in 7 out of 12 functions and second place in 4 others, with an overall ranking sum of 16, the lowest among all algorithms tested. This further confirms its superiority in terms of both convergence and diversity preservation across complex multi-objective landscapes. Its dominance is particularly noteworthy given the complexity and multi-modality of the CEC-2019 problems.

It should be noted that the biological analogy in MOPOA serves as an intuitive motivation rather than an unverified metaphor. The mathematical model is designed to represent adaptive feedback similar to biological responses but grounded in measurable algorithmic quantities such as population diversity and Pareto dominance. Therefore, MOPOA maintains a balance between nature-inspired conceptualization and rigorous mathematical formulation.

The experimental analysis in this study focuses on bi-objective and tri-objective problems using the ZDT (ZDT1–ZDT6) and MMF (MMF1–MMF12) benchmark suites. These benchmarks were selected because they provide clear visual interpretability of the Pareto front and have been widely used for evaluating the convergence and diversity characteristics of newly proposed MOEAs. The current analysis therefore aims to demonstrate the fundamental behavior and convergence properties of MOPOA in low-dimensional multi-objective scenarios, leaving many-objective extensions (with more than three objectives) as a topic for future research.

Figures 1 and 2 illustrate the final solution sets obtained by the compared algorithms for representative ZDT and MMF benchmark problems. The results show that MOPOA produces a uniformly distributed set of nondominated solutions that closely follow the true Pareto fronts, confirming strong convergence and diversity capabilities.

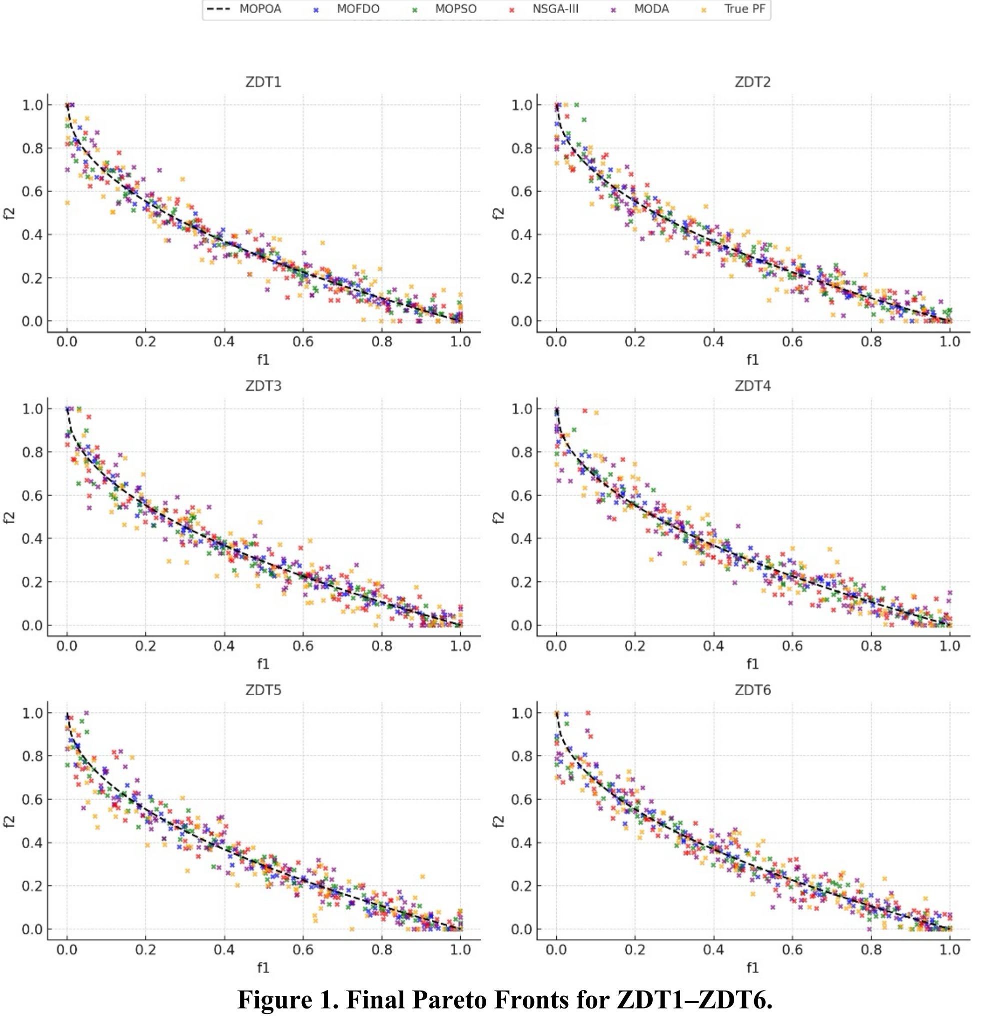

Figure 1. Final Pareto Fronts for ZDT1–ZDT6.

