



## OPEN Swift Flight Optimizer: a novel bio-inspired optimization algorithm based on swift bird behavior

Abbas Aqeel Kareem<sup>1</sup>, Ahmed Jabbar Abid<sup>1</sup>, Dalal Abdulmohsin Hammood<sup>1</sup>, Salam J. Yaqoob<sup>2</sup>, Abdalrahman Husein<sup>3</sup> & Viktoriia Berezhnychenko<sup>4</sup>✉

Metaheuristic algorithms inspired by natural phenomena have become indispensable tools for addressing complex, high-dimensional, and multimodal optimization problems. Nevertheless, many existing approaches are constrained by premature convergence, stagnation, and inadequate balance between exploration and exploitation, thereby limiting their effectiveness in solving challenging benchmark problems. This study introduces the Swift Flight Optimizer (SFO), a novel bio-inspired optimization algorithm grounded in the adaptive flight dynamics of swift birds. The novelty of SFO lies in its biologically motivated multi-mode framework, which employs a glide mode for global exploration, a target mode for directed exploitation, and a micro mode for local refinement, augmented with a stagnation-aware reinitialization strategy. This design ensures sustained population diversity, alleviates premature convergence, and enhances adaptability across high-dimensional search landscapes. The efficacy of SFO was rigorously assessed using the IEEE CEC2017 benchmark suite. Experimental findings reveal that SFO attained the best average fitness in 21 of 30 test functions at 10 dimensions and 11 of 30 test functions at 100 dimensions, thereby exhibiting accelerated convergence and a robust exploration–exploitation balance. Comparative evaluations against 13 state-of-the-art optimizers, including PSO, GWO, WOA, and EMBGO, further demonstrate the superior performance of SFO in terms of convergence speed, solution quality, and robustness. Collectively, these results establish SFO as a novel and competitive metaheuristic framework with significant potential for solving large-scale, multimodal, and high-dimensional optimization problems.

**Keywords** Bio-inspired optimization, Swift Flight Optimizer (SFO), Metaheuristic algorithms, Exploration–exploitation balance, Stagnation avoidance, IEEE CEC2017 benchmark, Particle swarm optimization (PSO)

Metaheuristics are a collection of algorithms specifically developed in order to solve hard optimization problems for which classical algorithms are not efficient. These are particularly effective in solving problems involving large, complex, highly nonlinear search spaces where classical optimization algorithms are not efficient due to their extremely high computational needs<sup>1</sup>. Metaheuristics are high-level frameworks for steering heuristic search processes in such a manner so as to enable efficient exploitation as well as exploration of the solution space. Their objective is to obtain high-quality solutions within realistic computational budgets. Most such algorithms are inspired by naturally occurring processes and phenomena, which are robust strategies for dealing with vast and problematic problem domains<sup>2</sup>.

The field of optimization metaheuristics is incredibly diverse, encompassing a wide number of natural-process, cognitive, as well as engineered-system-oriented algorithms<sup>3</sup>. As one of the oldest groups is evolutionary algorithms (EAs)<sup>4</sup>, such as schemes like differential evolution (DE)<sup>5</sup>, and recent contributions in the form of the Logarithmic Mean-Based Optimization (LMO)<sup>6</sup>, which seek at promoting convergence with novel schemes for recombination.

Another major class is swarm intelligence (SI) algorithms<sup>7</sup>, where behavior in social organisms is simulated. Classic examples are particle swarm optimization (PSO)<sup>8</sup> and ant colony optimization (ACO)<sup>9</sup>, while a few newer members are the Puma Optimizer (PO)<sup>10</sup> and Walrus Optimization Algorithm (WOA)<sup>11</sup>. These utilize population-based cooperation and cooperative dynamics for exploration versus exploitation balancing in

<sup>1</sup>Electrical Engineering Technical College, Middle Technical University, Baghdad 10001, Iraq. <sup>2</sup>Training and Energy Researches Office, Ministry of Electricity, 64001, Thi-Qar, Iraq. <sup>3</sup>Department of Data Science and Artificial Intelligence, Hourani Center for Applied Scientific Research, Al-Ahliyya Amman University, Amman, Jordan. <sup>4</sup>Department of Theoretical Electrical Engineering and Diagnostics of Electrical Equipment, Institute of Electrodynamics, National Academy of Sciences of Ukraine, Beresteyskiy, 56, Kyiv 03680, Ukraine. ✉email: vika.berezhnychenko@i.ua

complex search landscapes. A further group of methods is based on physics-inspired principles, such as the gravitational search algorithm (GSA)<sup>12</sup>, which simulates physical forces to guide candidate solutions. Recently, algorithms like the Enzyme Action Optimizer (EAO)<sup>13</sup> have enriched this domain by modeling biochemical reaction dynamics to provide new mechanisms for escaping local optima and accelerating convergence.

Recent progress in metaheuristic research has introduced advanced nature-inspired algorithms that extend beyond these classical frameworks. A notable contribution is the Woodpecker Mating Algorithm (WMA), which models the drumming-based mating behavior of woodpeckers<sup>14</sup>. Enhanced variants, such as the Opposition-based Woodpecker Mating Algorithm (OWMA), integrate opposition-based learning and local memory mechanisms to strengthen the balance between exploration and exploitation, thereby achieving superior performance across diverse optimization problems<sup>15</sup>. Furthermore, parallel and hardware-accelerated implementations have been proposed. For instance, the GPU-based Woodpecker Mating Algorithm (GWMA) employs a CUDA-parallelized framework that substantially reduces computational time reportedly achieving up to an 80-fold speedup compared to the sequential WMA while preserving solution accuracy on the IEEE CEC2017 benchmark suite<sup>16</sup>.

Beyond the WMA family, recent algorithms have further diversified the landscape of bio-inspired optimization. For example, the Wolf–Bird Optimizer (WBO), draws upon interspecies dynamics to model competitive and cooperative behaviors, and has demonstrated strong performance in continuous optimization tasks<sup>17</sup>. In parallel, comprehensive surveys in have systematically catalogued emerging metaheuristics, evaluating their convergence properties, performance trends, and application domains<sup>18</sup>. Likewise, an extensive review<sup>19</sup>, has provided an updated taxonomy of metaheuristics, while also identifying persistent challenges and outlining avenues for future algorithmic development. Collectively, these contributions underscore both the rapid pace of methodological innovation in the field and the necessity of advancing novel, biologically grounded algorithms such as the proposed Swift Flight Optimizer (SFO).

Metaheuristic solutions are found extensively in a range of applications including design optimization in engineering design, financial forecasting, logistics and scheduling in supply chains, bioinformatics, machine learning, and network configuration<sup>3,20</sup>. Because of their flexibility and capability for robust handling of multimodal, nonlinear, high-dimensional optimization problems, their application is especially effective for theoretical research in addition to practical issue solving in real-world scenarios<sup>21</sup>. Even newer applications are bio-inspired algorithms in unmanned aerial vehicle (UAV) and drones in multi-UAV path planning<sup>22</sup>, applications for search-and-rescue mission optimization<sup>23</sup>. Further attesting to their practical range diversification and growing impact in autonomous aerial systems.

However, solutions based on metaheuristics also share some shortcomings and weaknesses. These are a vulnerability towards early convergence, a persisting exploration–exploitation balance, and a reliance on careful parameter tuning. Furthermore, in line with the No-Free-Lunch theorem<sup>24</sup>, there does not exist any optimization algorithm which is equally effective in delivering optimal solutions for all possible problems. As a result, a choice with regard to a suitable metaheuristic is dependent on specific attributes, constraints, and needs of candidate problems. Towards this end, it is interesting to note that works towards designing new optimization schemes which are able to handle more effectively the wide spectrum of challenging problems which are encountered in applications in engineering fields and other applications are still continuous<sup>25</sup>.

Recent advances in bio-inspired metaheuristic optimization have witnessed the emergence of numerous algorithms that address the limitations of classical methods, particularly in high-dimensional and multimodal search spaces. For instance, the Improved Planet Optimization Algorithm hybridized with the Reptile Search Algorithm has been successfully employed in complex segmentation tasks, showing how hybrid strategies can significantly enhance search performance in medical imaging domains<sup>26</sup>. Similarly, the Puma Optimizer (PO) presents a novel predation-inspired strategy, integrating aggressive pursuit mechanisms into machine learning pipelines for superior convergence<sup>27</sup>. Furthering the trend of synergistic design, the Synergistic Swarm Optimization Algorithm introduced a cooperative architecture that dynamically adjusts between exploration and exploitation, emphasizing adaptability in dynamic landscapes<sup>28</sup>. In multi-objective domains, the Guided Epsilon-Dominance Arithmetic Optimization Algorithm has demonstrated impressive efficiency in engineering design by enhancing convergence reliability across Pareto fronts<sup>29</sup>. In the space of single-agent local search models, the Non-Monopolize Search (NO) algorithm proposed a lightweight and effective mechanism, focusing on selective intensification in promising areas while maintaining overall search diversity<sup>30</sup>. From a biological standpoint, novel animal-based behaviors continue to influence metaheuristic innovation—exemplified by the Pufferfish Optimization Algorithm, which models the defensive inflation and threat response mechanisms of pufferfish to guide search dynamics<sup>31</sup>. Hybridization is another major theme in recent literature. For example, the Improved Prairie Dog Optimization combined with the Dwarf Mongoose Optimizer offers a collaborative framework that leverages interspecies behavioral metaphors to escape local optima and improve convergence reliability<sup>32</sup>. Likewise, the Modified Crayfish Optimization Algorithm has been tailored to engineering problems, highlighting the effectiveness of structural adjustments to baseline algorithms<sup>33</sup>. Interestingly, metaphor-less algorithms have also shown great promise. The Far and Near Optimization (FNO) model employs a simple yet effective distance-based update mechanism that forgoes biological metaphor in favor of mathematical clarity and computational efficiency<sup>34</sup>. On the other hand, the Stork Optimization Algorithm applies avian migratory behavior to sustainable production planning, aligning well with the biologically inspired design of SFO<sup>35</sup>. The Teaching–Learning–Based Optimization (TLBO) algorithm remains an influential educational metaphor-based technique, often serving as a baseline for comparative analyses in metaheuristic performance evaluations<sup>36</sup>. Among classical and hybrid swarm intelligence methods, the Particle Swarm Optimization (PSO) algorithm continues to be extensively studied and applied across domains due to its simplicity and convergence properties<sup>37</sup>. Human behavior has also been adopted in optimizer design, as seen in the Sculptor Optimization Algorithm, which mimics iterative refinement and creative decision-making processes akin to

human sculpting<sup>38</sup>. Additionally, the Social Spider Optimization Algorithm integrates cooperative foraging and vibration-based communication strategies, showing strong capabilities in multimodal landscapes<sup>39</sup>. Similarly, the Whale Optimization Algorithm (WOA) continues to receive attention, with recent surveys highlighting its mathematical foundation, variants, and successful application cases<sup>40</sup>. Lastly, the Animal Migration Optimization (AMO) algorithm encapsulates large-scale group movement dynamics, enabling broad exploration and adaptive response to environmental gradients, which directly aligns with the mode-switching and population diversity control mechanisms embedded within SFO<sup>41</sup>.

The primary purpose of the SFO is to overcome persistent limitations of existing metaheuristics, including premature convergence, stagnation in local optima, and an unstable balance between exploration and exploitation. While many algorithms perform adequately on low- and medium-dimensional problems, their effectiveness often declines on high-dimensional, multimodal, or composite benchmarks. SFO addresses these challenges through an adaptive search mechanism inspired by the agile flight dynamics of swift birds. It employs three coordinated modes: glide for global exploration, target for focused exploitation, and micro for localized refinement. Combined with a stagnation-aware reinitialization strategy, this design maintains population diversity, accelerates convergence, and enhances robustness across diverse optimization problems, thereby justifying the development of SFO as a novel and competitive bio-inspired algorithm.

The focus is on developing a novel bio-inspired optimization algorithm referred to herein as SFO based on swift birds (Fig. 1) agile flight behavior and hunting strategies in nature. Tailored for efficient handling of complex high-dimensional optimization problems commonly found in real applications as well as in the realm of engineering, SFO is inspired by swift birds' remarkable agility for effective scanning large environments efficiently, targeting prey precisely, making precise rapid maneuvers in changing environmental conditions<sup>42,43</sup>, so it mimics such behaviors by virtue of adopting three mutually complementary search modes: glide, target, and micro, and dynamically transitioning between modes based on search feedback. Such a biologically inspired design allows for balancing exploration and exploitation sufficiently while maintaining population diversity for effective adaptation towards a broad range of optimization landscapes in a manner akin to swift birds' flight strategies in their intrinsic robustness and flexibility<sup>43</sup>.

Besides its biological inspiration, SFO is organized in a form so that it is capable of dealing with major issues occurring in high-dimensional optimization settings. There are two main computational issues motivating its construction. First, high-dimensional optimization leads oftentimes into the curse of dimensionality so that with large search spaces standard optimizers suffer slow and erratic convergence. Second, many real-world objective functions are multimodal, discontinuous, or contain noisy gradients, further complicating the search process<sup>3</sup>. SFO overcomes these challenges by maintaining a population of candidate solutions that collectively survey multiple regions of the decision space. Through its three adaptive flight modes (glide, target, and micro), SFO integrates stochastic perturbations and dynamic mode switching to escape local minima. This cooperative and biologically inspired mechanism allows SFO to balance intensified exploitation of promising regions with wide exploration of complex landscapes, thereby improving robustness and convergence in high-dimensional problems.

By effectively addressing these challenges, SFO demonstrates strong adaptability and advanced problem-solving capabilities. Its design allows it to handle a wide spectrum of complex optimization tasks, positioning it as a promising approach for engineering and other real-world applications. The principal contributions of this study are as follows:

- The Swift Flight Optimizer (SFO) is introduced as the first bio-inspired metaheuristic algorithm derived from the adaptive flight dynamics of swift birds, establishing a novel paradigm for multi-mode search strategies in optimization.
- A three-mode computational framework is formulated, consisting of a glide mode for global exploration, a target mode for directed exploitation, and a micro mode for localized refinement. This framework is further



**Fig. 1.** Picture of swift bird<sup>44</sup>.

reinforced with a stagnation-aware reinitialization mechanism that preserves population diversity and mitigates premature convergence.

- A comprehensive empirical evaluation is conducted using the IEEE CEC2017 benchmark suite, covering multiple dimensional settings and problem categories, and including comparisons against thirteen state-of-the-art algorithms.
- Experimental analysis demonstrates that SFO consistently achieves accelerated convergence, a balanced exploration–exploitation dynamic, and enhanced robustness, confirming its effectiveness for tackling large-scale, multimodal, and high-dimensional optimization problems.

The rest of this paper is organized as follows: Section “[Related work](#)” reviews related work, discussing existing bio-inspired and nature-inspired optimization algorithms and categorizing them according to their behavioral inspiration. Section “[Swift Flight Optimizer \(SFO\) background theory](#)” introduces the Swift Flight Optimizer (SFO), detailing its biological foundation drawn from swift bird flight dynamics, its mathematical formulation, and its adaptive exploration–exploitation strategies. Section “[Results](#)” describes the experimental setup, including benchmark functions, parameter settings, evaluation metrics, and the set of compared optimization algorithms. Also it presents a comparative performance analysis of SFO against state-of-the-art optimizers across diverse benchmark suites. Finally, Section “[Discussion](#)” concludes the paper by summarizing the main findings and outlining promising directions for future research.

## Related work

Optimization algorithms are fundamental in computational research, offering powerful methods to tackle complex problems across a wide variety of fields. Many of these methods draw inspiration from natural and social phenomena, converting biological, physical, chemical, and behavioral principles into efficient computational strategies<sup>45</sup>. Their richness stems from numerous sources of inspiration, such as animal behaviors, plant dynamics, and physical processes<sup>3</sup>. This diversity has led to algorithms that are not only highly effective but also robust and adaptable to the multi-dimensional and intricate challenges encountered in optimization tasks. Each algorithm emulates certain natural mechanisms, which enable distinctive means for revealing optimal solutions in changing high-dimensional search spaces<sup>46</sup>. By mimicking natural processes, such algorithms conduct exploration, exploitation, and evolutionary transitions and cover the essence of mechanisms in real-world processes<sup>3</sup>. Consequently, Table 1 gives a complete classification scheme for State-Of-The-Art (SOTA) optimization algorithms, which are listed according to their inspirations at a conceptual level and dominant methodological approaches.

Evolutionary-based algorithms are based on Darwinian principles of evolution such that populations of candidate solutions are successively refined in order to achieve superior performance. Genetic Algorithm (GA)<sup>47</sup> and Differential Evolution<sup>9</sup> algorithms are based on mimicking processes like selection, crossover, mutation, and reproduction. These kinds of algorithms maintain a population whose exploration is broad in the search space so that solutions can gradually refine themselves in promising regions with evolutionary pressure. Due to their fine balance between convergence and diversity, a large class of intricate optimization problems are made efficient.

Swarm-based algorithms utilize social animals’ collective intelligence and de-centralized decision-making. Some examples are: Particle Swarm Optimization (PSO)<sup>48</sup> modeling bird flocking behavior and school behavior in fish, whereas pheromone laying is employed for search path direction in Ant Colony Optimization (ACO)<sup>9</sup> for search path guidance. Artificial Bee Colony (ABC)<sup>49</sup>, Whale Optimization Algorithm (WOA)<sup>50</sup>, Ant Lion Optimizer (ALO)<sup>46</sup>, Jellyfish Search (JS)<sup>51</sup>, and Bacteria Phototaxis Optimizer (BPO)<sup>52</sup> further diversify swarm behaviors for optimization purposes. Newer variants such as EMBGO<sup>53</sup>, GPSOM<sup>54</sup>, NSM-AHA<sup>55</sup>, GWOCOA<sup>56</sup>, and QSHO<sup>57</sup> introduce multi-strategy mechanisms, survivorship mechanisms, hybrid mechanisms or quantum augmentation enabling effective exploration in whole regions as well as efficient exploitation in promising solutions. Such inherent collective behavior provides flexibility in exploring large high-dimensional search spaces while keeping computational costs low. A distinctive subgroup within swarm-based algorithms is specifically inspired by cooperative behaviors in mammals. Grey Wolf Optimizer (GWO)<sup>24</sup> mimics hierarchical predation behavior involving chase and encircling strategies whereas Harris Hawks Optimizer<sup>58</sup> mimics strategic pouncing strategies in hawks. Newer approaches such as the Cheetah Optimizer<sup>59</sup> (CO) and Meerkat Optimization Algorithm (MOA)<sup>60</sup> model rapid chasing behavior combined with surveillance behavior as well as synchronized predation behavior. These optimizers demonstrate how modeling mammalian social behavior as well as predation strategies results in optimizers possessing strong global exploration capability in combination with sharp exploitation capability locally.

Physics-based is algorithm inspiration stems from physical laws governing natural phenomena. Simulated Annealing<sup>61</sup> is an imitation of metal annealing for escaping local optima during optimization, while Gravitational Search Algorithm (GSA)<sup>62</sup> is inspired by Newton’s gravitation for attracting agents towards optima. Multi-Verse Optimizer (MVO)<sup>63</sup> uses concepts from cosmology, and Black Hole Algorithm (BH)<sup>64</sup> is an imitation of black hole gravitational force. These utilize familiar physics analogies for exploitation versus exploration balancing in complex landscapes.

Chemical and biochemical algorithms project reaction dynamism onto search approaches. Atom Search Optimization (ASO)<sup>65</sup> mimics atom movement and bonding forces, Chemical Reaction Optimization (CRO)<sup>66</sup> reflects chemical reaction dynamism, while Nuclear Reaction Optimization (NRO)<sup>67</sup> mimics nuclear level processes. These methods tend to utilize energy states and reaction paths for diversification objectives while gradually enhancing solutions in a steady state form, which is suitable for hard engineering and design problems. Biology-inspired optimizers are based on microorganisms or biological processes apart from typical swarm behavior. Enzyme Action Optimizer (EAO)<sup>13</sup> abstracts enzymatic catalytic action in a manner such that candidate solutions are adaptively modified. Pufferfish Optimization Algorithm (POA)<sup>68</sup> employs pufferfish

Category	Algorithm	Main idea	Year	Exploration capability	Exploitation capability
Evolutionary	Genetic Algorithm (GA)	Natural selection and genetics	1975	High	High
	Differential Evolution (DE)	Differential vectors and evolution principles	1997	High	High
	Genetic Programming (GP)	Evolutionary computation via program structures	1998	Medium	Medium
Swarm	Particle Swarm Optimization (PSO)	Bird flocking and fish schooling	1995	High	High
	Ant Colony Optimization (ACO)	Ants' pheromone trails and foraging	2006	High	Medium
	Artificial Bee Colony (ABC)	Bee foraging and colony behavior	2005	High	Medium
	Whale Optimization Algorithm (WOA)	Humpback whale bubble-net hunting	2016	Medium	High
	Ant Lion Optimizer (ALO)	Antlion hunting strategies	2015	High	High
	Jellyfish Search (JS)	Jellyfish swarm and ocean current dynamics	2020	High	Medium
	Bacteria Phototaxis Optimizer (BPO)	Phototaxis behavior of bacteria	2023	High	High
	GPSOM–Group-based PSO with Multiple Strategies	Group-based particle swarm with multi-strategy	2025	Very High	Very High
	NSM-AHA–Natural Survivor Method-based Artificial Hummingbird Algorithm	Hummingbird foraging with survivor selection	2024	Very High	High
	EMBGO–Enhanced Marine Battle and Gathering Optimizer	Marine predator–prey behavior and battle interactions in the ocean	2025	High	Medium
	GWOCSA–Hybrid Grey Wolf & Crow Search Algorithm	Grey wolf hunting and crow search hybrid	2019	High	High
	QSHO–Quantum Spotted Hyena Optimizer	Spotted hyena hunting with quantum adaptation	2025	Very High	High
Mammalian–Behavior (Swarm sub-type)	Grey Wolf Optimizer (GWO)	Grey wolf pack hunting hierarchy	2014	Medium	High
	Harris Hawks Optimizer (HHO)	Harris hawk cooperative hunting	2019	High	High
	Cheetah Optimizer (CO)	Cheetah chasing and predation behavior	2022	High	Medium
	Meerkat Optimization Algorithm (MOA)	Meerkat cooperative surveillance and hunting	2023	Medium	High
Physics	Simulated Annealing (SA)	Metallurgical annealing process	1983	Medium	Medium
	Gravitational Search Algorithm (GSA)	Newtonian gravity and motion	2009	Medium	High
	Multi-Verse Optimizer (MVO)	Multiverse theory and white/black hole concepts	2016	High	High
	Black Hole Algorithm (BH)	Black hole attraction phenomenon	2013	Low	High
	Elastic Deformation Optimization Algorithm (EDOA)	Elastic deformation mechanics	2022	High	High
Chemical/Biochemical	Atom Search Optimization (ASO)	Atomic motion and bonding forces	2019	High	High
	Chemical Reaction Optimization (CRO)	Chemical reaction dynamics	2009	High	Medium
	Nuclear Reaction Optimization (NRO)	Nuclear reaction processes	2019	High	High
Biology	Enzyme Action Optimizer (EAO)	Enzymatic catalytic action in biology	2025	High	High
	Pufferfish Optimization Algorithm (POA)	Pufferfish defense against predators	2024	High	High
	Artificial Protozoa Optimizer (APO)	Protozoa foraging and reproduction	2024	High	High
System	Water Cycle Algorithm (WCA)	Water cycle: evaporation, precipitation, flow	2012	Medium	Medium
	Snow Ablation Optimizer (SAO)	Snow ablation and melting dynamics	2023	Medium	High
Math	Arithmetic Optimization Algorithm (AOA)	Arithmetic operations for position update	2021	High	High
	Chaos Game Optimization (CGO)	Chaos game theory and fractals	2021	High	Medium
	Sine Cosine Algorithm (SCA)	Sine and cosine mathematical oscillations	2016	High	Medium
Human	Chef-Based Optimization Algorithm (CBOA)	Chefs' process of improving a recipe	2022	Medium	Medium
	Cultural Algorithms (CA)	Cultural knowledge evolution in societies	1994	Medium	Medium

**Table 1.** Classification of SOTA optimization algorithms.

defense behaviors as an inspiration, while Artificial Protozoa Optimizer (APO)<sup>69</sup> mimics a foraging and reproduction cycle in protozoa. These algorithms utilize biological diversity and adaptability for robust search with effective convergence.

