INFORMATICA, 2026, Vol. 0, No. 0, 1–40

DOI: https://doi.org/10.15388/26-INFOR635

Zebrafish Optimization Algorithm for Global Optimization Problems

Manickavasagam SURUTHI, Narayanan GANESH*

School of Computer Science and Engineering, Vellore Institute of Technology, Chennai 600127, Tamil Nadu, India
e-mail: ganesh.narayanan@vit.ac.in

Received: January 2026; accepted: June 2026

Abstract. A new population-based metaheuristic called Zebrafish Optimization Algorithm (ZFO) is proposed to find the global optimum solution using foraging behaviour of zebrafish larvae (Danio rerio). In ZFO, an exploration and exploitation trade-off is realized by simulation of exploratory dispersion, collaborative shoaling and directional searching. The movement strategy incorporates a stagnation-avoiding reinitialization mechanism, ensuring that diversity is maintained and premature convergence is avoided. To demonstrate the efficiency of ZFO, 23 benchmark functions including unimodal, multimodal and fixed dimension multimodal functions are tested and their optimal solutions are searched. Results obtained by ZFO are compared with existing metaheuristic algorithms, namely Cuttlefish Optimization, Jellyfish Search, Pufferfish Optimizer and Krill Herd algorithm. The Friedman rank test and Wilcoxon signed-rank test show that ZFO yields better-quality solutions than the compared algorithms in terms of convergence speed and robustness. The novel characteristic of ZFO is that it combines biologically inspired zebrafish dynamics and adaptive diversification strategy to enhance the capability of global search.

Key words: Zebrafish Optimization Algorithm, nature inspired metaheuristics, performance evaluation, global optimization, benchmark functions.

1. Introduction

Optimization is an integral part of any computational framework for engineering and sciences and it is essentially defined as the process of finding an optimal solution within a set of constraints for maximization or minimization of one or more objective functions (Osaba *et al.*, 2021). It finds applications in diverse fields such as signal and image processing, mechanical designing, manufacturing scheduling, chemical engineering and automatic control systems, etc. (Abualigah *et al.*, 2022). Real-world optimization problems possess the characteristics of high dimensionality, multimodal, non-linear and lack of gradient information so that the traditional deterministic methods cannot effectively solve these problems in practice (Pan *et al.*, 2023). These methods may become trapped in local optima, are based on convex assumptions and cannot be effectively and efficiently computed for NP-Hard problems classes (Zhan *et al.*, 2022; Gad, 2022).

*Correponding author.

Recently, metaheuristic algorithms have established themselves as a versatile and powerful paradigm that provides non-derivative, population-based search methods, approximating satisfactory solutions without explicitly relying on mathematical structure information of the problem (Beheshti and Shamsuddin, 2013). There are two classes of metaheuristics; the single solution based methods (S-metaheuristics), where an individual solution is successively refined with local search and the population based methods (P-metaheuristics), which use information collected from the population of candidate solutions (Talbi, 2009).

Metaheuristic algorithms are designed to solve problems with single solution and using multiple iterations to search for new improved solutions using local searches. S-metaheuristics include simulated annealing (SA) (Kirkpatrick *et al.*, 1983), Tabu search (TS) (Glover and Laguna, 1997), Hill climbing search (HC) (Sullivan and Jacobson, 2001), Iterated local search (ILS) (Lourenço *et al.*, 2003) and Variable neighbourhood search (VNS) (Mladenović and Hansen, 1997). These types of metaheuristic algorithms have been shown to perform quite well in terms of computational efficiency with fewer function evaluations and their localized search characteristics can also lead to problems with early convergence and local optima while solving very complicated multimodal or large-scale optimization problems.

In population-based approach, EAs mimic Darwinian principles of natural selection and genetic reproduction. In GA (Holland, 1992) and Differential Evolution (DE) (Price *et al.*, 2006), next-generation solutions are produced by various operators such as recombination, mutation and selection. Derivative versions such as BSA (Civicioglu, 2013) and Jaya Algorithm (JA) (Rao, 2016) have been developed to provide an equivalent level of performance.