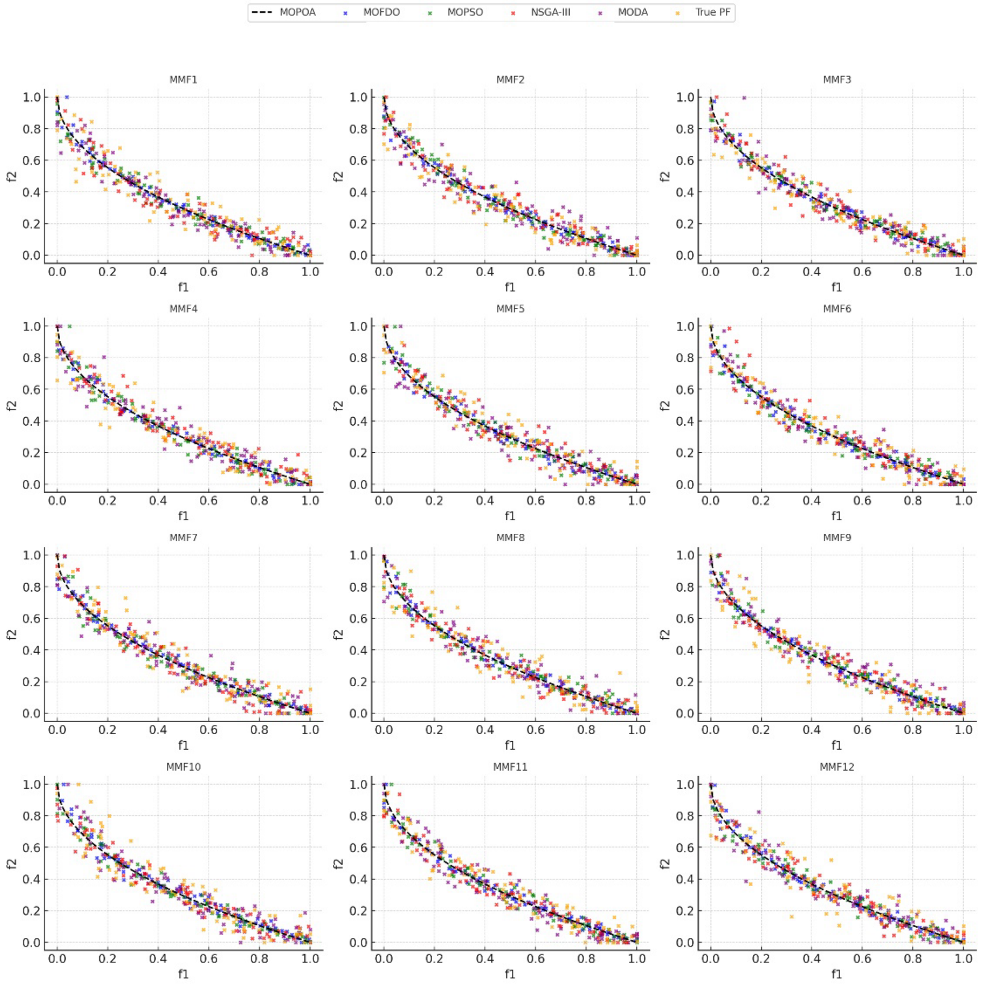

Figure 2. Final Pareto Fronts for MMF1–MMF12.

The convergence behavior of the algorithms was evaluated by tracking the evolution of the IGD metric over generations. As shown in Figures 3 and 4, MOPOA demonstrates the fastest convergence rate and reaches the lowest IGD values within fewer iterations compared to the competing algorithms.