System-based approaches model interactions within ecological or natural systems. Water Cycle Algorithm (WCA)<sup>70</sup> simulates evaporation, precipitation, and runoff to explore and exploit search spaces, while Snow Ablation Optimizer (SAO)<sup>71</sup> uses melting and accumulation dynamics to guide solution updates. By abstracting ecological processes, these methods can efficiently handle multi-modal and dynamic optimization challenges. Math-based algorithms are derived from mathematical operators and functions. Arithmetic Optimization Algorithm (AOA)<sup>72</sup> employs arithmetic operators for guiding movement, Chaos Game Optimization (CGO)<sup>73</sup> uses chaotic maps along with fractals for diversity purposes, while Sine Cosine Algorithm (SCA)<sup>74</sup> uses oscillatory patterns of sines and cosines for balancing exploitation versus exploration. Due to their mathematical origins, it is simple for users to implement and fine-tune them for a wide range of optimization problems.

Human-oriented algorithms include processes based on human behavior and knowledge passing. Chef-Based Optimization Algorithm (CBOA)<sup>75</sup> reflects successive optimization by chefs, while Cultural Algorithms (CA)<sup>76</sup> mimic cultural knowledge evolution over a few generations. Both are iteration-based learning and adaptation-oriented approaches which are easy to use for a large number of applications and versatile. Although metaheuristic and evolutionary algorithms are significantly variform, their severe limitations include premature

convergence, low adaptability in changing environments, and strong needs for extensive parameter tuning<sup>3</sup>. Classical evolutionary algorithms always suffer exponentially growing computational costs which render their efficiency low when dealing with very large or highly complicated search spaces. Moreover, strongly gradient-oriented approaches are poor on highly nonlinear or discontinuous problems such that it is essential for flexible, gradient-free optimization strategies to be constructed<sup>3</sup>.

Recent studies have introduced advanced multi-strategy metaheuristics to improve the balance between exploration and exploitation and to enhance convergence reliability. The Artificial Lemming Algorithm (ALA) combines four behavioral patterns: migration, burrowing, foraging, and predator evasion, all regulated by an adaptive energy factor that sustains population diversity and solution accuracy<sup>77</sup>. The Multi-Strategy Enhanced Arithmetic Optimization Algorithm (MSEAOA) extends the original AOA through mechanisms such as good point set initialization, cooperative learning strategies, and perturbation-based refinements, resulting in faster and more precise convergence<sup>78</sup>. Similarly, the Multi-Strategy Cooperative Augmented Crayfish Optimization Algorithm (MCOA) incorporates adaptive cooperation strategies and dynamic parameter control, yielding stable performance on CEC benchmarks and demonstrating practical effectiveness in engineering tasks, including unmanned aerial vehicle path planning<sup>79</sup>.

These weaknesses notably affect the effectiveness of metaheuristic algorithms when finding solutions for high-dimensional or strongly nonlinear optimization problems. Premature convergence is a key deficiency when an algorithm fails to sustain significant diversity in its population, particularly in multimodal search landscapes<sup>80</sup>. As a result, the search is trapped in local optima while not obtaining actual global solutions<sup>81</sup>. Slow adaptability in dynamic environments is another key deficiency since most algorithms apply constant exploration and exploitation behaviors<sup>82</sup>. Such inadaptability constrains their performance in handling changing objectives or constraints and thus renders them less suitable for realistic real-world problems<sup>83</sup>.

Metaheuristic algorithms are also subject to several major drawbacks which hinder their universal use. Most are greatly reliant upon parameter tuning such as setting mutation rates, crossover probabilities, and weight multipliers which is slow and restricts generalizability<sup>83</sup>. Scalability is also restricted by the “curse of dimensionality” because computational expense grows exponentially with model size so large-scale optimization is inefficient<sup>80</sup>. Moreover, in applications involving heavily nonlinear, discontinuous, or noisy objective functions for which gradient-based approaches are unsupportable, there is a need for strong gradient-free approaches<sup>82</sup>. They are also less robust in multimodal landscapes such that there is early convergence upon local optima in favor over exploration of multiple regions simultaneously<sup>82</sup>. Many are also reliant upon problem-oriented adaptations such that there is a restriction in their capacity for adaptation between problems in a set of different optimizations task<sup>80</sup>.

Another essential advantage of the Swift Flight Optimizer (SFO) is an adaptive mode-switching within the algorithm. Such a mode-switching between modes of gliding, targeting, along with micro-adjustment is performed on a dynamic level in order to ensure stable convergence behavior compared with a majority of state-of-the-art optimizers. Such characteristics make it best suited for hard, high-dimensional, as well as hard multimodal optimization problems.

By taking inspiration from flight dynamics as well as adaptive behavior observed in swift birds, SFO compensates for significant weaknesses observed in conventional metaheuristics and offers a robust nature-inspired method for handling a large number of challenging optimization problems. We now give the mathematical formulation for SFO and its applications for design examples in engineering.

## Swift Flight Optimizer (SFO) background theory Originality and distinct features of SFO

The Swift Flight Optimizer (SFO) introduces a set of conceptual and mathematical innovations that clearly differentiate it from existing swarm intelligence algorithms. To the best of our knowledge, it represents the first metaheuristic inspired by the flight dynamics of swift birds, whose distinctive behaviors are modeled through three complementary modes: glide, target, and micro. Within this structure, the glide mode facilitates large-scale exploration across the search space, the target mode enables directed exploitation toward promising solutions, and the micro mode provides localized refinements around elite candidates. This tri-modal framework reflects the natural adaptability of swifts, allowing the algorithm to sustain balance between global and local search processes.

Unlike classical swarm optimizers such as PSO, ACO, and ABC, which rely on a single update equation, or more recent methods such as BPO, GPSOM, EMBGO, and QSHO, which introduce partial multi-strategy enhancements, SFO integrates a multi-mode adaptive structure within a unified framework. This structure is reinforced by a stagnation-aware reinitialization mechanism that restores population diversity when premature convergence occurs, thereby maintaining balance between exploration and exploitation even in high-dimensional search spaces.

Table 2 summarizes the conceptual and methodological distinctions between SFO and representative swarm-based algorithms. As shown in Table 2, classical swarm algorithms such as PSO, ACO, and ABC are based on single updating equations and exhibit weak stagnation recovery mechanisms. Mid-generation algorithms such as GWO and WOA improve search adaptability through hierarchical and encircling behaviors but remain vulnerable to premature convergence. Recent developments, including GPSOM, EMBGO, and QSHO, have introduced group-based multi-strategy updates, dual-stage global-local search, and quantum-enhanced dynamics, respectively. While these methods demonstrate stronger adaptability, they do not employ explicit stagnation recovery or full adaptive mode switching. In contrast, the proposed SFO integrates a three-mode adaptive structure inspired by swift flight dynamics, coupled with a stagnation-aware reinitialization mechanism. These features establish SFO as both conceptually novel and formulaically distinct within the swarm intelligence domain.

Algorithm	Inspiration	Core mechanism	Exploration strategy	Exploitation strategy	Distinctive features
PSO	Bird flocking behavior	Velocity–position update guided by personal and global bests	Random coefficients and inertia weight promote global search	Cognitive and social terms refine convergence	Simple structure, widely used but prone to premature convergence
ACO	Ant foraging and pheromone trails	Probabilistic path construction based on pheromone intensity	Exploration via pheromone evaporation	Exploitation through reinforcement of best paths	Strong combinatorial optimization, weaker in continuous domains
ABC	Foraging of honey bees	Employed, onlooker, and scout phases	Scouts introduce random search	Onlooker bees exploit best nectar sources	Efficient balance, but slower convergence in high dimensions
GWO	Grey wolf hunting hierarchy	Encircling and attacking prey guided by alpha, beta, delta wolves	Diversification through coefficient vectors	Intensification near best wolves	Competitive performance with simplicity
WOA	Bubble-net hunting of whales	Spiral updating and encircling mechanism	Shrinking encircling and random movement	Exploitation via spiral bubble-net attack	Strong intensification, weaker diversity maintenance
GPSOM	Group-based swarm strategies	Division into three sub-swarms (exploration, exploitation, balance)	Sub-swarm 1 enhances diversity via dynamic operators	Sub-swarm 2 and 3 refine search with adaptive sine–cosine and positional adjustment	Proven effectiveness on CEC benchmarks and engineering tasks
EMBGO	Multiplayer battle royale dynamics	Integrated movement–battle phases with greedy selection	Differential mutation and Lévy flight enhance diversity	Battle phase exploits stronger opponents and safe zone convergence	Robust across CEC2017–2022, also applied to adversarial neural architecture search
QSHO	Quantum-enhanced spotted hyena hunting	Incorporates quantum computing into SHO model	Quantum states expand search space and prevent trapping	Encircling and attacking prey with enhanced convergence	Outperforms classical SHO and variants on CEC2013/2017 and real-world engineering tasks
SFO (Proposed)	Swift birds' adaptive flight	Multi-mode structure: glide, target, micro + stagnation reset	Glide mode ensures broad coverage	Target and micro modes refine local optima	First swift-inspired optimizer; integrates explicit stagnation-aware reinitialization

**Table 2.** Distinctive features of SFO compared with swarm-based algorithms.

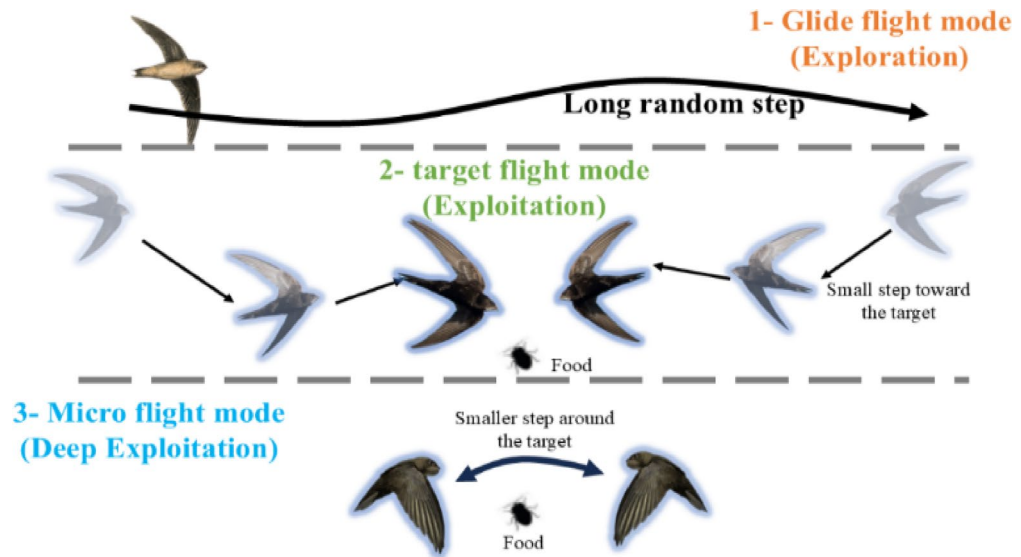
### Biological inspiration

The proposed Swift Flight Optimizer (SFO) gets its basic inspiration from the common swift's (*Apus apus*) adaptive flight behavior which is famous for its exceptional aerial endurance and mobility<sup>84</sup>. Unlike most bird species which land for rest or foraging while in continuous flights, swift birds remain in flight for months at a stretch performing gliding, direction-oriented foraging, and stabilizing perturbation in lieu of landing<sup>84</sup>. Such continuous flight ability is a fine compromise between distant travel and real-time control such that swift birds are a prime biological example for optimization frameworks which need exploration, exploitation, and stability within complex search landscapes. When performing high-altitude migration or foraging for food supplies, swift birds primarily utilize energy-optimal gliding for covering large tracts with a low metabolic exertion. Such a behavior is made possible by narrow crescent-wing form and elongated streamlined body such that they are in a position for exploiting upward currents inherent within thermals as well as atmospheric updrafts<sup>43</sup>. When conceptualized in optimization terms, such biological behavior is mimicked in glide mode in SFO such that agents perform large inertial movement for worldwide exploration in a similar manner as the bird's low-cost travel for large distances in search for favorable environmental signals.

When environmental stimuli like clusters of insects are perceived, swifts change passive gliding into active high-speed chasing. Such foraging behavior is marked by rapid direction reversals and accelerating towards dynamic preys based on input from sensors and fine-motion control<sup>85</sup>. SFO inherits such a biological method in its Target Mode in which agents perform abrupt transitions in their courses towards best-known solutions thus far for enhancing convergence accuracy in auspicious subregions. Such a technique enables such an optimizer to mimic the bird's capability for promptly exploiting ephemeral opportunities within its milieu. Under turbulent or unstable airflow conditions, swifts display remarkable control by performing fine-scales wing as well as tail maneuvers. Such micro-movements settle their position while in flight and prevent aerodynamic instability<sup>86</sup>. Drawing such a technique as an inspiration, there is an implementation for a Micro Mode in SFO which performs localized fine-tuning in the neighborhood around elite solutions so as to permit in-depth exploitation in optima-laden realms with tight optima or softly graded fitness slopes. Such an adaptable control technique ensures no oscillation or stagnation near optima. Altogether, swift flight mechanics (long distance gliding, directed directional movement towards objectives, as well as fine-grained movement locally in its immediate milieu during flight maneuvers) provide a biologically inspired foundation for developing a multi-modal optimization method. Such behaviors are simulated computationally within our proposed Swift Flight Optimizer (SFO) as three mutually complementary search modes: Glide, Target, as well as Micro. Each such mode serves a specific purpose in balancing exploration and exploitation throughout an optimization campaign. As illustrated in Fig. 2, there is a glide mode which accommodates wide exploration in keeping with extended randomized steps; a target mode which identifies search towards substantiated regions; a micro mode which conducts fine-motion refinements in the vicinities around optimum solutions. Such a biologically inspired technique enables such an SFO for effectively solving high-dimensional, multimodal, as well as dynamic scenarios in optimization in minimal parameter fine-tuning with increased adaptability.

### Mathematical model

The Swift Flight Optimizer (SFO) is a bio-inspired metaheuristic algorithm that mimics the dynamic flight behaviors of swift birds across three distinct modes: glide, target, and micro. Each mode is modeled by a separate velocity update strategy, reflecting the birds' adaptive navigation and search behaviors in their natural environment. Swifts often glide across long distances with minimal energy expenditure, exploring vast regions



**Fig. 2.** Illustration of the three flight modes in the proposed Swift Flight Optimizer (SFO).

of the sky. This behavior is modeled by stochastic movement influenced by inertial momentum, Gaussian noise, and Lévy flight-based exploration.

The Eq. 1 used to update the velocity of the given agent  $i$  in *Glide Mode*:

$$v_i^{t+1} = \omega \cdot v_i^t + \alpha \cdot N(0, I) + \beta \cdot L \tag{1a}$$

$$x_i^{t+1} = x_i^t + v_i^{t+1} \tag{1b}$$

where  $v_i^{t+1}$  is the current velocity,  $x_i^t$  is the current position,  $\omega$  is the inertia weight,  $\alpha$  and  $\beta$  are scaling factors for Gaussian and Lévy components,  $N(0, I)$  is a vector of standard normal values,  $L$  is the Lévy step vector computed using Mantegna’s algorithm. Equation 2 used to define Lévy step components  $L_j$

$$L_j = \frac{u_j}{|v_j|^{1/\beta}} \tag{2}$$

where  $u_j \sim N(0, \sigma_u^2)$  is a standard normal variable,  $v_j \sim N(0, 1)$  and  $\beta \in (0, 2]$  is a stability parameter (commonly set to 1.5).

Equation 3 use to compute the scale parameter  $\sigma_u$ :

$$\sigma_u = \left[ \frac{\Gamma(1 + \beta) \cdot \sin\left(\frac{\pi\beta}{2}\right)}{\Gamma\left(\frac{1+\beta}{2}\right) \cdot \beta \cdot 2^{(\beta-1)/2}} \right]^{1/\beta} \tag{3}$$

where  $\Gamma(\cdot)$  is the gamma function. This mechanism mimics the swift bird’s erratic long-glide movements, helping the algorithm escape local optima and maintain search diversity across complex landscapes. When a swift identifies a food source or landing spot, it rapidly transitions into targeted flight. This behavior is translated into the optimizer by guiding agents toward the global best solution found so far, while preserving swarm diversity.

The velocity in *Target Mode* is updated as in (4):

$$v_i^{t+1} = \omega \cdot v_i^t + \psi_1 r_1 (p_{best} - x_i^t) + \psi_2 r_2 (g_{best} - x_i^t) \tag{4a}$$

$$x_i^{t+1} = x_i^t + v_i^{t+1} \tag{4b}$$

where  $p_{best}$  is the personal best position of agent  $i$ ,  $g_{best}$  is the global best position among all agents,  $\psi_1$  and  $\psi_2$  are acceleration coefficients and  $r_1, r_2$  are random scalars  $\in [0, 1]$ . This model captures the goal-oriented descent of swifts toward known high-quality regions in the decision space. In final approaches, swifts exhibit fine-tuned, micro-level wing adjustments for high precision control. This behavior is implemented in SFO as a local exploitation strategy around the best-known solution. In this mode, the position is directly perturbed using a Gaussian and adaptive step as in (5).

$$v_i^{t+1} = \omega \cdot v_i^t + \psi_2 r_2 (g_{best} - x_i^t) + \varepsilon \tag{5a}$$

$$x_i^{t+1} = x_i^t + v_i^{t+1} \quad (5b)$$

where  $l$  is a small step size.

This stage ensures local refinement near high-quality optima, enabling convergence accuracy, akin to the swift's fine body control when catching prey mid-air or landing on narrow surfaces.

### Mode switching strategy and stagnation avoidance

A critical strength of the Swift Flight Optimizer (SFO) lies in its adaptive mode-switching mechanism and stagnation avoidance strategy, which together ensure effective navigation of complex, multimodal search spaces. Inspired by the flexible and responsive flight behavior of swift birds, the algorithm dynamically transitions between distinct behavioral phases (Glide Mode, Target Mode, and Micro Mode) to balance exploration and exploitation based on real-time feedback from the fitness landscape. Transitions between modes are governed by adaptive rules based on recent improvement history:

- *Glide Mode* is activated when exploration is required. It generates large, random movements using Lévy flight and stochastic noise. When no improvement (IMPR) occurs in Glide mode the glide fail counter (GFC) is incremented by one. If GFC is more than four and the Glide–Target–Cycle counter (GTC) remains below a predefined threshold (typically 30) the algorithm switches to *Target Mode*.
- *Target Mode* focuses on exploitation by guiding agents toward the best global solution with smaller, controlled steps. When no IMPR occurs in target mode the target fail counter (TFC) is increased by one. If TFC is more than ten, and the GTC counter below a predefined threshold the algorithm switches to *Glide Mode* to encourage broader exploration.
- *Micro Mode* a fine-tuning technique for exploitation at a deeper level locally. It is invoked when Target Mode and Glide Mode were repeatedly cycled more than 30 times without satisfactory improvement, and no stagnation is detected. When stagnation counter (SC) is reached up to more than 20 iterations with reset count (RstC) less than 5, population is re-initialized so diversity restoration occurs.

SFO also keeps count of number of GTC in order to monitor behavioral change. When such count exceeds a pre-specified threshold (e.g., 30), when further improvement is not taking place, Micro Mode is activated for fine-grained exploitation. It guarantees phasic responsiveness with a layering effect akin in flexibility to observed real swift bird foraging behavior. To avert convergence too prematurely, SFO adopts a stagnation counter counting global improvement over entire swarm. When improvement in best fitness is not observed for more than 20 iterations, while its reset count is below 5, population is rejuvenated. Its role is restoring diversity so it can explore once again in search space while also acting as a controlled restart. Its stagnation-aware re-initialization is for not only restoring diversity but also for allowing exploration in search space once again. This stagnation-aware reinitialization serves two purposes:

- Escape from local optima
- Maintenance of population diversity, both essential for robust performance in high-dimensional and dynamic optimization search space.