Swarm Intelligence mimics the spontaneous emergent collective behaviour of biological communities. In Particle Swarm Optimization (PSO) (Kennedy and Eberhart, 1995), inspired by bird flocking, particles search for optimum based on their personal best location and global best location. While PSO has quick convergence, it might suffer from velocity explosion, trapping in local minima and premature convergence of the population near global optimum. Grey Wolf Optimizer (GWO) (Mirjalili *et al.*, 2014) uses a rigid dominance hierarchy (alpha, beta, delta and omega) for leading cooperative hunting in a pack, which is only optimal on unimodal functions, whereas the strong dominance restriction can negatively affect adaptivity on high-dimensional multimodal problems. Whale Optimization Algorithm (WOA) (Mirjalili and Lewis, 2016) imitates the bubbling and circling motion of humpback whales in hunting for prey. WOA has good exploitation, but is trapped in local optimum more often due to its strong exploitation property. Phasmatodea Population Evolution (PPE) (Song *et al.*, 2022) is based on the behaviour of stick insect for population diversity and its population movement strategy seems less systematic than fairness-driven movement strategy of ZFO.

Physics-based metaheuristics include the Gravitational Search Algorithm (GSA) (Rashedi *et al.*, 2009), which was inspired by Newton's law of gravitation acting on mass-weighted agents. Arithmetic Optimization Algorithm (AOA) (Abualigah *et al.*, 2021) uses the distribution law of the arithmetic operation to specify the rules. The Fick's Law Optimization Algorithm (Hashim *et al.*, 2023) was inspired by molecular diffusion. Although they have analytically based update rules, they do not exhibit the adaptive behavioural flexibility that biological agents demonstrate during foraging the key motivation for ZFO.

Human-based metaheuristics can be seen as socio-cognitive events abstract to optimization operators. For example, Teaching-Learning-Based Optimization (TLBO) (Rao *et al.*, 2011) abstracts the learning between a teacher and pupils into operators. Supply-Demand-Based Optimization (SDO) (Zhao *et al.*, 2019) relies on economic theory for modelling moving candidates through the use of market principles. These algorithms are fast but not composed of distinct behavioural stages nor is the interaction and spatial coordination governed by processes that ZFO adopted from neurobiology of zebrafish.

Aquatic organism-inspired algorithms comprise a rapidly expanding sub-family of Swarm Intelligence. The Cuttlefish Optimization Algorithm (CFO) (Kalpanarani and Hannah Grace, 2025) aims to simulate the chameleon-like ability of cuttlefish to vary their reflection and visibility using chromatophores. CFO uses skin-pattern mimicry as its search operator, restricting its capacity for the type of group foraging-guided search, which drives the exploration of ZFO. The Jellyfish Search Optimizer (JSO) (Chou and Truong, 2021) mimics passive ocean-current induced motion and the interaction of the swarm with the food near a patch. However, as jellyfish possess no active sensory-guided locomotion, their motion update is based upon current driven stochastic perturbation rather than the ZFO form of actively track prey directed locomotion. The Pufferfish Optimization Algorithm (POA) (Al-Baik *et al.*, 2024) focuses upon both pufferfish puffing defensive action and on the preservation of population diversity. Rather than ZFO's coordination strategies in a group hunt, the preservation of diversity in POA is through avoidance of predation; group coordinated hunting does not play any part in this search strategy and its adaptation towards and away from the ZFO hunting state does not arise. The Krill Herd Optimizer (KHO) (Gandomi and Alavi, 2012) simulates movement in a herd of Antarctic krill driven by three movement forms, neighbour-induced motion, foraging motion and physical diffusion. The model of krill foraging is again based on diffusively random walks rather than ZFO's fairness weighted directional movement toward prey.