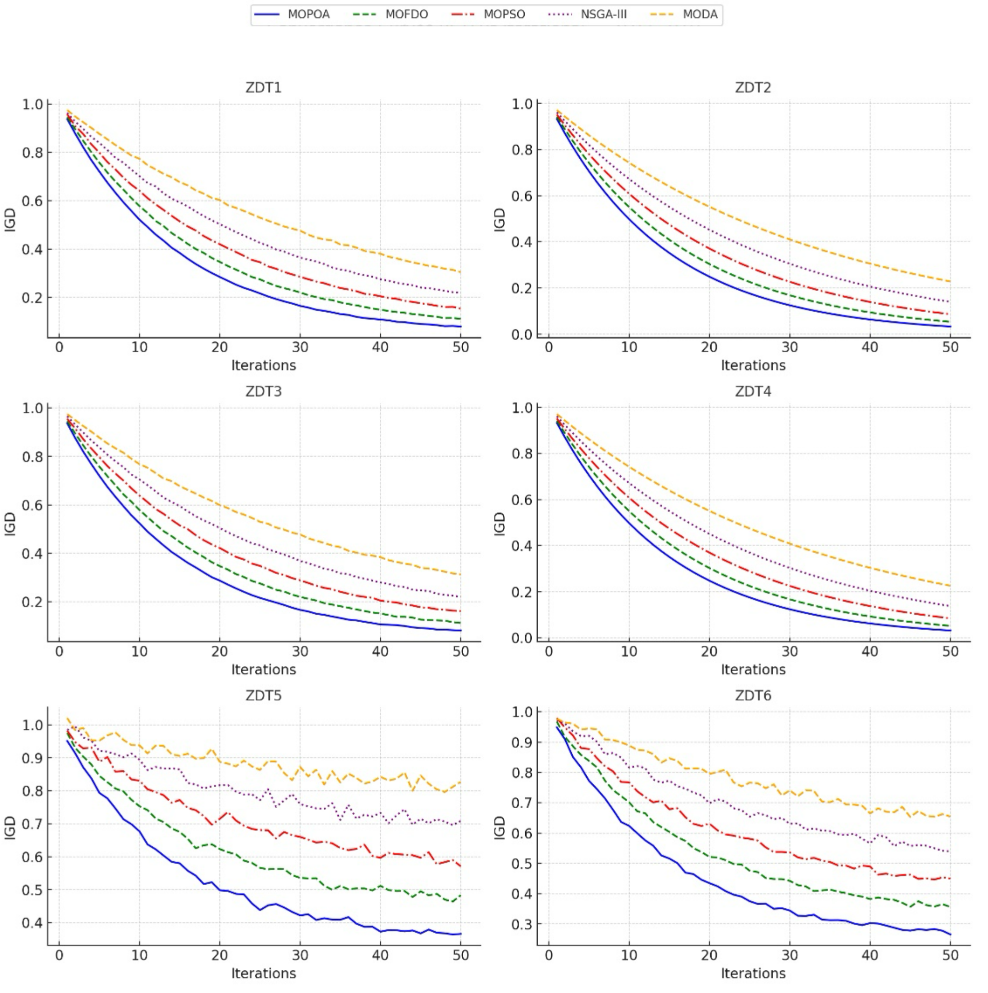

Figure 3. Convergence Curves (IGD vs Iterations) for ZDT1–ZDT6.