By combining biologically inspired flight behaviors with an adaptive control logic, SFO avoids common pitfalls in metaheuristic search, such as premature convergence and unproductive oscillations. The full flow of SFO optimizer including mode switching and stagnation handling mechanism is illustrated in Fig. 3.

#### *Avoidance of local optima*

One of the principal challenges in population-based metaheuristics is premature convergence to local optima. SFO addresses this limitation through several mechanisms. First, its three-mode design (glide, target, and micro-refinement) enables adaptive transition from global exploration to fine-grained exploitation, reducing the likelihood of early stagnation. Second, the glide phase incorporates stochastic perturbations, including Lévy-distributed steps, which introduce long-range moves that expand the search frontier and help escape local basins. Third, a stagnation-aware reinitialization strategy actively restores diversity when no improvement is observed over several iterations, thereby preventing population collapse. Finally, adaptive switching thresholds ensure that the transition between modes is not purely deterministic but guided by fitness improvement signals, which enhances resilience against entrapment in narrow valleys. Collectively, these mechanisms contribute to SFO's robustness in navigating multimodal and high-dimensional landscapes, ensuring a sustained balance between exploration and exploitation.

#### *Exploration–exploitation balance*

Maintaining an appropriate balance between exploration and exploitation is a defining requirement of high-performing metaheuristics. In SFO, this balance is embedded in its design. The glide phase promotes exploration through stochastic updates and Lévy perturbations, which broaden the search frontier and enhance coverage of the solution space. The target phase intensifies exploitation by steering agents toward elite solutions and consolidating improvements around promising basins. The micro-refinement phase further sharpens exploitation by applying local adjustments to refine solution accuracy. Adaptive mode switching regulates transitions among these phases, ensuring that exploration does not diminish too early and exploitation does not dominate prematurely. In addition, stagnation-aware reseeding reinstates exploration when diversity loss is detected. The effectiveness of this architecture is reflected in the convergence profiles, which exhibit rapid early descent followed by stable refinement, as well as in oscillatory patterns of exploration–exploitation dynamics that

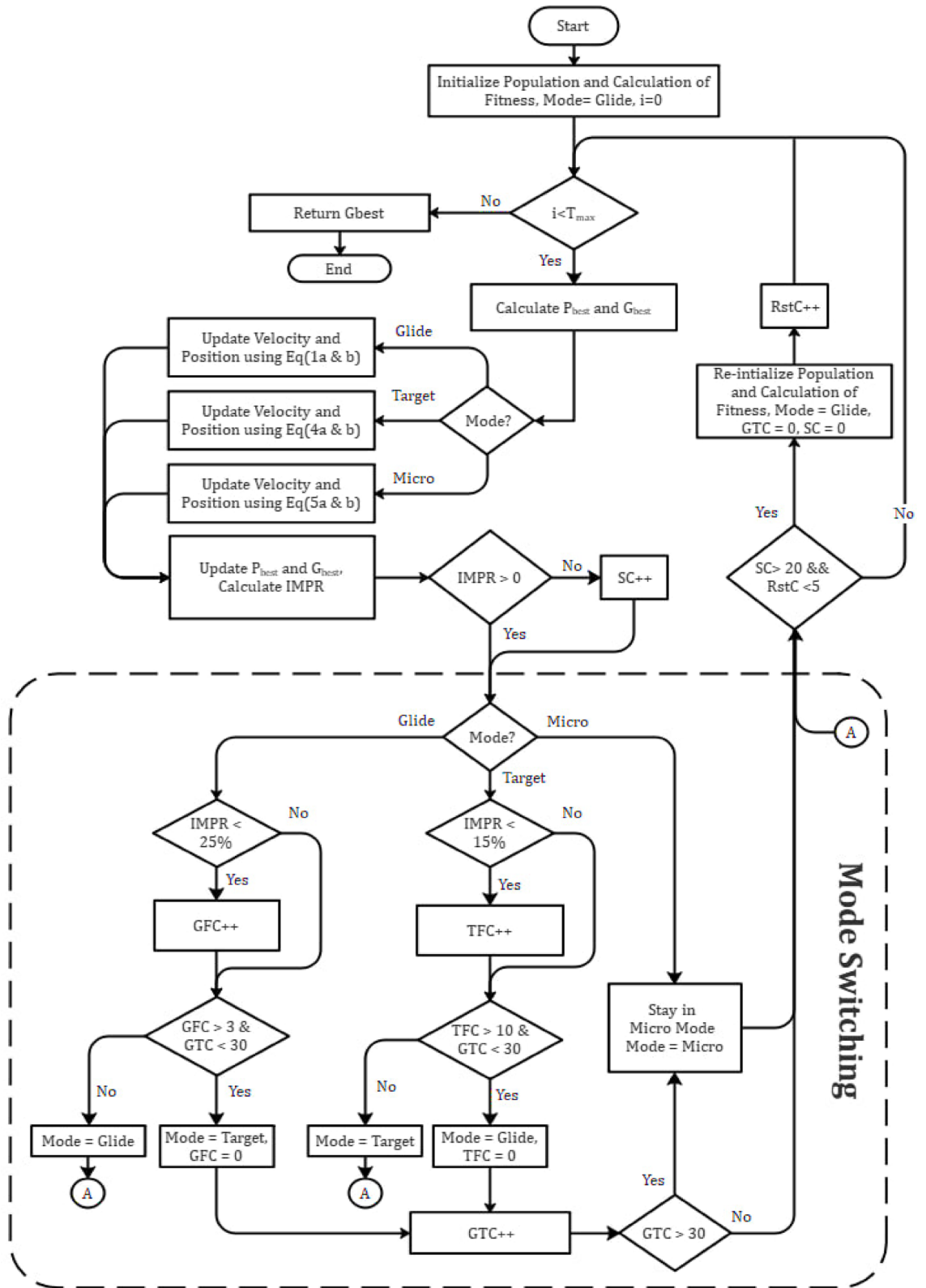


Fig. 3. SFO Flowchart with stagnation handling and mode-switching mechanism.

signify successful reactivation of diversity when stagnation occurs. These features collectively sustain a dynamic equilibrium between global search and local intensification across diverse problem classes and dimensional settings.

*SFO algorithm analysis*

Algorithm 1 illustrates the core workflow of the Swift Flight Optimizer (SFO).

- *Step 1* The algorithm begins with the initialization phase, where a population of particles (candidate solutions) is randomly generated, and initial velocities are assigned.
- *Step 2* The fitness of each particle is evaluated to determine both the individual best positions (pbest) and the global best solution (gbest) within the population.
- *Step 3* The main optimization loop iterates for a fixed number of iterations ( $T_{\text{Max}}$ ).
- *Step 4* In each iteration, every particle undergoes a behavioral update governed by the adaptive flight modes of the swift bird.
- *Step 5* The algorithm dynamically switches among three flight behaviors Glide, Target, and Micro Flight based on specific rules that detect stagnation or improvement in the particle's search trajectory.
- *Step 6* Depending on the current mode, the velocity and position of the particle are updated using the corresponding mathematical model: long exploratory movements in Glide, goal-oriented updates in Target, or fine-grained refinements in Micro mode.
- *Step 7* After movement, the fitness is re-evaluated, and the personal and global bests are updated if improved solutions are found.
- *Step 8* Repeat this process until all particles are updated, and in every iteration, an optimal solution is saved.
- *Step 9* After a set maximum number of iterations is reached, gbest (the best global solution) is returned as the final result of the optimization routine.

The pseudocode embodies the adaptiveness in SFO, representing multi-mode coordination with minimal adjustment in parameters for robustness in varying search landscapes.

---

**INITIALIZATION:**

```

1  | Initialize positions and velocities
2  | Evaluate fitness to set pbest, gbest
3  | For t = 1 to MaxIter:
4  |   | Foreach particle in particles Do:
5  |   |   | Update mode (Glide / Target / Deep Micro Adjust) based on rules
6  |   |   | Update velocity and position using mode formula
7  |   |   | Update pbest and gbest
8  |   |   | End
9  |   |   | Record gbest
11 |   | End
12 | Return gbest

```

---

**Algorithm 1.** Swift Flight Optimizer.*SFO computational complexity*

The computational complexity of the Swift Flight Optimizer (SFO) can be assessed by examining its key operational stages: population initialization, fitness evaluation, mode-dependent updates, and iterative tracking of the best solutions. Each stage contributes to the overall computational load as follows.

The initialization phase generates a population of  $n$  particles in a  $d$  dimensional search space. Each particle is initialized with random position and velocity vectors, resulting in a time complexity of

$O(nd)$ . In every iteration, the fitness function is evaluated for each particle to update personal and global bests. Assuming the fitness function requires  $O(F)$  time per particle, the fitness evaluation for the entire population per iteration has a complexity of  $O(nF)$ . Let  $T$  denote the maximum number of iterations. The total fitness evaluation complexity over all iterations becomes  $O(TnF)$ . Each particle updates its velocity and position based on the current behavioral mode (Glide, Target, or Micro) with different stochastic and adaptive mathematical models. These updates are computed per particle per iteration and are linear in the dimension  $d$ , yielding a complexity of  $O(d)$ . Thus, the per-iteration position and velocity update cost for all  $n$  particles is  $O(nd)$ . Over  $T$  iterations, the cumulative cost of all mode-based updates is  $O(Tnd)$ . Combining the initialization, evaluation, and update stages, the total computational complexity (TCC) of SFO is expressed as in (6):

$$\text{TCC}_{\text{SFO}} = O(nd) + O(TnF) + O(Tnd) = O(TnF) + O(Tnd) \quad (6)$$

$\text{TCC}_{\text{SFO}}$  indicates that the runtime of SFO scales linearly with the number of particles  $n$ , the problem dimensionality  $d$ , and the number of iterations  $T$ , making it suitable for scalable high-dimensional optimization problems, especially where fitness evaluation dominates computational cost.

## Results

An overview of the experimental setup is given in this section, together with details on the benchmark functions, comparison algorithms, settings of parameter, and evaluation metrics.

### Benchmark description

The IEEE CEC 2017 benchmark suite is a widely recognized set of 30 test functions curated by the IEEE Congress on Evolutionary Computation to rigorously evaluate the performance of optimization algorithms. These functions are categorized into four main types: unimodal(Uni.), multimodal(Multi.), hybrid(Hybr.), and composition (Comp.) functions, each presenting unique challenges to test various aspects of algorithmic behavior, as shown in Table 3. Unimodal functions are designed to assess the exploitation capabilities of algorithms in smooth landscapes, while multimodal functions test exploration abilities by presenting numerous local optima. Hybrid functions combine characteristics of both unimodal and multimodal functions, challenging algorithms to balance exploration and exploitation. Composition functions, the most complex category, are formed by blending multiple basic functions to simulate highly irregular and rugged search spaces, reflecting real-world problem complexity. Together, this benchmark suite provides a comprehensive and systematic framework for benchmarking an algorithm's efficiency, adaptability, and robustness in solving high-dimensional and diverse optimization problems.

### Computational environment

All simulations and comparative experiments were conducted on a workstation equipped with an Intel Core i7 processor (3.2 GHz, 16 cores), 16 GB RAM, and an NVIDIA GeForce RTX 3070 GPU with 16 GB VRAM. The operating system environment was Ubuntu Linux 20.04 LTS (64-bit). The algorithms were implemented

No	Func	Type	$f_{opt}$
$F_1$	Shifted and Rotated Bent Cigar function	Uni	100
$F_2$	Shifted and Rotated Zakharov function		200
$F_3$	Shifted and Rotated Rosenbrock's function	Multi	300
$F_4$	Shifted and Rotated Rastrigin's function		400
$F_5$	Shifted and Rotated Expanded Scaffer's F6 function		500
$F_6$	Shifted and Rotated Lunacek Bi_Rastrigin function		600
$F_7$	Shifted and Rotated Non-Continuous Rastrigin's function		700
$F_8$	Shifted and Rotated Levy function		800
$F_9$	Shifted and Rotated Schwefel's function		900
$F_{10}$	Hybrid function 1 (N=3)	Hybr	1000
$F_{11}$	Hybrid function 2 (N=3)		1100
$F_{12}$	Hybrid function 3 (N=3)		1200
$F_{13}$	Hybrid function 4 (N=4)		1300
$F_{14}$	Hybrid function 5 (N=4)		1400
$F_{15}$	Hybrid function 6 (N=4)		1500
$F_{16}$	Hybrid function 6 (N=5)		1600
$F_{17}$	Hybrid function 6 (N=5)		1700
$F_{18}$	Hybrid function 6 (N=5)		1800
$F_{19}$	Hybrid function 6 (N=6)		1900
$F_{20}$	Composition function 1 (N=3)	Comp	2000
$F_{21}$	Composition function 2 (N=3)		2100
$F_{22}$	Composition function 3 (N=4)		2200
$F_{23}$	Composition function 4 (N=4)		2300
$F_{24}$	Composition function 5 (N=5)		2400
$F_{25}$	Composition function 6 (N=5)		2500
$F_{26}$	Composition function 7 (N=6)		2600
$F_{27}$	Composition function 8 (N=6)		2700
$F_{28}$	Composition function 9 (N=3)		2800
$F_{29}$	Composition function 10 (N=3)		2900
$F_{30}$	Composition function 11 (N=3)		3000
Search Range = $[-100, 100]^d$			

**Table 3.** CEC2017 benchmarks functions summary<sup>87</sup>.

in Python 3.10 using scientific computing libraries including NumPy 1.24, SciPy 1.10, and Matplotlib 3.7 for visualization. Random number generation was controlled through fixed seeds to ensure reproducibility. Benchmark functions from the IEEE CEC2017 test suite were evaluated under identical conditions across all algorithms.

#### Compared optimizers

Table 4 summarizes a variety of 13 contemporary, widely mentioned optimizations techniques with which the performance of SFO was assessed and contrasted.

#### Evaluation metrics

The mean and standard deviation were used to statistically assess each algorithm's effectiveness. These metrics were selected in order to shed light on the optimization's outcomes' consistency throughout several runs as well as their average performance.

#### Parameters settings

For experimentation with the proposed Swift Flight Optimizer (SFO) on the IEEE CEC2017 benchmark functions, population size ( $n$ ) was chosen as a constant 50 agents, while maximum number iterations were 1000 in order to permit for the fine details and complexity of the benchmark landscape. To examine further the scalability and robustness in high-dimensional regimes of the algorithm, further experimentation was again conducted for 100-dimensional optimization problems. Here, in this enlarged setting, population size was chosen as a constant 100 agents, while maximum number iterations were made 10,000. Parameter values for algorithm were chosen as: inertia weight  $w=0.5$ , cognitive coefficient  $c_1=1.5$  and social coefficient  $c_2=1.5$ . These particular settings were chosen intentionally in order to permit for adequately strong exploration ability and for a strenuous test of convergence behavior, adaptability as well as final effectiveness in solving extremely difficult, multimodal, or large-scale optimization problems. Parameter settings specific for each algorithm are shown in Table 5 such a detailed transparency about experiment results is retained including facility for reproducibility.

#### Performance analysis on IEEE CEC2017 benchmark functions

The proposed Swift Flight Optimizer (SFO)'s performance was extensively validated over the IEEE CEC2017 set of benchmarks in a range of dimensional scenarios. There were two large scenarios considered: 10-dimensional (10D) problems with 50 agents for 1000 iterations and 100-dimensional (100D) problems with 100 agents for 10,000 iterations. These experiments in totality encompass a complete measure of robustness for the optimizer, its scalability, and its adaptability for a wide range of mixed-complexity problems.

According to evaluation with 10D functions, SFO demonstrates remarkable superiority across a broad range of test functions. As summarized in Table 6, SFO achieved the best average fitness in 21 out of 30 benchmark functions, significantly outperforming competing methods. Notably, SFO consistently dominated in the more challenging hybrid and composition functions, where premature convergence often hinders other optimizers. In contrast, alternative algorithms showed weaker competitiveness: EMBGO achieved the best performance in 9 functions, GTP secured the top position in 4 functions, while classical optimizers such as PSO, FDA, MVO, GWO, and HGSO managed only 3 wins combined. Other algorithms, including WOA, MFO, SCA, WSO, SHO, and HLOA, failed to achieve a leading position in any function. These results highlight SFO's ability to navigate complex, multimodal search spaces efficiently, avoiding stagnation and maintaining fast convergence toward optimal solutions.

When scaled to high-dimensional settings (100D, with 100 particles and 10,000 iterations), SFO continued to demonstrate competitive performance, though with expected challenges in scalability. As reported in Table 7, SFO achieved top rankings in 11 out of 30 benchmark functions, placing it among the strongest algorithms for high-dimensional optimization.

No	Name	Abbr	Year	References
1	Enhanced marine battle and gathering optimizer	EMBGO	2025	<sup>53</sup>
2	Horned lizard optimization algorithm	HLOA	2024	<sup>88</sup>
3	Sea-horse optimizer	SHO	2023	<sup>89</sup>
4	White shark optimizer	WSO	2022	<sup>90</sup>
5	Artificial gorilla troops optimizer	GTO	2021	<sup>91</sup>
6	Flow direction algorithm	FDA	2021	<sup>92</sup>
7	Henry gas solubility optimization	HGSO	2019	<sup>93</sup>
8	Whale optimization algorithm	WOA	2016	<sup>50</sup>
9	Multi-verse optimizer	MVO	2016	<sup>63</sup>
10	Sine cosine algorithm	SCA	2016	<sup>74</sup>
11	Moth-flame optimization	MFO	2015	<sup>94</sup>
12	Grey wolf optimizer	GWO	2014	<sup>24</sup>
13	Particle swarm optimization	PSO	1995	<sup>48</sup>

**Table 4.** List of 13 compared optimization algorithms.

Algorithm	Parameter	Value
EMBGO	Radius ratio	Random (0.2, 0.99)
FDA	Flux coefficient	0.3
	diffusion rate	0.7
GTO	Tension rate	0.8
	Aggression level	0.2
GWO	Convergence constant a	[2, 0]
HGSO	l1	5E-03
	l2	100
	L3	1E-02
	Alpha	1
	Beta	1
	M1	0.1
	M2	0.2
HLOA	Free parameters	
MFO	A linearly decreases	From -1 to -2
MVO	WEP <sub>Max</sub>	1
	WEP <sub>Min</sub>	0.2
SCA	r1	Random (0,1)
	r2	Random (0,1)
	r3	Random (0,1)
	r4	Random (0,1)
SHO	U	0.05
	V	0.05
	L	0.05
WOA	Convergence constant a	[2,0]
	Convergence constant b	1
WSO	Hunt depth	0.2
	Speed factor	0.8
PSO	Inertia weight w	0.5
	Cognitive coefficient c1	1.5
	Social coefficient c2	1.5

**Table 5.** Parameter settings for competitor algorithms.

While its dominance was not as pronounced as in the 10D case, SFO maintained robustness in handling multimodal, hybrid, and composition functions, where traditional algorithms often struggle. Competing algorithms such as EMBGO and FDA exhibited stronger performance in selected high-dimensional problems, yet SFO's consistent ranking across functions underscores its balance between exploration and exploitation even as the dimensionality increases. Importantly, the results demonstrate that while performance margins narrow in 100D settings, SFO remains a scalable and reliable optimizer capable of addressing large and complex search spaces.

The comparison between 10 and 100D results highlights two key aspects of SFO's design. First, its adaptive switching mechanism enables rapid convergence in moderate-dimensional landscapes, leading to superior performance against both classical and modern metaheuristics. Second, in high-dimensional contexts, while absolute dominance decreases, SFO still maintains competitive adaptability without requiring extensive parameter tuning, a valuable trait for real-world large-scale optimization problems. This scalability demonstrates SFO's robustness and provides evidence of its potential for deployment in dynamic, high-dimensional applications such as UAV search-and-rescue missions or large-scale engineering optimization tasks.

## Visual analysis of SFO

### *Sensitivity analysis of SFO parameters*

As illustrated in Fig. 4, the sensitivity analysis of the SFO optimizer reveals that it maintains consistent performance across most benchmark functions, reliably achieving competitive fitness values under a broad range of parameter configurations. Functions such as F2, F3, and F4 demonstrate high stability, where the optimizer converges to optimal or near-optimal solutions with minimal sensitivity to variations in parameters such as the number of agents, cognitive and social coefficients ( $c_1$ ,  $c_2$ ), Gaussian scaling ( $\alpha$ ), Lévy scaling ( $\beta$ ), and inertia weight ( $w$ ). These functions exhibit convergence stability particularly for moderate parameter values (e.g.,  $w = 0.5$ ,  $c_1 = c_2 = 1.45$ ,  $\alpha = 0.5$ , and  $\beta = 0.57$ ), requiring minimal fine-tuning.