In the recent years, metaheuristics have also been explored for solving global optimization problems over a wide range of applications. Firefly Algorithm and Ant Colony Optimization was proposed as hybrid metaheuristics, such as Colliding Bodies Optimization, for estimating the GMDH parameters used to predict software testability (Wu, 2026). Ant colony optimization algorithm is adapted for supply chain network design where the global search ability of the ant colony optimization is utilized to minimize the total costs and increase the service level (Huang, 2025a). A Hybrid particle swarm optimization has been presented based on Q-learning framework for the problems of airport parking space assignment and scheduling (Huang, 2025b). A multi-objective metaheuristics algorithm is introduced to implement energy efficient virtual machine placement in the cloud data centres (Vijaya and Srinivasan, 2024).

Despite substantial advances in metaheuristic optimization, the problem of determining an intelligent, adaptive trade-off between the global search and local exploitation of the metaheuristics without requiring excessive tuning has not been sufficiently answered. The No Free Lunch (NFL) theorem theoretically demonstrates that one algorithm is not better than all other algorithms in terms of any class of optimization problem, thus motivating the development of novel algorithms for optimization problems of specific structural kinds. Specifically, biologically validated multi-phase behaviours inspired by nature remain largely underexplored, in particular for continuous multimodal test functions, which often expose premature convergence and stagnation problems of standard population-based algorithms. Nevertheless, most studies concentrate on hybrid algorithm and parameter setting strategy with few concerns about bio-inspiration adaptive mechanism and intrinsic mechanism to escape local optima. This work tackles the issues mentioned above by presenting the Zebrafish Optimization Algorithm (ZFO), a novel population-based approach with autonomous interaction, fairness-weighted exploration and adaptive stagnation recovery strategy to achieve a powerful global search performance.

The present paper addresses this gap by introducing the Zebrafish Optimization Algorithm (ZFO), a novel population-based metaheuristic based on the neuro-collective behaviours during the foraging activities of larval zebrafish. Contrary to organisms inspiring the following other biological-based population algorithms PSO (birds), GWO (wolves), WOA (whales), CFO (cuttlefish) JSO (jellyfish), POA (pufferfish) and KHO (krill), zebrafish exhibits sophisticated, sensory-driven collective predatory behaviour which is characterized by clear-cut phase transitions between three biologically validated foraging behaviours: scattering-based movement toward areas of high information richness (dispersive search), fish schools that follow a predator and rapidly converge to corner prey (shoaling and directed prey-targeting) and group-hunting at a very high speed. ZFO embeds these dynamic, phased transitions as phased position-updating rules and additionally incorporates a fairness-governed update operator designed to give more aggressive search emphasis to low-fitness members and a detection mechanism that reinitializes a portion of the population to prevent convergence to optima from becoming stagnated. The following are the main contributions:

I. The novel three-phase zebrafish-inspired optimization framework based on the mathematical formulation of three observed fish school group foraging activities: exploration, shoaling and hunting behaviours.
II. An adaptive updating mechanism using fairness weight to dynamically balance exploration and exploitation while maintaining population diversity is proposed.
III. The incorporation of stagnation-aware detection and reinitialization mechanism to avoid premature convergence and robust behaviour for complicated search space.
IV. A computational framework and computational complexity are proposed.
V. Extensive simulation studies using 23 benchmark functions are implemented. Friedman rank and Wilcoxon signed-rank statistical test are carried out and compared to other four new aquatic-organism-inspired optimizers, CFO, JSO, POA and KHO.

2. Zebrafish Optimization (ZFO)

The proposed Zebrafish Optimization (ZFO) and how it was inspired by juvenile zebrafish. First, the biological characteristics that led to the design of ZFO, including the identification and analysis of some key foraging behaviours exhibited by zebrafish. These factors provide the basis for both the behavioural modelling and mathematical formulation described later in subsequent section.