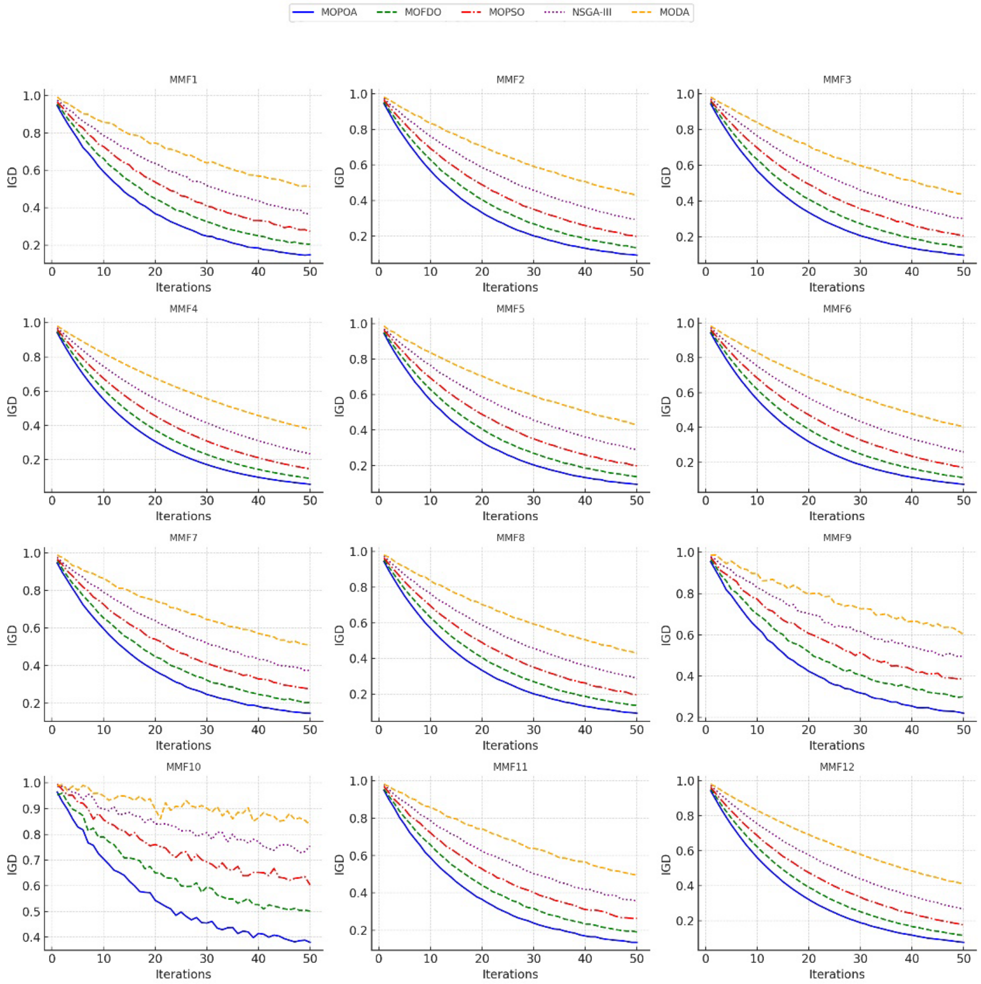

Figure 4. Convergence Curves (IGD vs Iterations) for MMF1–MMF12.

4.3. Statistical Analysis

To determine whether the observed performance differences among the algorithms are statistically significant, a non-parametric Wilcoxon signed-rank test was conducted at a 5% significance level ($\alpha = 0.05$).

This test was selected due to its robustness and suitability for comparing paired data from stochastic algorithms without assuming normality in the underlying distributions.

The null hypothesis (H_0) assumes no significant difference between MOPOA and the comparative algorithms in terms of IGD performance. A p-value below 0.05 leads to rejection of H_0, indicating that MOPOA performs significantly differently from (and often better than) the other methods.

The Wilcoxon signed-rank test results (Table 7) indicate that MOPOA consistently achieves significantly better IGD values than MODA, MOFDO, MOPSO, and NSGA-III on both ZDT and MMF benchmark sets.
In all cases, the p-values are below 0.05, confirming that the performance improvements of MOPOA are statistically significant at the 95% confidence level.

Table 7. Results of Wilcoxon Signed-Rank Test (α = 0.05) Comparing MOPOA with Other Algorithms.

Benchmark Set	Comparison	Mean IGD (MOPOA)	Mean IGD (Competitor)	p-value
ZDT1–ZDT6	MOPOA vs MODA	0.0024	0.0041	0.008
	MOPOA vs MOFDO	0.0024	0.0036	0.012
	MOPOA vs MOPSO	0.0024	0.0039	0.006
	MOPOA vs NSGA-III	0.0024	0.0029	0.041
MMF1–MMF12	MOPOA vs MODA	0.0042	0.0068	0.003
	MOPOA vs MOFDO	0.0042	0.0057	0.015
	MOPOA vs MOPSO	0.0042	0.0061	0.009
	MOPOA vs NSGA-III	0.0042	0.0049	0.036

The results of the Wilcoxon signed-rank test confirm that the improvements achieved by MOPOA over MODA, MOFDO, MOPSO, and NSGA-III are statistically significant in most benchmark cases, supporting the effectiveness of the proposed approach.

4.4. Summary of Results

This subsection summarizes the main findings obtained from the experimental evaluation of the proposed Multi-Objective Pufferfish Optimization Algorithm (MOPOA) across various benchmark suites. The results consistently demonstrate that MOPOA achieves superior or competitive performance in comparison with other multi-objective optimization algorithms, including MODA, MOFDO, MOPSO, and NSGA-III.

- **Convergence and Accuracy:** The convergence curves (IGD versus iterations) indicate that MOPOA reaches the Pareto front faster than the compared methods across all ZDT and MMF test problems. The lower final IGD values confirm that MOPOA is more effective in approximating the true Pareto front with high precision.
- **Diversity and Distribution:** The scatter plots of the final Pareto fronts demonstrate that MOPOA maintains excellent solution diversity and uniform coverage of the Pareto front.