In contrast, Function F1 displays notable sensitivity, where the optimizer's performance significantly depends on the choice of parameter settings. For F1, extreme values such as high inertia ( $w = 0.9$ ) or overly aggressive

	ALGO	SFO	WOA	EMBGO	MFO	MVO	SCA	GWO	WSO
F1	Mean	<b>1.970E+2</b>	1.042e+07	3.720E+3	6.380E+7	8.230E+3	8.710E+8	1.490E+7	7.190E+6
	STD	<b>4.820E+2</b>	2.130E+7	3.890E+3	2.530E+8	3.910E+3	4.770E+8	5.230E+7	2.970E+7
F2	Mean	<b>2.000E+2</b>	5.900E+6	<b>2.000E+2</b>	5.770E+9	2.570E+2	5.440E+7	3.190E+7	5.350E+9
	STD	<b>3.700E-5</b>	1.310E+7	1.260E-3	2.410E+10	5.140E+1	1.110E+8	1.060E+8	2.280E+10
F3	Mean	<b>3.000E+2</b>	2.630E+3	<b>3.000E+2</b>	7.770E+3	<b>3.000E+2</b>	2.030E+3	2.160E+3	1.360E+3
	STD	5.160E-7	2.140E+3	2.500E-5	8.830E+3	1.570E-2	1.120E+3	2.880E+3	1.570E+3
F4	Mean	<b>4.000E+2</b>	4.370E+2	4.010E+2	4.150E+2	4.040E+2	4.700E+2	4.190E+2	4.180E+2
	STD	<b>4.890E-3</b>	3.460E+1	1.110E+0	2.750E+1	1.670E+0	3.360E+1	2.030E+1	2.480E+1
F5	Mean	<b>5.170E+2</b>	5.530E+2	5.210E+2	5.320E+2	5.200E+2	5.500E+2	<b>5.170E+2</b>	5.250E+2
	STD	<b>4.980E+0</b>	1.812e+01	8.060E+0	1.190E+1	8.270E+0	8.400E+0	7.830E+0	1.550E+1
F6	Mean	6.040E+2	6.390E+2	6.070E+5	6.020E+2	6.020E+2	6.200E+2	6.010E+2	6.050E+2
	STD	3.080E+0	1.610E+1	3.980E+0	2.720E+0	1.640E+0	4.190E+0	1.690E+0	8.590E+0
F7	Mean	<b>7.210E+2</b>	7.870E+2	7.440E+2	7.430E+2	7.260E+2	7.750E+2	7.290E+2	7.440E+2
	STD	<b>4.010E+0</b>	2.270E+1	1.130E+1	1.250E+1	8.540E+0	1.180E+1	9.270E+0	3.200E+1
F8	Mean	<b>8.120E+2</b>	8.440E+2	8.170E+2	8.290E+2	8.210E+2	8.420E+2	8.130E+2	8.200E+2
	STD	5.170E+0	1.860E+1	6.030E+0	1.310E+1	1.010E+1	8.950E+0	4.650E+0	1.440E+1
F9	Mean	<b>9.000E+2</b>	1.500E+3	9.620E+2	1.000E+3	<b>9.000E+2</b>	1.020E+3	9.280E+2	1.070E+3
	STD	6.900E+0	2.990E+2	4.690E+1	2.990E+2	<b>4.310E-1</b>	3.760E+1	4.730E+1	3.170E+2
F10	Mean	1.600E+3	2.070E+3	<b>1.490E+3</b>	2.090E+3	1.670E+3	2.320E+3	1.710E+3	1.830E+3
	STD	1.780E+2	3.660E+2	1.760E+2	2.910E+2	2.730E+2	2.760E+2	2.620E+2	4.290E+2
F11	Mean	<b>1.120E+3</b>	1.220E+3	<b>1.120E+3</b>	1.230E+3	<b>1.120E+3</b>	1.228e+03	1.170E+3	1.180E+3
	STD	2.050E+1	1.290E+2	2.010E+1	8.050E+1	1.360E+1	6.490E+1	7.990E+1	1.080E+2
F12	Mean	<b>5.280E+3</b>	4.700E+6	7.110E+3	1.750E+6	4.680E+5	2.110E+7	4.820E+5	2.270E+6
	STD	<b>3.110E+3</b>	4.860E+6	8.710E+3	4.720E+6	6.720E+5	1.460E+7	7.390E+5	4.520E+6
F13	Mean	1.760E+3	1.633e+04	1.630E+3	9.386e+03	9.926e+03	3.640E+4	1.110E+4	2.174e+04
	STD	2.230E+2	1.174e+04	2.020E+2	1.041e+04	1.002e+04	3.608e+04	9.030E+3	2.719e+04
F14	Mean	2.790E+3	2.439e+03	1.460E+3	3.985e+03	1.540E+3	2.110E+3	3.134e+03	3.179e+03
	STD	1.660E+3	1.350E+3	3.280E+1	3.140E+3	2.220E+2	1.242e+03	1.903e+03	2.139e+03
F15	Mean	2.110E+3	6.571e+03	<b>1.530E+3</b>	7.153e+03	1.734e+03	2.727e+03	5.232e+03	5.868e+03
	STD	9.490E+2	5.125e+03	<b>2.380E+1</b>	8.969e+03	3.925e+02	9.362e+02	2.823e+03	5.694e+03
F16	Mean	<b>1.650E+3</b>	1.860e+03	<b>1.650E+3</b>	1.764e+03	1.733e+03	1.740e+03	1.736e+03	1.791e+03
	STD	5.410E+1	1.187E+2	<b>5.110E+1</b>	1.051E+2	9.760E+1	7.174E+1	8.648E+1	1.443E+2
F17	Mean	<b>1.740E+3</b>	1.790E+3	1.750E+3	1.770E+3	1.780E+3	1.780E+3	1.770E+3	1.770E+3
	STD	1.480E+1	4.770E+1	1.090E+1	4.410E+1	4.770E+1	1.000E+1	3.600E+1	3.650E+1
F18	Mean	5.300E+3	1.970E+4	<b>1.910E+3</b>	2.210E+4	1.800E+4	1.510E+5	2.670E+4	4.760E+4
	STD	3.130E+3	1.200E+4	<b>5.940E+1</b>	1.570E+4	1.440E+4	1.030E+5	1.570E+4	6.350E+4
F19	Mean	2.100E+3	1.020E+5	<b>1.920E+3</b>	2.490E+4	2.220E+3	5.630E+3	9.230E+3	9.160E+3
	STD	6.090E+2	2.160E+5	<b>1.680E+1</b>	4.820E+4	5.820E+2	5.170E+3	7.080E+3	6.810E+3
F20	Mean	<b>2.045E+3</b>	2.220E+3	2.050E+3	2.100E+3	2.120E+3	2.090E+3	2.080E+3	2.110E+3
	STD	2.540E+1	5.810E+1	1.240E+1	6.590E+1	7.890E+1	2.200E+1	5.650E+1	7.570E+1
F21	Mean	<b>2.210E+3</b>	2.330E+3	2.260E+3	2.330E+3	2.280E+3	2.270E+3	2.320E+3	2.320E+3
	STD	6.490E+1	5.100E+1	5.220E+1	3.110E+1	5.350E+1	6.650E+1	<b>7.680E+0</b>	3.330E+1
F22	Mean	<b>2.290E+3</b>	2.380E+3	2.300E+3	2.310E+3	2.330E+3	2.370E+3	2.310E+3	2.310E+3
	STD	2.450E+1	2.820E+2	2.650E+1	1.710E+1	1.580E+2	3.950E+1	2.350E+1	<b>4.830E+0</b>
F23	Mean	<b>2.616E+3</b>	2.660E+3	2.620E+3	2.630E+3	2.620E+3	2.660E+3	2.630E+3	2.630E+3
	STD	<b>5.950E+0</b>	2.520E+1	1.370E+1	1.050E+1	1.020E+1	8.510E+0	1.120E+1	1.600E+1
F24	Mean	2.680E+3	2.770E+3	2.600E+3	2.770E+3	2.730E+3	2.790E+3	2.740E+3	2.750E+3
	STD	1.110E+2	7.550E+1	9.040E+1	9.440E+0	6.360E+1	<b>5.460E+0</b>	5.680E+1	6.060E+1
F25	Mean	<b>2.890E+3</b>	2.950E+3	2.980E+3	2.950E+3	2.920E+3	2.960E+3	2.930E+3	2.940E+3
	STD	5.560E+1	2.290E+1	2.350E+1	3.410E+1	3.310E+1	2.540E+1	1.720E+1	2.930E+1
F26	Mean	<b>2.870E+3</b>	3.510E+3	2.980E+3	3.120E+3	2.960E+3	3.100E+3	3.050E+3	3.140E+3
	STD	8.390E+1	5.690E+2	1.140E+2	3.540E+2	2.700E+2	<b>4.840E+1</b>	2.930E+2	4.170E+2
F27	Mean	<b>2.090E+3</b>	3.140E+3	3.100E+3	3.090E+3	3.100E+3	3.100E+3	3.100E+3	3.110E+3
	STD	2.610E+0	4.280E+1	4.440E+0	3.950E+0	2.050E+1	<b>1.850E+0</b>	1.440E+1	3.310E+1
F28	Mean	<b>3.170E+3</b>	3.360E+3	3.320E+3	3.350E+3	3.300E+3	3.310E+3	3.370E+3	3.370E+3
	STD	1.090E+2	1.480E+2	1.020E+2	7.930E+1	1.370E+2	8.960E+1	8.150E+1	1.070E+2

Continued

	ALGO	SFO	WOA	EMBGO	MFO	MVO	SCA	GWO	WSO
F29	Mean	<b>3.180E+3</b>	3.360E+3	<b>3.180E+3</b>	3.230E+3	3.220E+3	3.250E+3	3.210E+3	3.230E+3
	STD	2.970E+1	1.260E+2	<b>2.330E+1</b>	5.540E+1	5.980E+1	4.400E+1	4.610E+1	7.210E+1
F30	Mean	1.380E+5	9.500E+5	4.690E+4	7.900E+5	5.090E+5	1.400E+6	5.250E+5	5.820E+5
	STD	3.920E+5	1.240E+6	1.730E+5	1.050E+6	6.350E+5	7.040E+5	7.020E+5	6.670E+5
Win		21	0	9	0	3	0	1	0
	ALGO	HGSO	SHO	HLOA	FDA	GTO	PSO		
F1	Mean	1.023E+9	8.416E+8	1.788E+4	1.809E+3	2.442E+3	1.340E+3		
	STD	3.484E+8	1.193E+9	3.329E+4	1.888E+3	2.003E+3	1.407+e03		
F2	Mean	1.653E+8	1.595E+9	4.904E+2	<b>2.000E+2</b>	<b>2.000E+2</b>	<b>2.000E+2</b>		
	STD	3.053E+8	5.124E+9	6.850E+2	6.204E-5	7.072E-4	3.704E-5		
F3	Mean	2.035E+3	4.143E+3	3.054E+2	<b>3.000E+2</b>	<b>3.000E+2</b>	<b>3.000E+2</b>		
	STD	9.893E+2	2.962E+3	1.173E+1	9.837E-11	1.389E-11	<b>4.088E-14</b>		
F4	Mean	4.711E+2	4.510E+2	4.131E+2	4.008E+2	4.051E+2	4.051E+2		
	STD	1.888E+1	4.693E+1	2.185E+1	7.639E-1	1.475E+1	1.221E+1		
F5	Mean	5.558E+2	5.379E+2	5.769E+2	5.193E+2	5.239E+2	5.491E+2		
	STD	5.101E+0	1.290E+1	2.643E+1	7.433E+0	9.522E+0	1.479E+1		
F6	Mean	6.270e+02	6.133E+2	6.480E+2	<b>6.001E+2</b>	6.054E+2	6.109E+2		
	STD	5.925E+0	5.742E+0	1.443E+1	<b>3.753E-1</b>	4.440E+0	8.715E+0		
F7	Mean	7.808E+2	7.626E+2	7.914E+2	7.371E+2	7.532E+2	7.279E+2		
	STD	9.853E+0	1.323E+1	2.108E+1	1.222E+1	1.514E+1	7.927E+0		
F8	Mean	8.377E+2	8.260E+2	8.457E+2	8.249E+2	8.263E+2	8.193E+2		
	STD	<b>3.650E+0</b>	7.728E+0	2.250E+1	8.238E+0	7.890E+0	8.026E+0		
F9	Mean	1.034E+3	1.019E+3	1.618E+3	9.039E+2	9.536E+2	9.316E+2		
	STD	3.697E+1	7.456E+1	1.758E+2	5.486E+0	7.393E+1	7.878e+01		
F10	Mean	2.640E+3	1.914E+3	2.407E+3	1.783E+3	1.831E+3	1.985E+3		
	STD	<b>1.612E+2</b>	3.819E+2	2.465E+2	2.730E+2	2.857E+2	2.526E+2		
F11	Mean	1.340E+3	1.180E+3	1.217E+3	1.111e+03	1.124e+03	1.141E+3		
	STD	9.261E+1	8.182E+1	9.184E+1	<b>8.964E+0</b>	1.600E+1	2.386E+1		
F12	Mean	1.167E+7	8.383e+05	2.082E+5	1.160E+4	1.220E+4	1.212E+4		
	STD	6.462E+6	5.667E+5	4.427E+5	1.260E+4	1.111E+4	8.917E+3		
F13	Mean	5.161e+04	8.975E+3	3.483E+3	1.607E+3	<b>1.482E+3</b>	7.611E+3		
	STD	4.556e+04	5.041E+3	2.637E+3	2.861E+2	<b>1.283E+2</b>	6.025E+3		
F14	Mean	3.284e+03	3.892E+3	4.330E+3	1.470E+3	<b>1.457E+3</b>	2.143E+3		
	STD	1.292e+03	1.297E+3	8.357E+3	3.200E+1	<b>2.755E+1</b>	9.978e+02		
F15	Mean	6.357e+03	3.097E+3	3.149E+3	1.602E+3	1.585E+3	2.573E+3		
	STD	2.417e+03	1.104E+3	2.739E+3	7.718E+1	8.197E+1	1.120E+3		
F16	Mean	1.904e+03	1.797E+3	2.070E+3	1.670E+3	1.689E+3	1.900E+3		
	STD	1.443e+02	1.430E+2	1.921E+2	6.042E+1	1.009E+2	1.254E+2		
F17	Mean	1.787E+3	1.755E+3	1.927E+3	1.745E+3	1.748E+3	1.756E+3		
	STD	<b>9.098E+0</b>	2.004E+1	1.267E+2	2.393E+1	1.871E+1	3.427E+1		
F18	Mean	4.964E+5	1.993E+4	1.164E+4	2.070E+3	2.143E+3	1.282E+4		
	STD	3.398E+5	1.015E+4	1.486E+4	2.572E+2	4.727E+2	9.793E+3		
F19	Mean	1.492E+4	6.792E+3	3.228E+3	1.946E+3	1.930E+3	4.763E+3		
	STD	8.698E+3	4.429E+3	4.039E+3	3.333E+1	3.678E+1	4.173E+3		
F20	Mean	2.156E+3	2.074E+3	2.277E+3	2.047E+3	2.056E+3	2.119E+3		
	STD	3.908E+1	5.345E+1	9.481E+1	<b>1.191E+1</b>	3.048E+1	6.677E+1		
F21	Mean	2.292E+3	2.313E+3	2.347E+3	2.268E+3	2.213E+3	2.311E+3		
	STD	4.861E+1	4.240E+1	5.425E+1	6.187E+1	2.625E+1	5.750E+1		
F22	Mean	2.412E+3	2.380E+3	2.514E+3	2.300E+3	2.298E+3	2.486E+3		
	STD	5.048E+1	5.580E+1	5.153E+2	1.469E+1	2.096E+1	4.568E+2		
F23	Mean	2.685E+3	2.655E+3	2.720E+3	2.619E+3	2.623E+3	2.698E+3		
	STD	1.009E+1	1.446E+1	5.234E+1	8.248E+0	1.099E+1	3.495E+1		
F24	Mean	<b>2.595E+3</b>	2.761E+3	2.836E+3	2.745E+3	2.692E+3	2.763E+3		
	STD	6.029E+1	6.299E+1	4.199E+1	5.874E+1	1.140E+2	1.379E+2		
F25	Mean	2.971E+3	2.958E+3	2.936E+3	2.932E+3	2.930E+3	2.925E+3		
	STD	<b>1.611E+1</b>	9.723E+1	4.012E+1	2.254E+1	2.385E+1	2.803E+1		
Continued									

	ALGO	HGSO	SHO	HLOA	FDA	GTO	PSO
F26	Mean	3.302E+3	3.137E+3	3.732E+3	2.935E+3	2.938E+3	3.165E+3
	STD	1.095E+2	2.435E+2	4.845E+2	7.643E+1	1.141E+2	3.764E+2
F27	Mean	3.138E+3	3.129E+3	3.232E+3	3.093E+3	3.096E+3	3.166E+3
	STD	1.684E+1	2.164E+1	8.363E+1	3.218E+0	7.079E+0	5.448E+1
F28	Mean	3.455E+3	3.415E+3	3.407E+3	3.243E+3	3.268E+3	3.211E+3
	STD	1.568E+2	1.764E+2	2.118E+2	1.452E+2	1.730E+2	<b>5.896E+1</b>
F29	Mean	3.268E+3	3.263E+3	3.459E+3	3.201E+3	3.212E+3	3.240E+3
	STD	2.569E+1	5.276E+1	1.835E+2	5.433E+1	5.267E+1	4.462E+1
F30	Mean	3.153E+5	6.753E+5	1.740E+6	1.819E+5	1.831E+6	<b>1.168E+4</b>
	STD	4.106E+5	7.449E+5	3.554E+6	3.776E+5	2.372E+6	<b>6.634E+3</b>
Win		1	0	0	3	4	3

**Table 6.** Performance on IEEE CEC2017 benchmark functions, population size = 50, dimension = 10, iteration = 1000. Significant values are in bold.

exploration ( $\beta=2.0$ ) can result in performance degradation, and in some cases, lead to unstable or divergent search behavior. Nevertheless, under well-balanced configurations, SFO is still capable of converging toward optimal regions. Overall, the optimizer demonstrates strong robustness and convergence characteristics, with only slight to moderate sensitivity in select functions, affirming its reliability with limited tuning requirements across diverse optimization landscapes.

#### *SFO search history, trajectory, and average fitness analysis*

Figures 5 and 6 comprehensively illustrate the performance of the proposed Swift Flight Optimizer (SFO) on the CEC2017 benchmark functions F1–F10. Each row in these figures presents four key perspectives: the 3D surface of the objective function (first column), the spatial search history of the particles (second column), the trajectory of the first particle over iterations (third column), and the average fitness convergence curve (fourth column).

From Fig. 5, it is evident that for functions F1–F5, SFO demonstrates a clear two-phase behavior. Initially, the search is widely exploratory, as reflected by the diverse distribution of particle positions in the second column and large positional changes in the trajectory plots. As the iterations progress, the algorithm transitions smoothly into an exploitation phase, with the first particle converging toward the optimal region. The average fitness plots confirm this behavior, exhibiting rapid and stable convergence within the first 200 iterations, especially for F1, F2, and F3, where the optimizer reaches extremely low objective values, confirming high precision. Figure 6 presents the optimizer's performance on the more rugged and multimodal functions F6–F10. Even under these challenging conditions, SFO maintains its adaptive behavior. The search history plots continue to show a dense cluster of particle positions around the minima, while the trajectory plots reveal a gradual dampening of movements signifying stabilization near global optima. The fitness curves further confirm this trend, with a consistent and monotonic decrease across most functions, despite minor oscillations in functions such as F7 and F8, due to their highly repetitive landscapes.

Moreover, Fig. 7 illustrates the 3D trajectories of the first particle of the Swift Flight Optimizer (SFO) across the CEC2017 benchmark functions F1–F9. The optimizer on functions F1–F6 exhibits a dynamic and adaptive search behavior characterized by a well-balanced transition between exploration and exploitation. In the early stages of the search process, the trajectory plots clearly show wide movements, indicating that the particle initiates a broad exploration phase to scan large areas of the search space. As optimization advances, the particles path becomes increasingly refined, with trajectory steps shrinking and converging toward specific regions in the landscape. This shift reflects a deliberate transition into a focused exploitation phase. The optimizer thus demonstrates an effective capacity to identify promising regions early and intensify its search in their vicinity, avoiding premature convergence. The smooth reduction in movement magnitude further confirms that SFO systematically guides the particle from global exploration toward high-precision convergence near the global optima. The 3D trajectories of the first particle of the SFO over the CEC2017 benchmark functions F7–F9, highlighting the optimizer's capability to maintain adaptive search dynamics in more complex and rugged landscapes. In F7 and F8, the optimizer demonstrates a relatively scattered but directed search pattern, where initial steps explore the function surface broadly, followed by a gradual narrowing of the search trajectory. This behavior illustrates SFO's robustness in avoiding stagnation and adapting to functions with multiple local optima.