2.1. *Inspiration*

The investigation of biological systems reveals that many groups of animals can create very effective methods of problem-solving using group intelligence. Aquatic species such as zebrafish (*Danio rerio*) demonstrate coordinated behaviours to develop effective foraging strategies for foraging of food through the development of cooperative behaviours. Despite the comparatively simple brain structure, larval zebrafish have evolved into extremely sophisticated hunt-and-catch mechanisms with regards to their prey. Their foraging mechanism is random, cooperative and specifically targets certain types of food.

Larval zebrafish are small, freshwater organisms that are being used extensively in the study of behavioural biology and neuroscience because they have transparent bodies and grow quickly, plus have thoroughly characterized sensorimotor responses. Larval zebrafish are typically found in shallow freshwater habitats such as shallow ponds and streams. Although food resources may be concentrated within a water body, their availability varies across locations and time. To sustain themselves in this type of environment, larval zebrafish must employ an efficient foraging strategy, which utilizes sensory perception, motion control and the ability to interact with others.

Biological research has demonstrated that freely swimming larval zebrafish will utilize an array of different strategies to retrieve their prey instead of one fixed strategy. As such, larval zebrafish's foraging behaviour is comprised of three varying and interchanging phases of behaviour, exploratory, collective shoaling and focused hunting. This phase-switching occurs in an adaptive manner, due to prey availability, the specific environmental circumstances and the degree of individual success of the zebrafish's foraging. Therefore, larval zebrafish allow for groups of larval zebrafish to achieve high hunting efficiency while remaining as a cohesive and equitable group.

The similarity between zebrafish in their foraging structure represents an egalitarian characteristic. Zebrafish have a shared set of rules, where weaker and less successful fish may have more freedom to move and elicit opportunities to forage. More successful (stronger) zebrafish will tend to limit the amount of aggressive pursuit that they engage in with less successful zebrafish. The presence of biological fairness across zebrafish populations prevents the monopolization of food resources and facilitates the performance of an entire group of zebrafish over extended periods, thus effectively making zebrafish much more suitable models for simulating population-based optimization processes.

Zebrafish also represent a natural form of resiliency to stagnation. If, after repeated attempts to capture prey, an area does not yield any food, zebrafish leave that area and return to exploring. This form of adaptive resetting of behaviour prevents prolonged unproductive behaviour and enables zebrafish to escape rapidly from areas that do not yield successful catches and continue searching for more productive areas to forage.

Inspired by the biological observations that are related to zebrafish behaviour, the Zebrafish Optimization (ZFO) is a computational simulation of the foraging intelligence of zebrafish that integrates exploration, search/schooling and hunting behaviour into one cohesive framework. Rather than relying on global adaptation alone, ZFO incorporates a fairness-driven adaptive mechanism and a stagnation escape providing a more effective balance between global exploration and local exploitation when applied to complex optimization landscapes.

2.2. *Zebrafish Behaviour*

2.2.1. *Exploration Behaviour*

In typical schooling fish including sardines, tuna and herrings, individuals show highly synchronised collective movement behaviours driven by a uniform search process, in which the environment is explored using a predominantly similar set of rules with little dependence on individual foraging experiences. By contrast, larval zebrafish exhibit an individually responsive and autonomous exploratory strategy, where individual larvae explore unknown regions using their own visual and mechanosensory systems rather than fixed group formations or coordinated leadership structures. Additionally, movement magnitude is adaptively controlled in accordance with individual foraging performance, such that a low-performing individual is afforded greater exploratory freedom through a fairness-weighted behavioural adjustment. This unique biological mechanism forms the foundation of the individual foraging-success-based exploration strategy in the proposed Zebrafish Optimization Algorithm (ZFO) and consequently contributes to enhancing population diversity, balancing the exploration-exploitation process and improving resistance to premature convergence.

In ZFO, exploration behaviour refers to the manner in which zebrafish conduct their random foraging movements when they are finding food sources. During this stage of learning, an agent will move in a random or arbitrary fashion to discover new areas of the search space for food.