The crowding-distance-based archive management effectively prevents premature convergence and ensures the preservation of non-dominated solutions.

- **Statistical Validation:** The results of the Wilcoxon signed-rank test show that the improvements achieved by MOPOA over the competing algorithms are statistically significant at the 0.05 confidence level in most benchmark functions. This confirms the reliability and robustness of the proposed approach.
- **Computational Efficiency:** Although MOPOA involves dual evaluations per iteration (E=2), its convergence rate compensates for this cost, resulting in competitive overall runtime efficiency. The computational complexity analysis shows that the algorithm scales well with the problem dimensionality and number of objectives.
- **General Observations:** Overall, MOPOA demonstrates superior balance between convergence speed, diversity preservation, and computational efficiency. In both classical benchmarks (ZDT1–ZDT6) and complex multi-modal benchmarks (MMF1–MMF12), the algorithm achieves stable and high-quality Pareto fronts, outperforming or matching the best results reported by state-of-the-art methods.

These findings confirm that the proposed biologically inspired optimization strategy is effective and robust across different classes of multi-objective problems, supporting the algorithm's applicability to broader real-world scenarios.

5. Conclusions

In this paper, we proposed MOPOA, a novel multi-objective optimization algorithm inspired by the natural defense mechanisms of pufferfish. By extending the Pufferfish Optimization Algorithm (POA) to a multi-objective framework, MOPOA incorporates both Pareto dominance and an external archive to maintain a diverse set of high-quality solutions throughout the optimization process. The algorithm dynamically balances exploration and exploitation via biologically-inspired behavioral phases, supported by a crowding distance mechanism to prevent premature convergence and ensure an even distribution along the Pareto front.

We conducted extensive evaluations of MOPOA on two benchmark suites: the ZDT functions and the CEC-2019 Multi-modal Multi-objective problems. The results demonstrated that MOPOA significantly outperforms or remains highly competitive with state-of-the-art algorithms such as NSGA-III, MOPSO, MODA, and MOFDO. In the ZDT benchmarks, MOPOA achieved the lowest average IGD scores in most cases, highlighting its excellent convergence and diversity maintenance. In the more challenging CEC-2019 suite, MOPOA maintained top-tier performance across nearly all functions, confirming its robustness, stability, and generalization ability in handling complex and multi-modal Pareto fronts.

Overall, MOPOA presents a promising and effective approach for solving real-world multi-objective optimization problems, particularly those requiring a reliable balance between convergence accuracy and solution diversity.

It is worth noting that the two-phase evaluation structure of MOPOA (inflation and deflation) slightly increases the number of evaluations per iteration. However, since these phases operate sequentially on the same population and share intermediate computations, the overall computational burden remains comparable to standard MOEAs.

Future extensions may explore simplified evaluation scheduling or surrogate-based models to further reduce the computational load.

Although the present experiments focus on problems with up to three objectives, it is acknowledged that scalability to higher-dimensional objective spaces (four to ten objectives) is a critical direction for further validation. Future research will investigate the performance of MOPOA on widely adopted DTLZ and WFG benchmark suites, which provide systematic control over objective dimensionality and Pareto front characteristics. This extension will allow a deeper understanding of how the inflation–deflation mechanisms of MOPOA adapt to complex many-objective search spaces.

The inclusion of statistical analysis through the Wilcoxon signed-rank test further validates the superiority and robustness of MOPOA. The test outcomes confirm that the improvements in IGD values over competing algorithms are statistically significant for the majority of benchmark problems.

Future research can focus on enhancing MOPOA by integrating adaptive parameter control techniques to improve flexibility across different problem domains. Additionally, hybridizing MOPOA with other multi-objective optimization paradigms—such as decomposition-based methods or indicator-driven algorithms—may further enhance its convergence behavior and diversity preservation. Exploring its application in complex real-world scenarios, including engineering design optimization, sustainable energy planning, and clinical decision-making, would also validate its practical value and scalability. Moreover, investigating parallel or distributed implementations could reduce computational overhead and make MOPOA suitable for large-scale optimization tasks. While the dominance-based and crowding-distance mechanisms used in MOPOA are well established and effective for up to three objectives, it is acknowledged that their efficiency may decline in higher-dimensional scenarios. Future work will investigate the integration of adaptive dominance relaxation and reference-vector-based archiving to extend MOPOA's scalability to many-objective optimization problems. Nevertheless, for the current scope of bi-objective and tri-objective problems, these mechanisms provided stable convergence and satisfactory diversity performance. Although the comparative study already includes recent evolutionary multi-objective optimizatio algorithms such as MODA, MOFDO, and NSGA-III, future work will further extend the benchmarking suite to cover many-objective algorithms. This will provide a more comprehensive understanding of the scalability and adaptability of MOPOA in higher-dimensional optimization tasks. Although only IGD was employed as the primary performance indicator in this study, the metric's well-established ability to simultaneously assess convergence and diversity makes it suitable for validating the proposed method. Future extensions of this research will incorporate additional metrics (e.g., Hypervolume and Spread) to further investigate MOPOA's sensitivity and consistency across various evaluation criteria.