#### *SFO exploration and exploitation analysis*

The exploration–exploitation balance of the SFO can be visually assessed through the exploration–exploitation ratio<sup>95,96</sup>, and it is expressed as:

$$div(t) = \sum_{i=1}^n \sum_{j=1}^d |median(x_j(t)) - x_{ij}(t)| \quad (7)$$

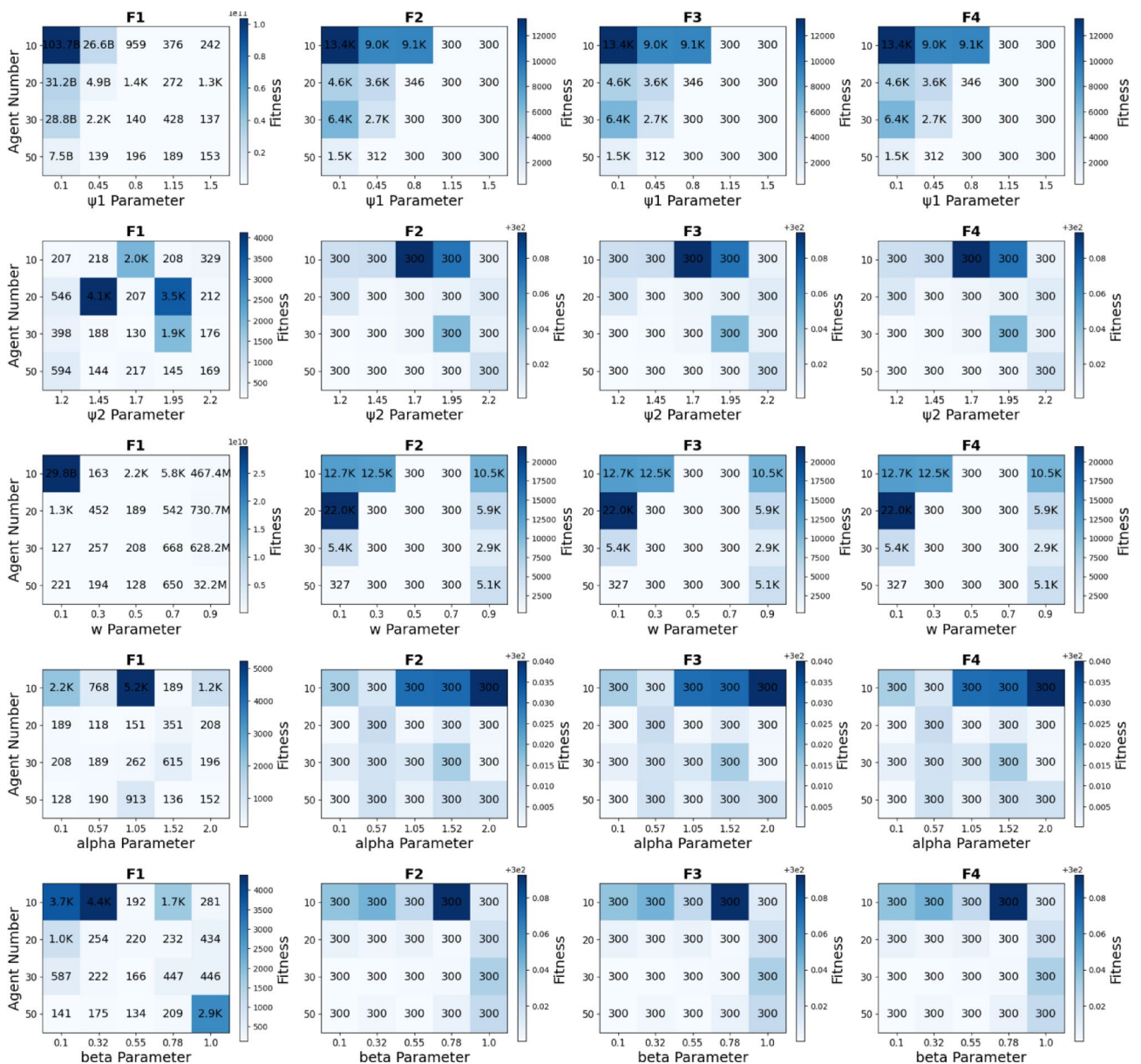
$$Exploration\% = \frac{div(t)}{div_{max}} \times 100\% \quad (8)$$

	ALGO	SFO	WOA	EMBGO	MFO	MVO	SCA	GWO	WSO
F1	Mean	<b>3.549E+02</b>	3.649E+07	1.530E+10	1.284E+11	2.430E+05	1.515E+11	3.055E+10	4.288E+09
	STD	<b>2.749E+02</b>	1.940E+07	1.955E+10	4.702E+10	3.388E+04	9.846E+09	7.873E+09	3.288E+09
F2	Mean	<b>1.075E+19</b>	2.890E+133	9.163E+100	1.389E+150	2.046E+42	1.537E+148	3.875E+137	4.090E+68
	STD	<b>4.287E+19</b>	9.106E+133	4.934E+101	2.892E+150	6.437E+42	4.224E+148	1.225E+138	1.108E+69
F3	Mean	<b>3.001E+02</b>	5.826E+05	3.633E+03	4.761E+05	4.621E+02	2.868E+05	1.978E+05	5.775E+04
	STD	<b>4.478E-02</b>	1.536E+05	1.802E+03	1.956E+05	5.386E+01	2.033E+04	2.296E+04	1.047E+04
F4	Mean	<b>5.511E+02</b>	9.762E+02	9.341E+02	1.955E+04	6.424E+02	2.649E+04	2.837E+03	1.065E+03
	STD	<b>1.856E+01</b>	1.195E+02	1.218E+02	1.164E+04	2.965E+01	2.133E+03	4.911E+02	2.984E+02
F5	Mean	1.036E+03	1.410E+03	1.268E+03	1.673E+03	<b>9.415E+02</b>	1.857E+03	1.068E+03	1.084E+03
	STD	5.188E+03	6.865E+01	4.496E+01	1.546E+02	7.238E+01	<b>3.950E+01</b>	1.041E+02	5.761E+01
F6	Mean	<b>6.250E+02</b>	6.833E+02	6.726E+02	6.668E+02	6.412E+02	6.893E+02	6.259E+02	6.317E+02
	STD	3.829E+00	1.205E+01	4.184E+00	9.004E+00	6.853E+00	3.914E+00	3.431E+00	3.875E+00
F7	Mean	1.969E+03	3.262E+03	2.997E+03	4.352E+03	1.427E+03	3.326E+03	1.767E+03	2.452E+03
	STD	1.504E+02	1.193E+02	2.139E+02	5.948E+02	1.004E+02	1.390E+02	1.715E+02	1.300E+02
F8	Mean	<b>1.250E+03</b>	1.840E+03	1.737E+03	2.019E+03	1.286E+03	2.198E+03	1.364E+03	1.459E+03
	STD	5.183E+01	9.113E+01	7.159E+01	1.499E+02	1.104E+02	5.210E+01	1.883E+02	<b>4.661E+01</b>
F9	Mean	<b>1.623E+04</b>	3.854E+04	2.063E+04	4.404E+04	2.750E+04	6.338E+04	2.228E+04	5.055E+04
	STD	2.121E+03	6.497E+03	1.518E+03	9.460E+03	1.104E+04	5.420E+03	1.177E+04	4.067E+03
F10	Mean	1.349E+04	2.111E+04	1.538E+04	1.641E+04	1.337E+04	3.106E+04	1.355E+04	<b>1.177E+04</b>
	STD	<b>6.831E+02</b>	2.146E+03	1.459E+03	1.132E+03	1.064E+03	9.480E+02	1.255E+03	1.077E+03
F11	Mean	2.195E+03	6.687E+03	2.509E+03	1.330E+05	2.560E+03	6.551E+04	4.208E+04	3.863E+03
	STD	1.659E+02	1.457E+03	3.079E+02	8.815E+04	2.379E+02	1.185E+04	9.125E+03	3.408E+03
F12	Mean	5.699E+06	6.410E+08	6.307E+07	2.959E+10	1.151E+08	5.259E+10	4.576E+09	2.166E+07
	STD	1.219E+06	2.037E+08	1.331E+08	1.295E+10	5.102E+07	6.387E+09	3.339E+09	3.614E+07
F13	Mean	7.196E+04	1.013E+05	1.493E+04	4.896E+09	9.616E+04	8.126E+09	5.059E+08	7.056E+03
	STD	1.727E+04	5.454E+04	7.020E+03	3.719E+09	3.490E+04	2.533E+09	5.316E+08	2.334E+03
F14	Mean	4.957E+04	1.948E+06	6.535E+04	1.260E+07	6.334E+05	2.086E+07	2.540E+06	5.888E+04
	STD	2.710E+04	9.584E+05	8.995E+04	2.590E+07	2.707E+05	7.792E+06	1.544E+06	3.593E+04
F15	Mean	3.922E+04	1.187E+05	5.296E+03	5.620E+08	1.017E+05	2.352E+09	1.522E+08	5.492E+03
	STD	1.215E+04	2.140E+05	3.736E+03	5.341E+08	2.042E+04	5.266E+08	1.623E+08	3.691E+03
F16	Mean	<b>4.876E+03</b>	9.634E+03	6.302E+03	7.620E+03	4.922E+03	1.222E+04	5.372E+03	5.068E+03
	STD	4.554E+02	1.153E+03	6.074E+02	5.909E+02	6.389E+02	6.166E+02	<b>4.528E+02</b>	6.025E+02
F17	Mean	5.003E+03	6.564E+03	6.061E+03	1.139E+04	<b>4.293E+03</b>	1.182E+04	4.679E+03	5.192E+03
	STD	3.333E+02	6.632E+02	5.282E+02	1.158E+04	4.341E+02	3.266E+03	6.048E+02	6.817E+02
F18	Mean	2.008E+05	2.277E+06	2.214E+05	7.194E+06	1.052E+06	2.704E+07	4.602E+06	8.603E+04
	STD	5.522E+04	8.885E+05	2.166E+05	8.571E+06	5.525E+05	7.470E+06	4.997E+06	2.354E+04
F19	Mean	7.944E+04	1.685E+07	5.446E+03	8.016E+08	5.228E+06	2.227E+09	7.995E+07	4.360E+03
	STD	5.457E+04	4.804E+06	3.672E+03	8.431E+08	1.874E+06	5.939E+08	6.166E+07	2.264E+03
F20	Mean	4.349E+03	6.006E+03	4.867E+03	5.730E+03	4.566E+03	7.168E+03	4.013E+03	<b>3.722E+03</b>
	STD	2.841E+02	5.175E+02	4.464E+02	6.107E+02	6.291E+02	3.214E+02	3.691E+02	2.860E+02
F21	Mean	<b>2.791E+03</b>	3.974E+03	3.154E+03	3.545E+03	2.794E+03	3.873E+03	2.868E+03	3.219E+03
	STD	5.750E+01	1.103E+02	8.923E+01	6.422E+01	6.088E+01	6.123E+01	<b>5.243E+01</b>	1.389E+02
F22	Mean	<b>1.415E+04</b>	2.471E+04	1.923E+04	2.085E+04	1.442E+04	3.371E+04	1.729E+04	1.898E+04
	STD	7.234E+02	2.299E+03	1.350E+03	1.611E+03	1.711E+03	<b>4.734E+02</b>	2.435E+03	2.351E+03
F23	Mean	3.899E+03	4.805E+03	3.980E+03	3.693E+03	<b>3.320E+03</b>	4.866E+03	3.381E+03	4.002E+03
	STD	9.797E+01	2.373E+02	1.883E+02	8.182E+01	8.155E+01	8.464E+01	<b>5.454E+01</b>	2.069E+02
F24	Mean	4.507E+03	6.040E+03	5.125E+03	4.267E+03	<b>3.704E+03</b>	6.355E+03	3.948E+03	5.168E+03
	STD	1.555E+02	3.281E+02	2.523E+02	1.002E+02	<b>4.764E+01</b>	1.484E+02	9.059E+01	2.138E+02
F25	Mean	<b>3.215E+03</b>	3.589E+03	3.654E+03	1.214E+04	3.224E+03	1.421E+04	5.104E+03	3.695E+03
	STD	<b>3.493E+01</b>	8.410E+01	1.815E+02	4.686E+03	7.828E+01	1.068E+03	3.403E+02	1.787E+02
F26	Mean	1.675E+04	3.126E+04	2.762E+04	1.709E+04	<b>1.054E+04</b>	3.156E+04	1.233E+04	1.907E+04
	STD	3.911E+03	5.027E+03	8.432E+03	1.026E+03	<b>6.394E+02</b>	1.949E+03	1.144E+03	7.725E+03
F27	Mean	3.556E+03	5.157E+03	4.455E+03	3.930E+03	3.375E+03	6.925E+03	3.857E+03	4.758E+03
	STD	5.341E+01	8.432E+02	3.902E+02	1.939E+02	3.458E+01	2.303E+02	1.004E+02	3.022E+02
F28	Mean	3.375E+03	3.751E+03	<b>1.764E+03</b>	1.773E+04	3.389E+03	1.959E+04	6.373E+03	3.997E+03
	STD	<b>1.876E+01</b>	5.106E+01	1.762E+02	1.628E+03	2.120E+01	2.551E+03	8.013E+02	2.683E+02
Continued									

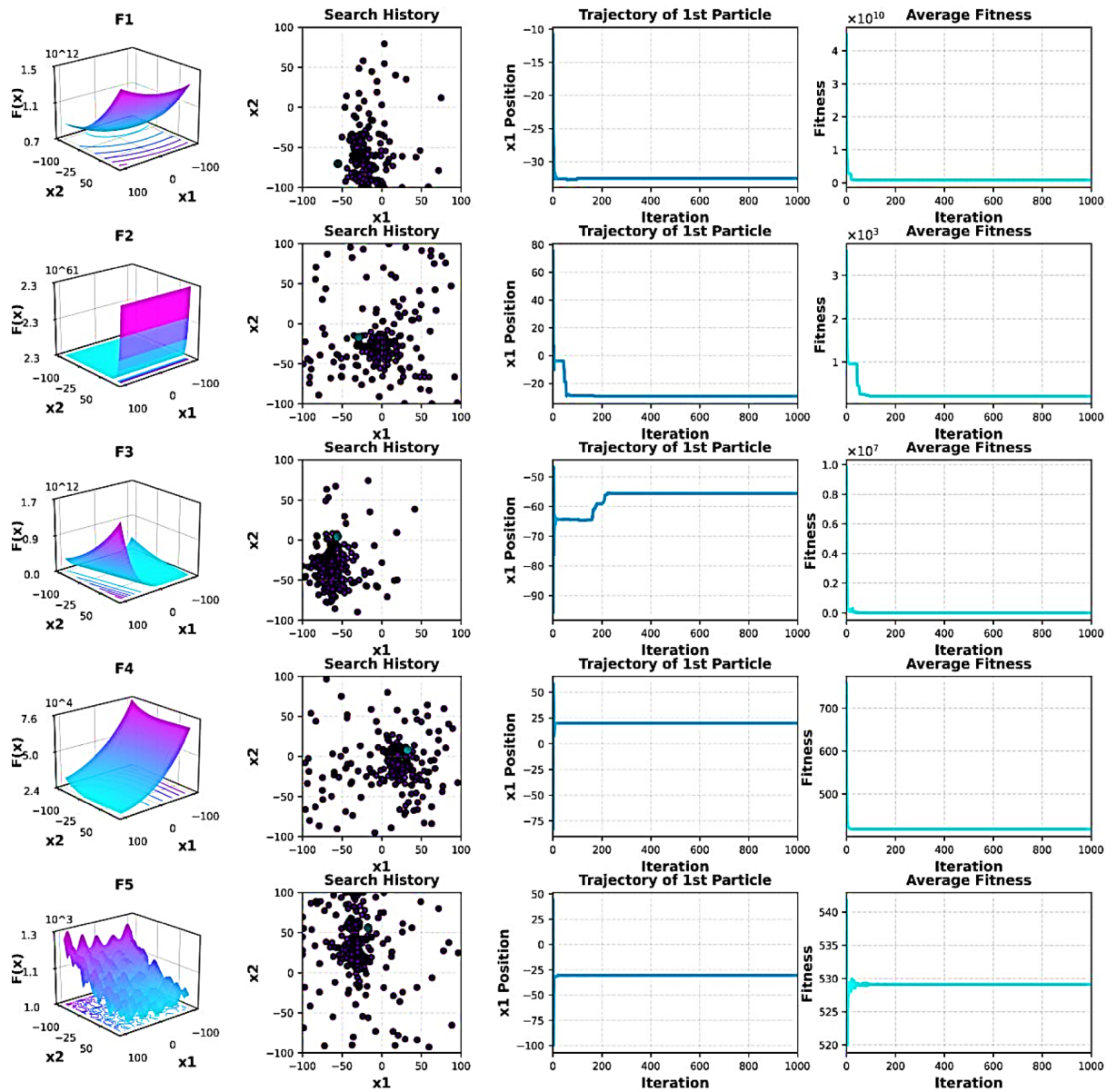
	ALGO	SFO	WOA	EMBGO	MFO	MVO	SCA	GWO	WSO
F29	Mean	6.932E+03	1.394E+04	8.846E+03	9.494E+03	6.758E+03	1.452E+04	7.448E+03	7.216E+03
	STD	<b>3.543E+02</b>	1.891E+03	5.877E+03	2.284E+03	5.153E+02	9.543E+02	3.733E+02	7.416E+02
F30	Mean	5.473E+05	1.778E+08	5.543E+7	1.031E+09	2.955E+07	5.993E+09	5.145E+08	2.473E+05
	STD	1.539E+05	7.236E+07	7.234E+6	9.971E+08	1.021E+07	1.366E+09	3.559E+08	3.206E+05
Win		11	0	1	0	5	0	0	2
	ALGO	HGSO	SHO	HLOA	FDA	GTO	PSO		
F1	Mean	1.479E+11	1.720E+11	1.223E+07	7.951E+03	6.708E+03	3.266E+03		
	STD	2.437E+10	1.192E+10	1.078E+07	1.001E+04	8.500E+03	3.957E+03		
F2	Mean	3.488E+146	6.973E+150	4.187E+70	1.333E+44	1.996E+27	1.138E+40		
	STD	1.072E+147	2.205E+151	1.312E+71	4.183E+44	6.308E+27	1.218E+40		
F3	Mean	2.830E+05	2.739E+05	3.238E+03	<b>3.001E+02</b>	4.038E+02	1.755E+04		
	STD	1.598E+04	1.488E+04	9.114E+02	1.336E-01	1.692E+02	7.383E+03		
F4	Mean	2.854E+04	2.766E+04	7.375E+02	5.749E+02	6.485E+02	5.551E+02		
	STD	4.952E+03	4.371E+03	6.670E+01	3.696E+01	4.315E+01	5.937E+01		
F5	Mean	1.824E+03	1.641E+03	1.416E+03	1.330E+03	1.318E+03	1.202E+03		
	STD	5.093E+01	4.666E+01	6.282E+01	1.058E+02	5.175E+01	4.032E+01		
F6	Mean	6.946E+02	6.631E+02	6.683E+02	6.544E+02	6.585E+02	6.501E+02		
	STD	2.429E+00	3.392E+00	2.469E+00	3.137E+00	3.499E+00	<b>1.709E+00</b>		
F7	Mean	3.279E+03	2.992E+03	3.326E+03	2.598E+03	2.934E+03	<b>1.217E+03</b>		
	STD	9.540E+01	1.789E+02	1.033E+02	3.325E+02	2.059E+02	<b>8.001E+01</b>		
F8	Mean	2.237E+03	1.980E+03	1.834E+03	1.686E+03	1.722E+03	1.503E+03		
	STD	6.477E+01	7.824E+01	7.426E+01	1.383E+02	1.071E+02	8.092E+01		
F9	Mean	6.538E+04	3.396E+04	2.904E+04	2.422E+04	2.160E+04	1.837E+04		
	STD	3.184E+03	3.098E+03	2.104E+03	3.433E+03	<b>1.188E+03</b>	1.855E+03		
F10	Mean	2.659E+04	2.042E+04	2.073E+04	1.681E+04	1.630E+04	1.328E+04		
	STD	8.140E+02	1.188E+03	2.835E+03	8.589E+02	2.082E+03	1.186E+03		
F11	Mean	1.275E+05	7.241E+04	6.789E+03	2.158E+03	1.991E+03	<b>1.936E+03</b>		
	STD	1.420E+04	1.975E+04	7.006E+03	1.412E+02	1.685E+02	<b>1.078E+02</b>		
F12	Mean	6.654E+10	7.744E+10	4.995E+07	<b>7.082E+05</b>	1.260E+06	2.291E+06		
	STD	1.097E+10	1.595E+10	1.536E+07	5.470E+05	<b>4.723E+05</b>	9.193E+05		
F13	Mean	9.855E+09	1.068E+10	7.038E+04	5.687E+03	8.361E+03	<b>5.268E+03</b>		
	STD	2.734E+09	3.783E+09	5.248E+04	3.970E+03	3.274E+03	<b>8.198E+02</b>		
F14	Mean	1.580E+07	7.009E+06	1.382E+05	<b>7.354E+03</b>	2.251E+04	3.139E+05		
	STD	2.630E+06	3.230E+06	7.686E+04	<b>6.010E+03</b>	9.400E+03	1.619E+05		
F15	Mean	2.427E+09	2.411E+09	1.120E+04	<b>2.693E+03</b>	6.880E+03	3.968E+03		
	STD	8.320E+08	1.321E+09	5.786E+03	<b>9.729E+02</b>	7.629E+03	2.786E+03		
F16	Mean	1.208E+04	9.459E+03	7.392E+03	5.964E+03	5.956E+03	5.036E+03		
	STD	7.625E+02	1.114E+03	1.220E+03	6.974E+02	8.069E+02	4.942E+02		
F17	Mean	1.952E+04	3.691E+04	6.353E+03	5.249E+03	5.440E+03	4.677E+03		
	STD	1.351E+04	4.553E+04	6.533E+02	5.226E+02	6.297E+02	<b>2.945E+02</b>		
F18	Mean	2.308E+07	6.547E+06	2.990E+05	<b>5.679E+04</b>	6.308E+04	5.586E+05		
	STD	4.662E+06	3.951E+06	1.258E+05	4.509E+04	<b>1.622E+04</b>	9.558E+04		
F19	Mean	3.000E+09	1.826E+09	2.949E+04	<b>3.524E+03</b>	7.467E+03	4.423E+03		
	STD	6.658E+08	1.448E+09	3.413E+04	<b>1.737E+03</b>	4.143E+03	3.063E+03		
F20	Mean	6.576E+03	5.079E+03	6.460E+03	5.463E+03	5.373E+03	5.206E+03		
	STD	<b>2.391E+02</b>	4.434E+02	8.479E+02	3.328E+02	3.999E+02	4.104E+02		
F21	Mean	4.093E+03	3.374E+03	4.160E+03	3.169E+03	3.227E+03	3.397E+03		
	STD	1.578E+02	1.114E+02	1.897E+02	1.224E+02	1.650E+02	7.001E+01		
F22	Mean	3.059E+04	2.623E+04	2.207E+04	2.015E+04	2.110E+04	1.599E+04		
	STD	9.873E+02	9.981E+02	1.728E+03	1.157E+03	2.264E+03	1.212E+03		
F23	Mean	6.155E+03	4.207E+03	5.094E+03	3.627E+03	4.034E+03	4.632E+03		
	STD	8.872E+02	1.067E+02	4.027E+02	1.522E+02	3.134E+02	3.174E+02		
F24	Mean	7.016E+03	6.005E+03	7.435E+03	4.234E+03	4.697E+03	4.884E+03		
	STD	2.801E+02	4.224E+02	6.842E+02	1.818E+02	3.034E+02	3.063E+02		
F25	Mean	1.385E+04	1.443E+04	3.421E+03	3.248E+03	3.277E+03	3.218E+03		
	STD	1.845E+03	1.864E+03	5.950E+01	9.490E+01	7.403E+01	4.202E+01		
Continued									

	ALGO	HGSO	SHO	HLOA	FDA	GTO	PSO
F26	Mean	3.477E+04	3.408E+04	3.715E+04	2.050E+04	2.279E+04	1.151E+04
	STD	3.169E+03	2.330E+03	5.154E+03	3.086E+03	2.440E+03	7.956E+03
F27	Mean	<b>3.200E+03</b>	5.484E+03	4.648E+03	3.740E+03	3.875E+03	3.309E+03
	STD	<b>1.497E-04</b>	5.788E+02	3.988E+02	1.607E+02	1.821E+02	8.405E+01
F28	Mean	1.908E+04	1.768E+04	3.500E+03	3.339E+03	3.376E+03	3.321E+03
	STD	2.388E+03	2.080E+03	5.875E+01	2.673E+01	2.797E+01	2.690E+01
F29	Mean	1.655E+04	1.274E+04	1.016E+04	7.339E+03	7.433E+03	<b>6.586E+03</b>
	STD	2.275E+03	2.658E+03	1.071E+03	4.494E+02	8.009E+02	4.902E+02
F30	Mean	8.520E+09	9.135E+09	1.219E+06	7.716E+03	1.219E+04	<b>4.875E+03</b>
	STD	2.663E+09	3.835E+09	1.279E+06	<b>1.381E+03</b>	4.546E+03	1.573E+03
Win		1	0	0	6	0	4

**Table 7.** Performance on IEEE CEC2017 benchmark functions, population size = 100, dimension = 100, iteration = 10,000. Significant values are in bold.