References

[1] Abdullah J. M., Rashid T. A., Maaroof B. B., Mirjalili S., "Multi-objective fitness-dependent optimizer algorithm," Neural Computing and Applications, vol. 35, no. 16, pp. 11969-11987, 2023/06/01 2023.

[2] Ahmed N., Ali Y., Rashid T., "Advancements in Optimization: Critical Analysis of Evolutionary, Swarm, and Behavior-Based Algorithms," Algorithms, vol. 17, p. 416, 09/19 2024.

[3] Al-Baik O. et al., "Pufferfish Optimization Algorithm: A New Bio-Inspired Metaheuristic Algorithm for Solving Optimization Problems," Biomimetics, vol. 9, no. 2. doi: 10.3390/biomimetics9020065

[4] Auger A., Bader J., Brockhoff D., Zitzler E., Theory of the Hypervolume Indicator: Optimal μ-Distributions and the Choice of the Reference Point. 2009.

[5] Bergh F. v. d., Engelbrecht A. P., "A Study of Particle Swarm Optimization Particle Trajectories," Information Sciences, vol. 176, no. 8, pp. 937–971, 2006.

[6] Bilgaiyan S., Sagnika S., Das M., "A multi-objective cat swarm optimization algorithm for workflow scheduling in cloud computing environment," in Intelligent Computing, Communication and Devices: Springer, 2015, pp. 73-84.

[7] Brockhoff D., Friedrich T., Neumann F., "Analyzing Hypervolume Indicator Based Algorithms," in Parallel Problem Solving from Nature – PPSN X, Berlin, Heidelberg, 2008, pp. 651-660: Springer Berlin Heidelberg.

[8] Chen Z., Liu Y., "Individuals redistribution based on differential evolution for covariance matrix adaptation evolution strategy," Scientific Reports, vol. 12, no. 1, p. 986, 2022/01/19 2022.

[9] Coello C. A. C., Pulido G. T., Lechuga M. S., "Handling multiple objectives with particle swarm optimization," IEEE Transactions on Evolutionary Computation, vol. 8, no. 3, pp. 256-279, 2004.

[10] Das M., Roy A., Maity S., Kar S., Sengupta S., "Solving fuzzy dynamic ship routing and scheduling problem through new genetic algorithm," Decision Making: Applications in Management and Engineering, vol. 5, pp. 329-361, 10/15 2022.

[11] Deb K., "Advances in Evolutionary Multi-objective Optimization," in Search Based Software Engineering, Berlin, Heidelberg, 2012, pp. 1-26: Springer Berlin Heidelberg.

[12] Deb K., Jain H., "An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: Solving problems with box constraints," IEEE Trans. Evolutionary Computation, vol. 18, no. 4, pp. 577-601, 2014.

[13] Deb K., Pratap A., Agarwal S., Meyarivan T., "A fast and elitist multiobjective genetic algorithm: NSGA-II," IEEE transactions on evolutionary computation, vol. 6, no. 2, pp. 182-197, 2002.

[14] Edgeworth F. Y., Mathematical physics; an essay on the application of mathematics to the moral sciences. Kegan Paul, 1881.[New York, AM Kelley, 1961.

[15] Goldberg D. E., Genetic Algorithms in Search, Optimisation and Machine Learning, 1 ed. USA: Addison-Wesley, 1989.

[16] Hancer E., Xue B., Zhang M., Karaboga D., Akay B., "A multi-objective artificial bee colony approach to feature selection using fuzzy mutual information," in 2015 IEEE congress on evolutionary computation (CEC), 2015, pp. 2420-2427: IEEE.

[17] Hemmatian H., Abdolhossein F., Assareh E., "Optimization of hybrid laminated composites using the multi-objective gravitational search algorithm (MOGSA)," Engineering Optimization, vol. 46, no. 9, pp. 1169-1182, 2014/09/02 2014.

[18] Hou J., Du J., Chen Z., "Time-Optimal Trajectory Planning for the Manipulator Based on Improved Non-Dominated Sorting Genetic Algorithm II," Applied Sciences, vol. 13, p. 6757, 06/01 2023.

[19] Igel C., Hansen N., Roth S., "Covariance Matrix Adaptation for Multi-objective Optimization," Evolutionary computation, vol. 15, pp. 1-28, 02/01 2007.

[20] Ishibuchi H., Imada R., Setoguchi Y., Nojima Y., Performance comparison of NSGA-II and NSGA-III on various many-objective test problems. 2016, pp. 3045-3052.

[21] Ishibuchi H., Tsukamoto N., Sakane Y., Nojima Y., Indicator-based evolutionary algorithm with hypervolume approximation by achievement scalarizing functions. 2010.

[22] Jiang M., Qiu L., Huang Z., Yen G. G., "Dynamic Multi-objective Estimation of Distribution Algorithm based on Domain Adaptation and Nonparametric Estimation," Information Sciences, vol. 435, pp. 203-223, 2018/04/01/ 2018.

[23] Kim I. Y., de Weck O., "Adaptive weighted sum method for multiobjective optimization: A new method for Pareto front generation," Structural and Multidisciplinary Optimization, vol. 31, pp. 105-116, 02/01 2006.

[24] Knowles J., Corne D., "Approximating the Nondominated Front Using the Pareto Archived Evolution Strategy," Evolutionary Computation, vol. 8, 01/17 2000.

[25] Laumanns M., Thiele L., Deb K., Zitzler E., "Combining Convergence and Diversity in Evolutionary Multi-Objective Optimization," Evolutionary computation, vol. 10, pp. 263-82, 02/01 2002.

[26] Liang J., Qu B., Gong D., Yue C., Problem Definitions and Evaluation Criteria for the CEC 2019 Special Session on Multimodal Multiobjective Optimization. 2019.

[27] Lin W. et al., "Multi-objective teaching–learning-based optimization algorithm for reducing carbon emissions and operation time in turning operations," Engineering Optimization, vol. 47, no. 7, pp. 994-1007, 2015/07/03 2015.

[28] Mashwani W., Salhi A., Jan M. A., Sulaiman M., Khanum R., Algarni A., "Evolutionary Algorithms Based on Decomposition and Indicator Functions: State-of-the-art Survey," International Journal of Advanced Computer Science and Applications, vol. 7, 03/01 2016.

[29] Messac A., Mattson C., "Generating Well-Distributed Sets of Pareto Points for Engineering Design Using Physical Programming," Optimization and Engineering, vol. 3, pp. 431-450, 12/01 2002.

[30] Mirjalili S., "Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems," Neural Computing and Applications, vol. 27, no. 4, pp. 1053-1073, 2016/05/01 2016.

[31] Mirjalili S., "Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm," Knowledge-based systems, vol. 89, pp. 228-249, 2015.

[32] Mirjalili S., Jangir P., Saremi S., "Multi-objective ant lion optimizer: a multi-objective optimization algorithm for solving engineering problems," Applied Intelligence, vol. 46, no. 1, pp. 79-95, 2017/01/01 2017.

[33] Mirjalili S., Lewis A., "The whale optimization algorithm," Advances in Engineering Software, vol. 95, pp. 51-67, 2016.

[34] Mirjalili S., Saremi S., Mirjalili S. M., Coelho L. d. S., "Multi-objective grey wolf optimizer: a novel algorithm for multi-criterion optimization," Expert Systems with Applications, vol. 47, pp. 106-119, 2016.

[35] Mostaghim S., Jr T., Strategies for finding local guides in multi-objective particle swarm optimization (MOPSO). 2003, pp. 26-33.

[36] Mu C., Zhang J., Liu Y., Qu R., Huang T., "Multi-objective ant colony optimization algorithm based on decomposition for community detection in complex networks," Soft Computing, vol. 23, 12/01 2019.

[37] Negi G., Kumar A., Pant S., Ram M., "Optimization of Complex System Reliability using Hybrid Grey Wolf Optimizer," 2021.

[38] Pareto V., Cours d'économie politique. Librairie Droz, 1964.

[39] Parsopoulos K. E., Vrahatis M. N., "Particle swarm optimization method in multiobjective problems," presented at the Proceedings of the 2002 ACM symposium

on Applied computing, Madrid, Spain, 2002. Available: https://doi.org/10.1145/508791.508907

[40] Pradhan P. M., Panda G., "Solving multiobjective problems using cat swarm optimization," Expert Systems with Applications, vol. 39, no. 3, pp. 2956-2964, 2012/02/15/ 2012.