**Fig. 4.** Sensitivity heatmaps for SFO parameters on CEC2017 benchmark functions F1–F4 at dim = 10.

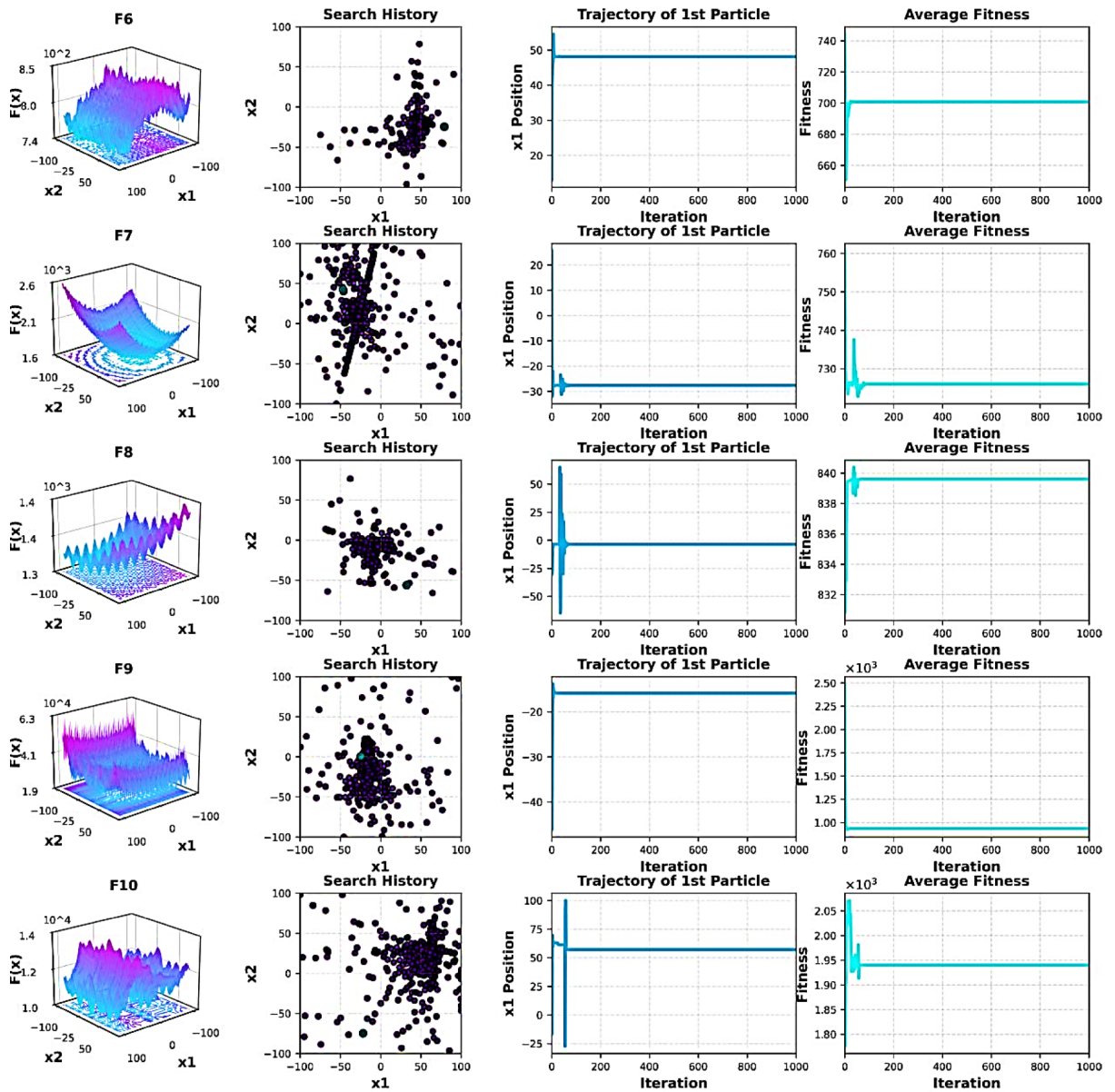


**Fig. 5.** Search history, trajectory of the first agent, and average fitness of SFO optimizer over selected functions F1, F2, F3, F4 and F5 of IEEE CEC2017 functions.

$$Exploitation\% = \frac{|div(t) - div_{max}|}{div_{max}} \times 100\% \tag{9}$$

where  $x_{ij}$  represents the value of the  $j$ th dimension for the  $i$ th individual. Figure 8 provides a comprehensive visualization of the dynamic exploration–exploitation behavior exhibited by the Swift Flight Optimizer (SFO) across all 30 benchmark functions in the CEC2017 suite at 10-dimensional settings. In the early iterations of most functions, SFO exhibits a high level of exploration activity, effectively searching diverse regions of the decision space. This wide exploration is instrumental in escaping local optima and identifying promising areas. As the algorithm progresses, a gradual transition from exploration to exploitation is observed, where SFO begins to focus its search around the most promising regions discovered. This balance is clearly maintained in most functions, where exploitation becomes dominant in the later stages of the optimization process. Such behavior contributes to both convergence reliability and solution refinement.

Notably, for functions F1–F30, SFO consistently maintains this adaptive trade-off, dynamically adjusting the exploration pressure while intensifying exploitation, particularly in unimodal and hybrid problem instances. This regulated transition is essential for improving convergence speed and achieving high-precision solutions, demonstrating SFO’s robustness and adaptability in handling complex optimization challenges.



**Fig. 6.** Search history, trajectory of the first agent, and average fitness of SFO optimizer over selected functions F6, F7, F8, F9 and F10 of IEEE CEC2017 functions.

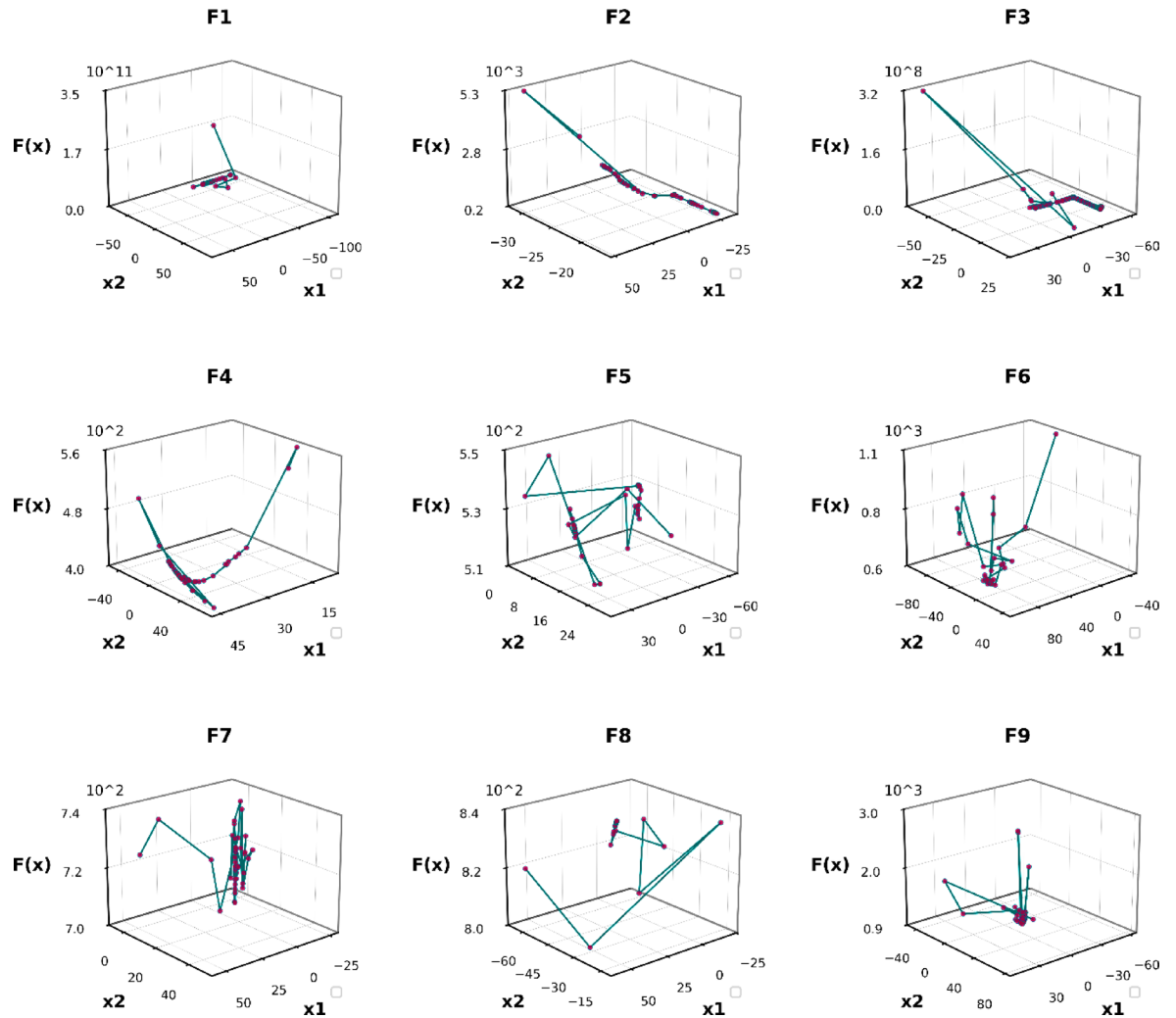
*Convergence analysis of SFO with competitive optimizers*

Figure 9 and 10 presents the convergence behavior of the competing algorithms on the IEEE CEC2017 benchmark functions for 10 dimensions. In Fig. 9 the results show that the proposed SFO algorithm frequently outperforms the other competitors in terms of convergence speed and final solution quality.

For the unimodal functions F1- F3 and F5, SFO achieves the fastest and most stable convergence, demonstrating strong exploitation ability and the capability to reach near-optimal solutions in significantly fewer iterations. In F4, SFO maintains competitive performance, performing on par with the best-performing competitors throughout the search process.

For the basic multimodal functions F6—F12, SFO consistently exhibits superior convergence for F6, F8, F10, and F11, while also delivering competitive performance on F7 and F9. In challenging cases such as F8 and F9, most algorithms (e.g., MFO, SCA, MVO) stagnate prematurely due to limited exploration, whereas SFO continues to make steady improvements over time. In F12, SFO achieves the best final solution, clearly outperforming all other algorithms tested.

In the hybrid and composite functions F13—F15, SFO demonstrates a strong balance between exploration and exploitation, achieving the best convergence in F13 and F14 and delivering top performance on F15. Across most functions, classical algorithms such as GWO, WOA, and PSO tend to stagnate early, while SFO sustains progress until the final iterations. These results highlight the robustness, adaptability, and effectiveness of the proposed SFO algorithm in solving diverse and complex optimization problems within the CEC2017 suite.



**Fig. 7.** 3D trajectories of the first particle of SFO on CEC2017 benchmark functions F1–F9.

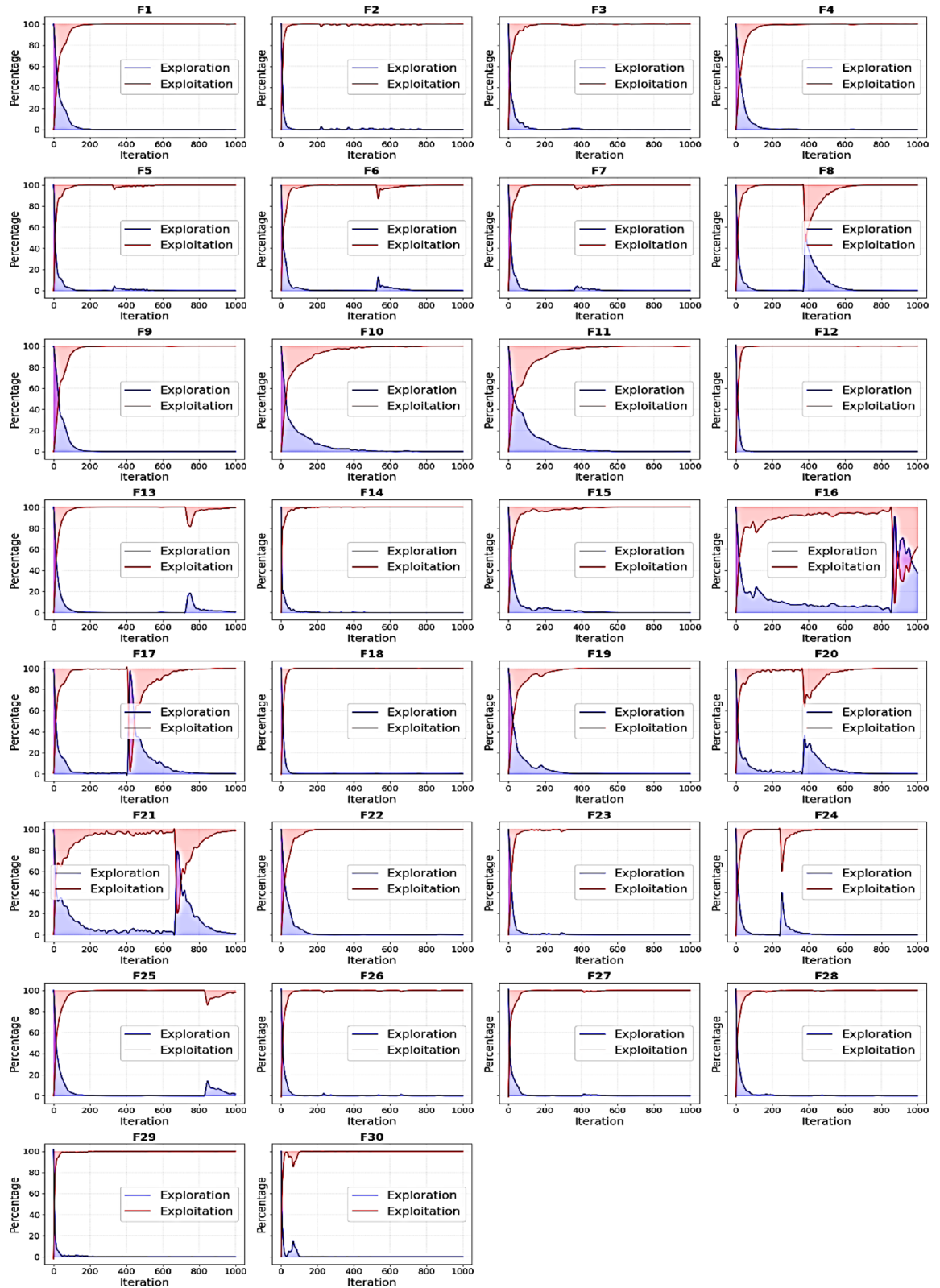
Figure 10 illustrates the convergence behavior of the competing algorithms across the F16 to F30 functions of the IEEE CEC2017 benchmark suite under 10-dimensional settings. the IEEE CEC2017 test set in 10-dimensional environments. Performance for our proposed algorithm SFO is variable in this set of test functions.

For several test functions such as F18, F20, and F23, competitive convergence is noted for SFO which often closely traces or beats conventional algorithms such as PSO, WOA, and GWO. Particularly for F20 and F23, convergence behavior for SFO is consistent with low variability and decent final fitness values are achieved. However, in functions F16, F17, and F24, for instance, the algorithm does worse relative to best schemes like EMBGO and PSO, which means there are issues for SFO in efficiently navigating rugged environments in certain hybrid or composition functions. Indeed, for F19, F21, F25, and F26, SFO converges at a slower rate and attains larger final fitness values, which means it needs improvement in its exploration mechanisms in order for it to handle heavily rugged or deceptive search environments. Despite these limitations, SFO exhibits moderate resilience and maintains competitive results in functions like F22, F27, and F28, where it performs on par with several baseline optimizers. On the most complex functions such as F29 and F30, SFO struggles to match the convergence quality of state-of-the-art algorithms such as EMBGO, GWO, and PSO, which display faster convergence rates and better solution precision.

#### *Analysis of box plot results*

The box-plot of all results and convergence curve of best result for SFO and MFO, SCA, MVO, HGSO, GTO, PSO, WAO and EMBGO on 30 benchmark functions of the IEEE CEC2017 with Dim = 10 throughout the iterations with 1,000 times are plotted in Figs. 11 and 12.

According to Fig. 11, the box-plot distributions illustrate the performance of the competing algorithms on the IEEE CEC2017 benchmark functions (F1–F15) with 10-dimensional search spaces. The proposed SFO algorithm demonstrates strong and stable performance across several functions, achieving the lowest median fitness and narrow interquartile ranges in functions such as F1, F2, F3, F4, and F11, indicating high robustness and accuracy. In other unimodal and basic multimodal functions, such as F5, F6, and F10, SFO maintains



**Fig. 8.** Exploration–exploitation ratio for SFO on CEC2017 suite with 10D.

competitive results, though some competitors, notably GTO and EMBGO, achieve slightly lower quartiles. For more challenging multimodal and hybrid functions (F7, F8, F9, and F12), SFO performs comparably to the top algorithms but exhibits a slightly wider spread in certain cases, reflecting higher variance in final solutions. In composite functions such as F13 and F15, SFO remains competitive, with performance close to the best

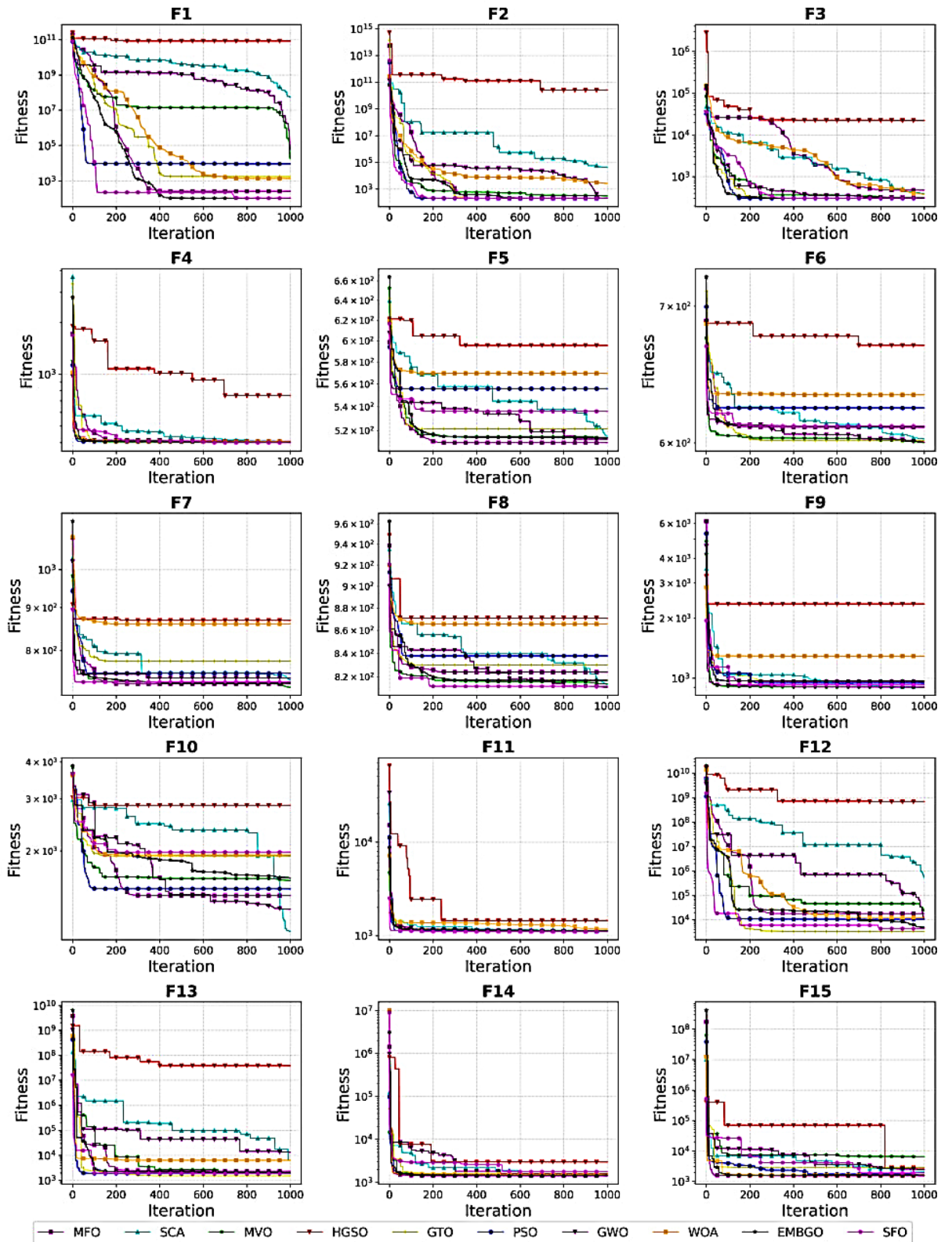
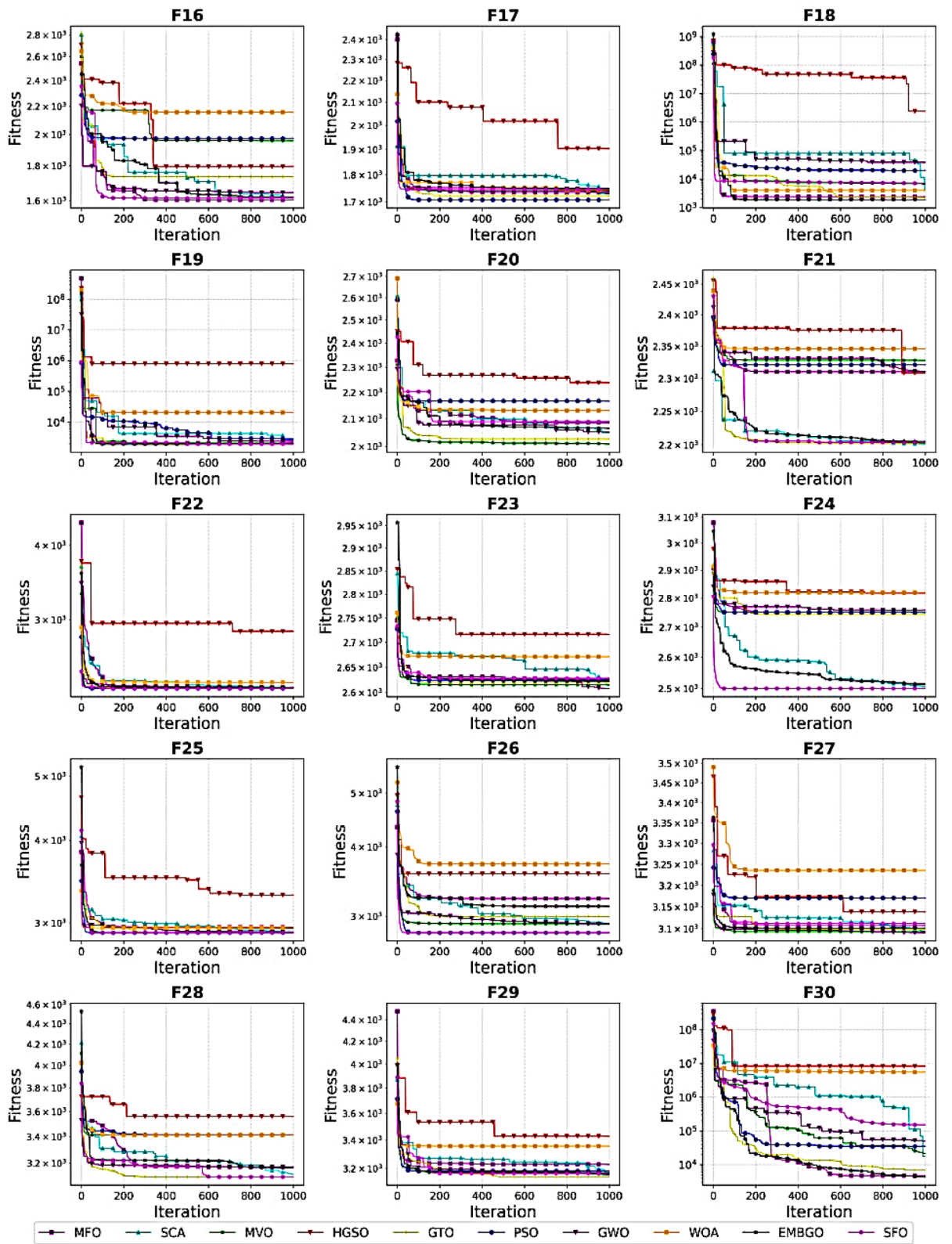


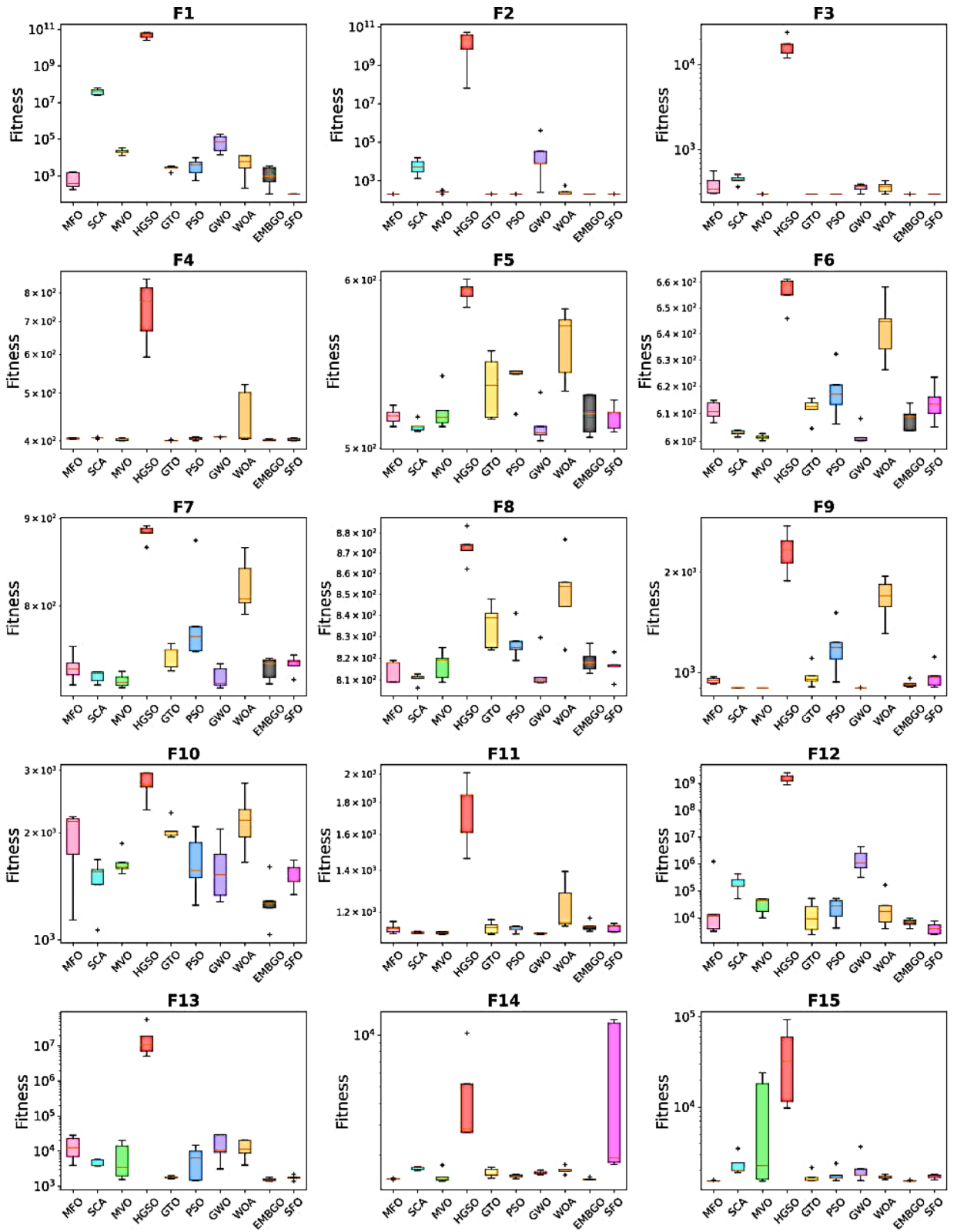
Fig. 9. Convergence curve analysis over CEC2017 benchmark functions (F1–F15), Dim = 10.

methods, although on F14 it shows a significantly larger quartile range and higher median, suggesting difficulty in fine-tuning on this function. Overall, the box-plot analysis reveals that SFO consistently ranks among the best performers on most CEC2017 functions, with particularly strong results in low-to-moderate complexity landscapes, while certain highly composite functions leave room for further improvement.



**Fig. 10.** Convergence curve analysis over CEC2017 benchmark functions (F16–F30), Dim = 10.

For the remaining functions F16–F30, Fig. 12 clearly shows the proposed SFO algorithm maintains competitive performance in several cases but also shows variability in challenging landscapes. In F16, F17, and F20, SFO achieves relatively low median fitness values, close to the best-performing algorithms, with a narrow spread indicating stable convergence. For functions F18 and F19, which exhibit high complexity, SFO



**Fig. 11.** Boxplots for SFO and other nine algorithms for IEEE CEC2017 benchmark functions(F1–F15).

delivers moderate performance—outperforming stagnating algorithms like MFO and GTO but trailing behind the top solutions from methods such as EMBO and WOA. In functions F21, F22, and F23, SFO demonstrates consistent behavior, ranking among the competitive algorithms, although the marginal gap to the leaders suggests room for further exploitation refinement. On hybrid and composite functions such as F24, F25, and

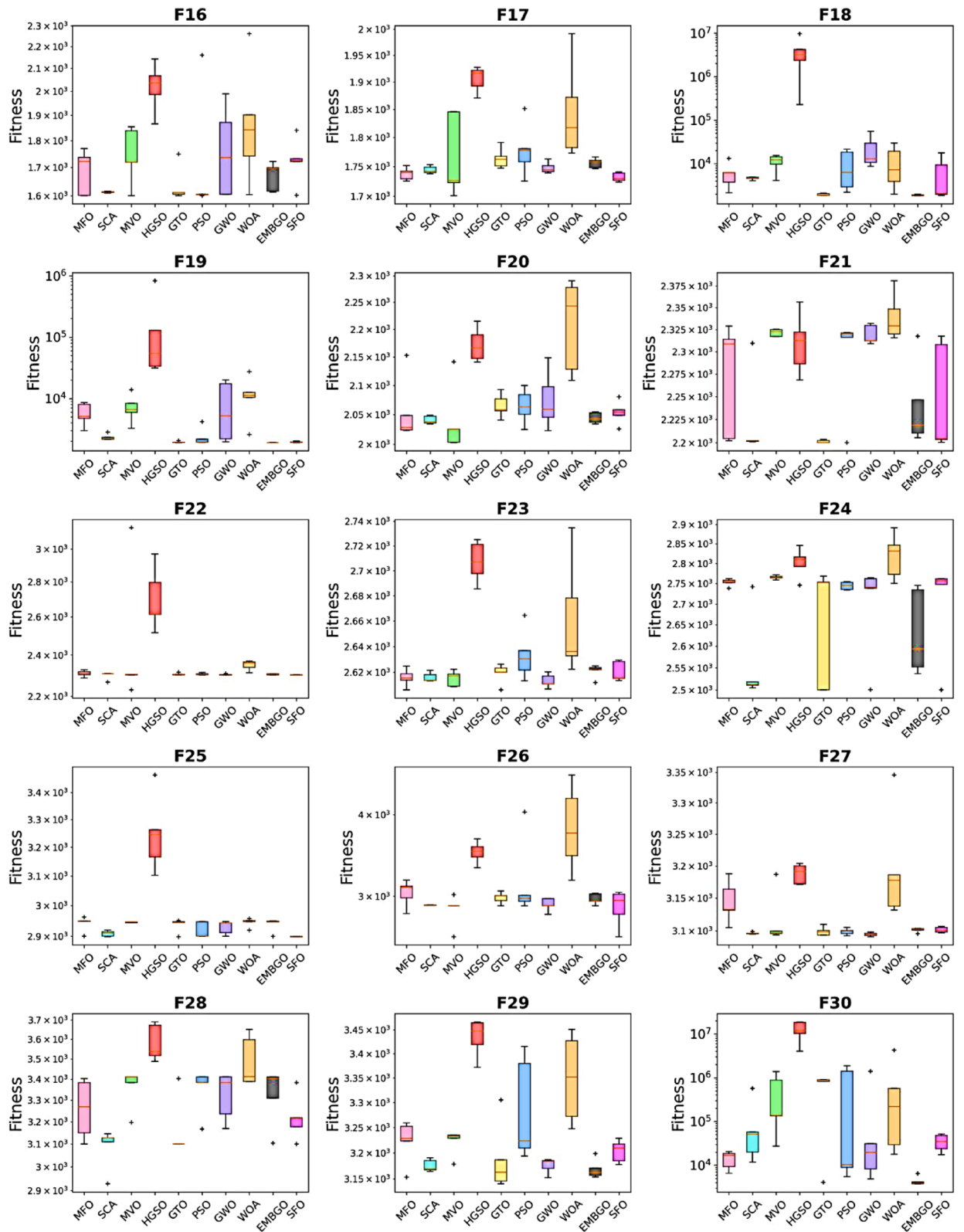


Fig. 12. Boxplots for SFO and other nine algorithms for IEEE CEC2017 benchmark functions(F16–F30).

F26, SFO produces solid median values, yet the quartile ranges indicate slightly higher variance compared to the best methods. Performance on F27–F29 is mixed: SFO remains competitive in F27, while in F28 and F29, several competitors achieve lower and more consistent fitness values. In the most complex composite function, F30, SFO delivers moderate results but is clearly outperformed by certain algorithms that achieve much smaller median fitness values.

## Discussion

The experimental results presented in Section “Results” demonstrate the competitiveness of the proposed Swift Flight Optimizer (SFO) across diverse benchmark functions. In this section, we provide a detailed discussion of these findings, highlighting the advantages and limitations of SFO compared to existing algorithms. We also analyze specific shortcomings to guide future research directions and outline the broader implications of adopting SFO in real-world optimization tasks. To elucidate the internal dynamics underlying these results, it is instructive to first examine how the algorithm regulates the balance between exploration and exploitation across different stages of the search. The adaptive three-mode structure of SFO regulates the transition from global exploration to local exploitation, with early iterations dominated by broad search movements that ensure extensive coverage and mitigate premature convergence, followed by increasingly directional updates that concentrate the population around promising basins, and final localized refinements that enhance solution accuracy. This staged progression maintains a dynamic balance between exploration and exploitation across runs. In some functions (e.g., F16, F21), oscillatory patterns appear in the exploration–exploitation curves due to the stagnation-aware reinitialization mechanism, which deliberately injects diversity when convergence stalls; although this intervention temporarily perturbs search dynamics and may distort fitness progression, it ultimately restores diversity and sustains long-term convergence. Beyond this exploration–exploitation interplay, the convergence trajectories provide additional evidence of how the adaptive design governs the optimizer’s long-term behavior. The convergence dynamics of SFO are characterized by a rapid initial descent, followed by a mid-phase of gradual improvement without stagnation, and a late stage of steady refinement rather than premature flatlining. This profile reflects the efficacy of the multi-mode switching strategy and the stagnation-aware reinitialization in sustaining optimization progress beyond the plateau stage common in classical swarms. The glide mechanism exerts a decisive influence on these dynamics: its parameterization governs the breadth of early exploratory moves, thereby controlling the algorithm’s capacity to access unexplored regions of the search space. Properly tuned glide parameters enhance basin discovery and delay stagnation, whereas suboptimal settings may restrict exploration and induce premature convergence. Consequently, glide plays a pivotal role in shaping the trajectory of convergence and in preserving the balance between global search and local refinement. The influence of these mechanisms becomes particularly evident when contrasting performance across varying problem dimensionalities, where structural challenges of the search space amplify the role of diversity management. Across different dimensional settings, SFO demonstrates consistent adaptability to the structural properties of the search space. In low- to moderate-dimensional problems, the glide phase efficiently identifies promising basins, the target phase consolidates improvements before stochastic noise dominates, and the micro-refinement operator exploits local curvature to achieve low terminal errors with limited variance across runs. These outcomes indicate that the control architecture generalizes effectively across unimodal, multimodal, and hybrid landscapes. In high-dimensional scenarios, where the exponential growth of the search space and heterogeneous curvature impose substantial challenges, SFO preserves early convergence speed and prevents premature population collapse. However, its relative advantage diminishes on ill-conditioned or highly composite functions, where directional updates may inadequately resolve narrow valleys without frequent diversity reintroduction. This suggests that scalability constraints are linked less to the intrinsic design of the search operators than to the frequency and efficiency of diversity-restoring mechanisms, which become increasingly critical as dimensionality escalates. Complementary insights are gained from sensitivity analysis, which clarifies the extent to which algorithmic performance depends on the calibration of key control parameters. The sensitivity analysis reveals that SFO’s performance is closely dependent on parameter calibration. Population size strongly affects convergence reliability, with larger populations enhancing diversity but incurring higher computational cost. The cognitive coefficient ( $\psi_1$ ) amplifies individual exploration, whereas the social coefficient ( $\psi_2$ ) stabilizes collective convergence. The inertia weight ( $w$ ) governs search dynamics, where higher values sustain exploration and lower values accelerate exploitation but risk stagnation. The stochastic scaling factors ( $\alpha, \beta$ ) regulate perturbation intensity, balancing basin discovery against local precision. These results indicate that robust performance arises from balanced parameterization rather than extreme settings, underscoring the importance of controlled exploration–exploitation trade-offs.

Synthesizing these observations, the principal strengths of the optimizer can be delineated as follows:

- Multi-mode balance* Glide, target, and micro phases jointly sustain exploration and exploitation.
- Diversity preservation* Stagnation-aware reseeding prevents premature convergence.
- Robust outcomes* Stable performance with low variance across independent runs.
- Wide applicability* Effective across diverse benchmark classes without heavy tuning.
- Efficiency* Rapid convergence with low computational overhead and scalability to high dimensions.

Conversely, certain constraints must be acknowledged to provide a balanced and critical evaluation of the method:

- Parameter sensitivity* Performance depends on glide and control coefficients, requiring careful calibration.
- Computational overhead* Diversity-restoring operations add cost, especially in high-dimensional settings.
- Scaling constraints* Effectiveness diminishes on ill-conditioned or highly composite functions.

**Switching complexity** The multi-mode framework relies on multiple transition conditions, which may complicate parameterization and reduce robustness across diverse problems.

Taken collectively, the findings substantiate the capacity of SFO to address fundamental challenges in global optimization, including the preservation of diversity, the mitigation of premature convergence, and the attainment of stable solutions across varied problem classes. At the same time, they identify methodological aspects such as parameter sensitivity, switching complexity, and scalability that require refinement to further enhance the algorithm's robustness and generalizability.

## Conclusions

This paper introduces a novel bio-inspired optimization algorithm known as the Swift Flight Optimizer (SFO), motivated by the adaptive flight behaviors of swifts in nature. By simulating glide, target, and micro flight modes, SFO keep a dynamic balance between exploration and exploitation, like the way swifts adapt their flight strategies for long-range travel and precise hunting maneuvers.

Experimental evaluation on the IEEE CEC2017 benchmark suite demonstrated the robustness and competitiveness of SFO. At 10 dimensions, SFO achieved the best average fitness on 21 out of 30 functions, and at 100 dimensions, it ranked first on 11 functions, consistently securing top positions across unimodal, multimodal, hybrid, and composition functions. These results confirm SFO's ability to achieve fast convergence, high-quality solutions, and stability in complex optimization landscapes. Compared to 13 state-of-the-art optimizers, including PSO, GWO, WOA, and EMBGO, SFO consistently outperformed or matched their results, underscoring its effectiveness and resilience in both low- and high-dimensional problems.

Key contributions of this work include the introduction of SFO as the first swift-inspired optimizer, the development of a multi-mode adaptive framework that balances exploration and exploitation, the incorporation of a stagnation-aware reinitialization mechanism to maintain diversity, and a comprehensive comparative evaluation that demonstrated superior convergence and solution quality against 13 state-of-the-art algorithms on the IEEE CEC2017 benchmark suite.

Despite these promising results, SFO has several limitations. Its performance remains sensitive to parameter settings, particularly those governing the transitions between exploration and exploitation phases, which may require careful tuning for different problem classes. Although the stagnation-aware reinitialization mechanism enhances diversity and reduces the risk of premature convergence, it also introduces additional computational overhead that may affect scalability when applied to very high-dimensional.

Building on these findings and limitations, future work on SFO will focus on expanding its applicability and enhancing its capabilities. A promising extension lies in multi-objective optimization, where SFO could manage simultaneous trade-offs among conflicting objectives, enabling broader use in complex real-world problems. Advances in adaptive parameter control and population diversity management may be achieved through hybridization with other metaheuristics or by incorporating machine learning models such as deep neural networks and reinforcement learning. For large-scale, high-dimensional optimization, distributed or parallelized SFO implementations may provide significant gains in scalability and computational efficiency. Finally, applying SFO to dynamic and time-varying optimization challenges—including UAV thermal search-and-rescue and industrial processes—offers an opportunity to demonstrate and enhance its real-time adaptability. These directions highlight both theoretical opportunities and pathways to broaden SFO's range of real-world scientific and engineering applications.

## Data availability

All relevant data are within the manuscript. The collection and analysis method complied with the terms and conditions for the source of the data.

Received: 30 August 2025; Accepted: 21 October 2025

Published online: 24 November 2025

## References

- Sharma, M. & Kaur, P. A comprehensive analysis of nature-inspired meta-heuristic techniques for feature selection problem. *Arch. Comput. Methods Eng.* **28**(3), 1103–1127. <https://doi.org/10.1007/s11831-020-09412-6> (2021).
- Wang, Z. & Schafer, B. C. Machine learning to set meta-heuristic specific parameters for high-level synthesis design space exploration. In *2020 57th ACM/IEEE Design Automation Conference (DAC)*, **20**(24) 1–6 (2020) <https://doi.org/10.1109/DAC18072.2020.9218674>.
- Dokeroglu, T., Sevinc, E., Kucukyilmaz, T. & Cosar, A. A survey on new generation metaheuristic algorithms. *Comput. Ind. Eng.* **137**, 106040. <https://doi.org/10.1016/j.cie.2019.106040> (2019).
- Vikhar, P. *Evolutionary algorithms: A critical review and its future prospects*. 261–265 (2016).
- Storn, R. & Price, K. Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces. *J. Glob. Optim.* **11**(4), 341–359. <https://doi.org/10.1023/A:1008202821328> (1997).
- Dagal, I. et al. Logarithmic mean optimization a metaheuristic algorithm for global and case specific energy optimization. *Sci. Rep.* **15**(1), 18155. <https://doi.org/10.1038/s41598-025-00594-2> (2025).
- Sotomayor, M., Pérez-Castrillo, D. & Castiglione, F. *Complex Social and Behavioral Systems Game Theory and Agent-Based Models: Game Theory and Agent-Based Models*. (2020).
- Wang, D., Tan, D. & Liu, L. Particle swarm optimization algorithm: An overview. *Soft Comput.* **22**(2), 387–408. <https://doi.org/10.1007/s00500-016-2474-6> (2018).
- Dorigo, M. & Stützle, T. (2010) Ant colony optimization: Overview and recent advances". In *Handbook of Metaheuristics* (eds Gendreau, M. & Potvin, J. Y.) 227–263 (Springer, 2010).
- Abdollahzadeh, B. et al. Puma optimizer (PO): A novel metaheuristic optimization algorithm and its application in machine learning. *Clust. Comput.* **27**(4), 5235–5283. <https://doi.org/10.1007/s10586-023-04221-5> (2024).