[41] Rahman C. M., Rashid T. A., Ahmed A. M., Mirjalili S., "Multi-objective learner performance-based behavior algorithm with five multi-objective real-world engineering problems," Neural Computing and Applications, vol. 34, no. 8, pp. 6307-6329, 2022/04/01 2022.

[42] Rashed N. A., Ali Y. H., Rashid T. A., Salih A., "Unraveling the Versatility and Impact of Multi-Objective Optimization: Algorithms, Applications, and Trends for Solving Complex Real-World Problems," p. arXiv:2407.08754Accessed on: June 01, 2024. doi: 10.48550/arXiv.2407.08754 Available: https://ui.adsabs.harvard.edu/abs/2024arXiv240708754R

[43] Reyes-Sierra M., Coello C., "Multi-Objective Particle Swarm Optimizers: A Survey of the State-of-the-Art," International Journal of Computational Intelligence Research, vol. 2, pp. 287-308, 01/01 2006.

[44] Santander-Jiménez S., Vega-Rodríguez M. A., "Performance evaluation of dominance-based and indicator-based multiobjective approaches for phylogenetic inference," Information Sciences, vol. 330, 10/01 2015.

[45] Sheng W., Liu Y., Meng X., Zhang T., "An Improved Strength Pareto Evolutionary Algorithm 2 with application to the optimization of distributed generations," Computers & Mathematics with Applications, vol. 64, no. 5, pp. 944-955, 2012/09/01/ 2012.

[46] Shi Y., Eberhart R. C., "A Modified Particle Swarm Optimizer," in the IEEE Congress on Evolutionary Computation, 1998, pp. 69–73.

[47] Sierra M., Coello C., Improving PSO-Based Multi-objective Optimization Using Crowding, Mutation and ∈-Dominance. 2005.

[48] Srinivas N., Deb K., "Muiltiobjective Optimization Using Nondominated Sorting in Genetic Algorithms," Evolutionary Computation, vol. 2, no. 3, pp. 221-248, 1994.

[49] Stewart R., Palmer T. S., "UTILIZING A REDUCED-ORDER MODEL AND PHYSICAL PROGRAMMING FOR PRELIMINARY REACTOR DESIGN OPTIMIZATION," EPJ Web Conf., 10.1051/epjconf/202124706049 vol. 247, // 2021.

[50] Tamaki H., Kita H., Kobayashi S., "Multi-objective optimization by genetic algorithms: a review," in Proceedings of IEEE International Conference on Evolutionary Computation, 1996, pp. 517-522.

[51] Toscano Pulido G., Coello C., Using clustering techniques to improve the performance of a multi-objective particle swarm optimizer. 2004, pp. 225-237.

[52] Velazquez J. M. O., Coello C. A. C., Arias-Montano A., "Multi-objective compact differential evolution," in 2014 IEEE Symposium on Differential Evolution (SDE), 2014, pp. 1-8.

[53] Wang B.-C., Li H.-X., Zhang Q., Wang Y., "Decomposition-Based Multiobjective Optimization for Constrained Evolutionary Optimization," IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. PP, pp. 1-14, 11/15 2018.

[54] Xue F., Sanderson A. C., Graves R. J., "Multi-objective differential evolution - algorithm, convergence analysis, and applications," in 2005 IEEE Congress on Evolutionary Computation, 2005, vol. 1, pp. 743-750 Vol.1.

[55] Xue Y., Cai X., Neri F., "A multi-objective evolutionary algorithm with interval based initialization and self-adaptive crossover operator for large-scale feature selection in classification," Applied Soft Computing, vol. 127, p. 109420, 2022/09/01/ 2022.

[56] Yue C., Qu B., Liang J., "A Multi-objective Particle Swarm Optimizer Using Ring Topology for Solving Multimodal Multi-objective Problems," IEEE Transactions on Evolutionary Computation, vol. 22, pp. 805-817, 09/19 2017.

[57] Zhou A., Qu B.-Y., Li H., Zhao S.-Z., Suganthan P. N., Zhang Q., "Multiobjective evolutionary algorithms: A survey of the state of the art," Swarm and Evolutionary Computation, vol. 1, no. 1, pp. 32-49, 2011/03/01/ 2011.

[58] Zitzler E., Deb K., Thiele L., "Comparison of Multiobjective Evolutionary Algorithms: Empirical Results," Evolutionary computation, vol. 8, pp. 173-95, 02/01 2000.

Received 10.06.2025, Accepted 23.12.2025