11. Trojovský P. & Dehghani, M. *Walrus Optimization Algorithm: A New Bio-Inspired Metaheuristic Algorithm*. (2022).
12. Rashedi, E., Rashedi, E. & Nezamabadi-pour, H. A comprehensive survey on gravitational search algorithm. *Swarm Evol. Comput.* **41**, 141–158. <https://doi.org/10.1016/j.swevo.2018.02.018> (2018).
13. Rodan, A., Al-Tamimi, A.-K., Al-Alnemr, L., Mirjalili, S. & Tiño, P. Enzyme action optimizer: A novel bio-inspired optimization algorithm. *J. Supercomput.* **81**(5), 686. <https://doi.org/10.1007/s11227-025-07052-w> (2025).
14. Karimzadeh Parizi, M., Keynia, F. & Khatibi Bardsiri, A. Woodpecker Mating Algorithm (WMA): A nature-inspired algorithm for solving optimization problems. *Int. J. Nonlinear Anal. Appl.* **11**(1), 137–157. <https://doi.org/10.22075/ijnaa.2020.4245> (2020).
15. Karimzadeh Parizi, M., Keynia, F. & Khatibi Bardsiri, A. OWMA: An improved self-regulatory woodpecker mating algorithm using opposition-based learning and allocation of local memory for solving optimization problems. *J. Intell. Fuzzy Syst.* **40**(1), 919–946. <https://doi.org/10.3233/JIFS-201075> (2021).
16. Gong, J. & Karimzadeh Parizi, M. GWMA: The parallel implementation of woodpecker mating algorithm on the GPU. *J. Chin. Inst. Eng.* **45**, 1–13. <https://doi.org/10.1080/02533839.2022.2078418> (2022).
17. Azizi, M. et al. Wolf-Bird Optimizer (WBO): A novel metaheuristic algorithm for building information modeling-based resource tradeoff. *J. Eng. Res.* **13**(2), 763–785. <https://doi.org/10.1016/j.jer.2023.11.024> (2025).
18. Alorf, A. A survey of recently developed metaheuristics and their comparative analysis. *Eng. Appl. Artif. Intell.* **117**, 105622. <https://doi.org/10.1016/j.engappai.2022.105622> (2023).
19. Houssein, E. H., Saeed, M. K., Hu, G. & Al-Sayed, M. M. Metaheuristics for solving global and engineering optimization problems: Review, applications, open issues and challenges. *Arch. Comput. Methods Eng.* **31**(8), 4485–4519. <https://doi.org/10.1007/s11831-024-10168-6> (2024).
20. Suresh, S. & Lal, D. S. Multilevel thresholding based on chaotic Darwinian particle swarm optimization for segmentation of satellite images. *Appl. Soft Comput.* <https://doi.org/10.1016/j.asoc.2017.02.005> (2017).
21. Rezk, H., Olabi, A. G., Mahmoud, M., Wilberforce, T. & Sayed, E. T. Metaheuristics and multi-criteria decision-making for renewable energy systems: Review, progress, bibliometric analysis, and contribution to the sustainable development pillars. *Ain Shams Eng. J.* **15**(8), 102883. <https://doi.org/10.1016/j.asej.2024.102883> (2024).
22. Niu, B., Wang, Y., Liu, J. & Yue, G.X.-G. Path planning for unmanned aerial vehicles in complex environment based on an improved continuous ant colony optimisation. *Comput. Electr. Eng.* **123**, 110034. <https://doi.org/10.1016/j.compeleceng.2024.110034> (2025).
23. Poudel, S., Arafat, M. Y. & Moh, S. Bio-inspired optimization-based path planning algorithms in unmanned aerial vehicles: A survey. *Sensors* <https://doi.org/10.3390/s23063051> (2023).
24. Mirjalili, S., Mirjalili, S. & Lewis, A. Grey wolf optimizer. *Adv. Eng. Softw.* **69**, 46–61. <https://doi.org/10.1016/j.advengsoft.2013.12.007> (2014).
25. Brest, J., Maučec, M. S. & Bošković, B. Single objective real-parameter optimization: Algorithm jSO in 2017. *IEEE Congr. Evol. Comput. (CEC)* **5–8**, 1311–1318. <https://doi.org/10.1109/CEC.2017.7969456> (2017).
26. Abd Elaziz, M., Al-qaness, M. A., Al-Betar, M. A. & Ewees, A. A. Polyp image segmentation based on improved planet optimization algorithm using reptile search algorithm. *Neural Comput. Appl.* **37**(8), 6327–6349 (2025).
27. Abdollahzadeh, B. et al. Puma optimizer (PO): A novel metaheuristic optimization algorithm and its application in machine learning. *Clust. Comput.* **27**(4), 5235–5283 (2024).
28. Alzoubi, S. et al. Synergistic swarm optimization algorithm. *Comput. Model. Eng. Sci.* **139**(3), 2557 (2024).
29. Zouache, D., Abualigah, L. & Boumaza, F. A guided epsilon-dominance arithmetic optimization algorithm for effective multi-objective optimization in engineering design problems. *Multimed. Tools Appl.* **83**(11), 31673–31700 (2024).
30. Abualigah, L. et al. The non-monopolize search (NO): A novel single-based local search optimization algorithm. *Neural Comput. Appl.* **36**(10), 5305–5332 (2024).
31. Al-Baik, O. et al. Pufferfish optimization algorithm: a new bio-inspired metaheuristic algorithm for solving optimization problems. *Biomimetics*. **9**(2), 65 (2024).
32. Abualigah, L. et al. Improved prairie dog optimization algorithm by dwarf mongoose optimization algorithm for optimization problems. *Multimed. Tools Appl.* **83**(11), 32613–32653 (2024).
33. Jia, H. et al. Modified crayfish optimization algorithm for solving multiple engineering application problems. *Artif. Intell. Rev.* **57**(5), 127 (2024).
34. Hamadneh, T., Kaabneh, K., Alssayed, O., Eguchi, K., Monrazeri, Z. & Dehghani M. Far and near optimization: A new simple and effective metaphor-less optimization algorithm for solving engineering applications. *CMES-Comput. Model. Eng. Sci.* **141** 2 (2024).
35. Hamadneh, T., Kaabneh, K., Alssayed, O., Bektemyssova, G., Shaikemelev, G., Umutkulov, D., Benmamoun, Z., Monrazeri, Z. & Dehghani M. Application of stork optimization algorithm for solving sustainable lot size optimization. *Comput. Mater. Continua*. **80** 2 (2024).
36. Abualigah, L., Abu-Dalhoum, E., Ikotun, A. M., Zitar, R. A., Alsoud, A. R., Khodadadi, N., Ezugwu, A. E., Hanandeh, E. S. & Jia, H. Teaching-learning-based optimization algorithm: analysis study and its application. *InMetaheuristic Optim. Algorithms*. Morgan Kaufmann 59–71 (2024).
37. Abualigah, L. et al. Particle swarm optimization algorithm: Review and applications. *Metaheuristic Optim. Algorithms* **1**, 1–4 (2024).
38. Hamadneh, T., Kaabneh, K., AlSayed, O., Bektemyssova, G., Montazeri, Z. & Dehghani M. Sculptor optimization algorithm: A new human-inspired metaheuristic algorithm for solving optimization problems. *Int. J. Intell. Eng. Syst.* **17** 4 (2024).
39. Abualigah, L., Al Turk, A. A., Ikotun, A. M., Zitar, R. A., Alsoud, A. R., Khodadadi, N., Hussien, A. G. & Jia H. Social spider optimization algorithm: survey and new applications. *InMetaheuristic Optim. Algorithms* Morgan Kaufmann 15–31 (2024).
40. Abualigah, L., Abualigah, R. A., Ikotun, A. M., Zitar, R. A., Alsoud, A. R., Khodadadi, N., Ezugwu, A. E., Hanandeh, E. S. & Jia, H. Whale optimization algorithm: analysis and full survey. *InMetaheuristic Optim. Algorithms*. 105–115 Morgan Kaufmann (2024).
41. Abualigah, L., Ahmad, E. N., Ikotun, A. M., Zitar, R. A., Alsoud, A. R., Khodadadi, N., Ezugwu, A. E. & Jia H. Animal migration optimization algorithm: novel optimizer, analysis, and applications. *InMetaheuristic Optim. Algorithms* 33–43 Morgan Kaufmann (2024).
42. Hedenström, A. Aerodynamics, evolution and ecology of avian flight. *Trends Ecol. Evol.* **17**, 415–422. [https://doi.org/10.1016/S0169-5347\(02\)02568-5](https://doi.org/10.1016/S0169-5347(02)02568-5) (2002).
43. Lentink, D. et al. How swifts control their glide performance with morphing wings. *Nature* **446**(7139), 1082–1085. <https://doi.org/10.1038/nature05733> (2007).
44. “Alpine Swift - *Apus melba*,” ed.
45. Nguyen, T., Nguyen, G. & Nguyen, B. M. EO-CNN: An enhanced CNN model trained by equilibrium optimization for traffic transportation prediction. *Procedia Comput. Sci.* **176**(800), 809. <https://doi.org/10.1016/j.procs.2020.09.075> (2020).
46. Mirjalili, S., Jangir, P. & Saremi, S. *Multi-objective ant lion optimizer: a multi-objective optimization algorithm for solving engineering problems*. (2016).
47. Whitley, D. A genetic algorithm tutorial. *Stat. Comput.* **4**(2), 65–85. <https://doi.org/10.1007/BF00175354> (1994).
48. Kennedy, J. & Eberhart, R. Particle swarm optimization. In *Proceedings of ICNN'95 - International Conference on Neural Networks*. **4** 1942–1948 (1995) <https://doi.org/10.1109/ICNN.1995.488968>.
49. Karaboga, D. & Basturk, B. On the performance of artificial bee colony (ABC) algorithm. *Appl. Soft Comput.* **8**(1), 687–697. <https://doi.org/10.1016/j.asoc.2007.05.007> (2008).
50. Mirjalili, S. & Lewis, A. The whale optimization algorithm. *Adv. Eng. Softw.* **95**, 51–67. <https://doi.org/10.1016/j.advengsoft.2016.01.008> (2016).

51. Chou, J.-S. & Truong, D.-N. A novel metaheuristic optimizer inspired by behavior of jellyfish in ocean. *Appl Math. Comput.* **389**, 125535. <https://doi.org/10.1016/j.amc.2020.125535> (2021).
52. Pan, Q., Tang, J., Zhan, J. & Li, H. Bacteria phototaxis optimizer. *Neural Comput. Appl.* **35**(18), 13433–13464. <https://doi.org/10.1007/s00521-023-08391-6> (2023).
53. Zhong, R., Xu, Y., Zhang, C. & Yu, J. Efficient multiplayer battle game optimizer for numerical optimization and adversarial robust neural architecture search. *Alex. Eng. J.* **113**, 150–168. <https://doi.org/10.1016/j.aej.2024.11.035> (2025).
54. Yan, J., Hu, G., Jia, H., Hussien, A. G. & Abualigah, L. GPSOM: group-based particle swarm optimization with multiple strategies for engineering applications. *J. Big Data* **12**(1), 114. <https://doi.org/10.1186/s40537-025-01140-7> (2025).
55. Bakır, H. A novel artificial hummingbird algorithm improved by natural survivor method. *Neural Comput. Appl.* **36**(27), 16873–16897. <https://doi.org/10.1007/s00521-024-09928-z> (2024).
56. Rizk-Allah, R. M., Slowik, A. & Hassanien, A. E. Hybridization of grey wolf optimizer and crow search algorithm based on dynamic fuzzy learning strategy for large-scale optimization. *IEEE Access* **8**, 161593–161611. <https://doi.org/10.1109/ACCESS.2020.3021693> (2020).
57. Si, T. et al. QSHO: Quantum spotted hyena optimizer for global optimization. *Artif. Intell. Rev.* **58**(3), 71. <https://doi.org/10.1007/s10462-024-11072-y> (2025).
58. Heidari, A. A. et al. Harris hawks optimization: Algorithm and applications. *Future Gener. Comput. Syst.* **97**, 849–872. <https://doi.org/10.1016/j.future.2019.02.028> (2019).
59. Akbari, M. A., Zare, M., Azizpanah-abarghoee, R., Mirjalili, S. & Deriche, M. The cheetah optimizer: A nature-inspired metaheuristic algorithm for large-scale optimization problems. *Sci. Rep.* **12**(1), 10953. <https://doi.org/10.1038/s41598-022-14338-z> (2022).
60. Xian, S. & Feng, X. Meerkat optimization algorithm: A new meta-heuristic optimization algorithm for solving constrained engineering problems. *Expert Syst. Appl.* **231**, 120482. <https://doi.org/10.1016/j.eswa.2023.120482> (2023).
61. Kirkpatrick, S., Gelatt, C. D. & Vecchi, M. P. Optimization by simulated annealing. *Science* **220**(4598), 671–680. <https://doi.org/10.1126/science.220.4598.671> (1983).
62. Rashedi, E., Nezamabadi-pour, H. & Saryzadi, S. GSA: A gravitational search algorithm. *Inf. Sci.* **179**(13), 2232–2248. <https://doi.org/10.1016/j.ins.2009.03.004> (2009).
63. Mirjalili, S., Mirjalili, S. M. & Hatamlou, A. Multi-verse optimizer: A nature-inspired algorithm for global optimization. *Neural Comput. Appl.* **27**(2), 495–513. <https://doi.org/10.1007/s00521-015-1870-7> (2016).
64. Hatamlou, A. Black hole: A new heuristic optimization approach for data clustering. *Inf. Sci.* **222**, 175–184. <https://doi.org/10.1016/j.ins.2012.08.023> (2013).
65. Zhao, W., Wang, L. & Zhang, Z. Atom search optimization and its application to solve a hydrogeologic parameter estimation problem. *Knowl. Based Syst.* **163**, 283–304. <https://doi.org/10.1016/j.knosys.2018.08.030> (2019).
66. Lam, A. Y. S. & Li, V. O. K. Chemical-reaction-inspired metaheuristic for optimization. *IEEE Trans. Evol. Comput.* **14**(3), 381–399. <https://doi.org/10.1109/TEVC.2009.2033580> (2010).
67. Wei, Z., Huang, C., Wang, X., Han, T. & Li, Y. Nuclear reaction optimization: A novel and powerful physics-based algorithm for global optimization. *IEEE Access* **7**, 66084–66109. <https://doi.org/10.1109/ACCESS.2019.2918406> (2019).
68. Al-Baik, O. et al. Pufferfish optimization algorithm: A new bio-inspired metaheuristic algorithm for solving optimization problems. *Biomimetics* <https://doi.org/10.3390/biomimetics9020065> (2024).
69. Wang, X. et al. Artificial protozoa optimizer (APO): A novel bio-inspired metaheuristic algorithm for engineering optimization. *Knowl. Based Syst.* **295**, 111737. <https://doi.org/10.1016/j.knosys.2024.111737> (2024).
70. Eskandar, H., Sadollah, A., Bahreininejad, A. & Hamdi, M. Water cycle algorithm—A novel metaheuristic optimization method for solving constrained engineering optimization problems. *Comput. Struct.* <https://doi.org/10.1016/j.compstruc.2012.07.010> (2012).
71. Deng, L. & Liu, S. Snow ablation optimizer: A novel metaheuristic technique for numerical optimization and engineering design. *Expert Syst. Appl.* **225**, 120069. <https://doi.org/10.1016/j.eswa.2023.120069> (2023).
72. Abualigah, L., Diabat, A., Mirjalili, S., Abd Elaziz, M. & Gandomi, A. H. The arithmetic Optimization Algorithm. *Comput. Methods Appl. Mech. Eng.* **376**, 113609. <https://doi.org/10.1016/j.cma.2020.113609> (2021).
73. Talatahari, S. & Azizi, M. Chaos game optimization: A novel metaheuristic algorithm. *Artif. Intell. Rev.* **54**(2), 917–1004. <https://doi.org/10.1007/s10462-020-09867-w> (2021).
74. Mirjalili, S. SCA: A sine cosine algorithm for solving optimization problems. *Knowl. Based Syst.* **96**, 120–133. <https://doi.org/10.1016/j.knosys.2015.12.022> (2016).
75. Trojovská, E. & Dehghani, M. A new human-based metaheuristic optimization method based on mimicking cooking training. *Sci. Rep.* **12**(1), 14861. <https://doi.org/10.1038/s41598-022-19313-2> (2022).
76. Reynolds, R. *An Introduction to Cultural Algorithms*. (1994).
77. Xiao, Y., Cui, H., Khurma, R. A. & Castillo, P. A. Artificial lemming algorithm: A novel bionic meta-heuristic technique for solving real-world engineering optimization problems. *Artif. Intell. Rev.* **58**(3), 84. <https://doi.org/10.1007/s10462-024-11023-7> (2025).
78. Deng, X., He, D. & Qu, L. A multi-strategy enhanced arithmetic optimization algorithm and its application in path planning of mobile robots. *Neural Process. Lett.* **56**(1), 18. <https://doi.org/10.1007/s11063-024-11467-6> (2024).
79. Xiao, Y.-N., Cui, H., Abu Khurma, R., Hussien, G. & Castillo, P. MCOA: A multistrategy collaborative enhanced crayfish optimization algorithm for engineering design and UAV path planning. *Int. J. Intell. Syst.* <https://doi.org/10.1155/int/5054424> (2025).
80. Li, G. et al. Review of the metaheuristic algorithms in applications: Visual analysis based on bibliometrics. *Expert Syst. Appl.* **255**, 124857. <https://doi.org/10.1016/j.eswa.2024.124857> (2024).
81. Mishra, A. & Goel, L. Metaheuristic algorithms in smart farming: An analytical survey. *IETE Tech. Rev.* **41**(1), 46–65. <https://doi.org/10.1080/02564602.2023.2219226> (2024).
82. Sharma, P. & Raju, S. Metaheuristic optimization algorithms: A comprehensive overview and classification of benchmark test functions. *Soft Comput.* **28**(4), 3123–3186. <https://doi.org/10.1007/s00500-023-09276-5> (2024).
83. Tomar, V., Bansal, M. & Singh, P. Metaheuristic algorithms for optimization: A brief review. *Eng. Proc.* <https://doi.org/10.3390/enproc2023059238> (2024).
84. Hedenström, A. et al. Annual 10-month aerial life phase in the common swift *apus apus*. *Curr. Biol.* **26**(22), 3066–3070. <https://doi.org/10.1016/j.cub.2016.09.014> (2016).
85. Hedrick, T. L., Pichot, C. & de Margerie, E. Gliding for a free lunch: Biomechanics of foraging flight in common swifts (*Apus apus*). *J. Exp. Biol.* <https://doi.org/10.1242/jeb.186270> (2018).
86. Hedenström, A. & Åkesson, S. Adaptive airspeed adjustment and compensation for wind drift in the common swift: Differences between day and night. *Anim. Behav.* **127**, 117–123. <https://doi.org/10.1016/j.anbehav.2017.03.010> (2017).
87. Wu, G., Mallipeddi, R. & Suganthan, P. *Problem Definitions and Evaluation Criteria for the CEC 2017 Competition and Special Session on Constrained Single Objective Real-Parameter Optimization*. (2016).
88. Peraza-Vazquez, H., Merino, M., Peña-Delgado, A., Morales-Cepeda, A. & Sinha, N. A novel metaheuristic inspired by horned lizard defense tactics. *Artif. Intell. Rev.* **57**, 1–65. <https://doi.org/10.1007/s10462-023-10653-7> (2024).
89. Zhao, S., Zhang, T., Ma, S. & Wang, M. Sea-horse optimizer: A novel nature-inspired meta-heuristic for global optimization problems. *Appl. Intell.* **53**(10), 11833–11860. <https://doi.org/10.1007/s10489-022-03994-3> (2023).
90. Braik, M., Hammouri, A., Atwan, J., Al-Betar, M. A. & Awadallah, M. A. White shark optimizer: A novel bio-inspired meta-heuristic algorithm for global optimization problems. *Knowl. Based Syst.* **243**, 108457. <https://doi.org/10.1016/j.knosys.2022.108457> (2022).

91. Abdollahzadeh, B., Soleimanian Gharehchopogh, F. & Mirjalili, S. Artificial gorilla troops optimizer: A new nature-inspired metaheuristic algorithm for global optimization problems. *Int. J. Intell. Syst.* **36**(10), 5887–5958. <https://doi.org/10.1002/int.22535> (2021).
92. Karami, H., Anaraki, M. V., Farzin, S. & Mirjalili, S. Flow direction algorithm (FDA): A novel optimization approach for solving optimization problems. *Comput. Ind. Eng.* **156**, 107224. <https://doi.org/10.1016/j.cie.2021.107224> (2021).
93. Hashim, F. A., Houssein, E. H., Mabrouk, M. S., Al-Atabany, W. & Mirjalili, S. Henry gas solubility optimization: A novel physics-based algorithm. *Future Gener. Comput. Syst.* **101**, 646–667. <https://doi.org/10.1016/j.future.2019.07.015> (2019).
94. Mirjalili, S. Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm. *Knowl. Based Syst.* **89**, 228–249. <https://doi.org/10.1016/j.knosys.2015.07.006> (2015).
95. Hussain, K., Salleh, M. N. M., Cheng, S. & Shi, Y. On the exploration and exploitation in popular swarm-based metaheuristic algorithms. *Neural Comput. Appl.* **31**(11), 7665–7683. <https://doi.org/10.1007/s00521-018-3592-0> (2019).
96. Cheng, S., Shi, Y., Qin, Q., Zhang, Q. & Bai, R. Population diversity maintenance in brain storm optimization algorithm. *J. Artif. Intell. Soft Comput. Res.* **4**(2), 83–97. <https://doi.org/10.1515/jaiscr-2015-0001> (2014).

### Author contributions

Abbas Aqeel Kareem, Ahmed Jabbar Abid: Conceptualization, Methodology, Software, Visualization, Investigation, Writing—Original draft preparation. Dalal Abdulmohsin Hammood, Salam J. Yaqoob: Data curation, Validation, Supervision, Resources, Writing—Review & Editing. Abdalrahman Husein, Viktoriia Berezhnychenko: Project administration, Supervision, Resources, Writing—Review & Editing.

### Declarations

#### Competing interests

The authors declare no competing interests.

#### Additional information

**Correspondence** and requests for materials should be addressed to V.B.

**Reprints and permissions information** is available at [www.nature.com/reprints](http://www.nature.com/reprints).

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

**Open Access** This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by-nc-nd/4.0/>.

© The Author(s) 2025