ZFO's method of regulating how much exploration takes place during each learning iteration involves the implementation of a time-dependent or adaptive step size. In the ZFO algorithm, the initial iterations use a larger-than-normal step size and gradually reduce the amount of exploration as optimization continues.

The second aspect of regulating exploration intensity is its incorporation of a fitness-dependent fairness factor. As a result of the incorporation of the fitness-dependent fairness factor, agents with the worst fitness (i.e. those with the least success) can take larger exploratory actions, thereby allowing them to escape from areas where they are unsuccessful and re-enter the competition for food. On the other hand, well-performing agents will engage in more conservative actions to avoid dominating the exploratory actions of the remaining agents.

2.2.2. *Shoaling Behaviour*

The significant distinction between shoaling zebrafish and other standard forms is that spatial regrouping adjusts dynamically to local conditions. The dynamic spatial organization of zebrafish shoal's results in constant local rearrangement within the group as each individual reacts to neighbour attraction and communal chemical cue detection. This process constitutes environmentally induced, information based dynamic modulation of spatial positioning rather than predetermined locomotion and random drift is the underlying basis for the neighbourhood centroid shoaling mechanic of ZFO and guarantees that the population does not become uniform or otherwise cease to be effective throughout the course of the search.

The shoaling behaviour of zebrafish is the coordinated swimming patterns with keeping safe distances between shoal members through the collective behaviour of zebrafish. In ZFO, every agent interacts with its local neighbourhood based only on the spatial proximity to other agents, not on any global information. When determining how to move, an agent will move towards the centroid of its neighbourhood to facilitate information sharing among nearby agents.

A separation mechanism is incorporated to prevent agents from colliding with one another when they become close together. An agent will be repelled when it comes too close to a neighbouring agent. Furthermore, perturbations are added to allow for continued diversity while not disrupting the overall trend of convergence. The shoaling behaviour acts as an intermediary stage in stabilizing population dynamics and allows for the gradual improvement of the most promising areas.

2.2.3. *Hunting Behaviour*

This ZFO attack technique is derived from the leaderless, synchronous predatory behaviour of larval zebrafish and distinct from many hierarchical predator models and conceptually dissimilar to hierarchically based search attack strategies of existing metaheuristic algorithms. Other predators typically coordinate attacks in a predetermined social structure or by a defined leader whose task assignments are known beforehand but larval zebrafish do not utilize a hierarchy and instead coordinate attack on the prey based on a dynamic interaction between their current distance to prey and individual search potential at that particular instant. This biologically uncommon, fitness-based, interaction-driven coordination serves as the foundation of the fitness-modulated with multi-agent exploitation method implemented in ZFO, where the population concentrates its search on potentially high-fitness regions while ensuring distribution and resistance to premature convergence.

Zebrafish exhibit a shift in their hunting behaviour once they detect a potential prey item. In this stage, they exhibit rapid and directed motion toward it. During this step, zebrafish are able to adapt their swimming paths to enhance their chance for a successful capture, often through the combination of sensory input from themselves and members of their immediate shoal.

Candidate solutions actively search prospective regions during the exploitation phase of the proposed optimiser, which is equivalent to the hunting activity mentioned in promising candidate solutions intensively and concurrently while pursuing both global and local optima.

2.2.4. *Stagnation Avoidance Behaviour*

The ZFO stagnation recovery mechanism draws inspiration from a biologically unusual mode of adaptation observed in larval zebrafish. While many predator-inspired models do not explicitly model after a bout of searching failures, larval zebrafish make an individual decision and, once they accumulate a specific number of consecutive search failures, immediately leave that poor patch and redisperse back into unexplored space. The intelligent adaptive reset forms the basis of ZFO stagnation recovery, the stagnant agents are triggered for reactivation in order to restore diversity, get out of the local optimum and improve the exploration of several potential basins.

Zebrafish show an important trait of fairness-driven adaptation in zebrafish shoals. In a shoal, less-successful individuals have more space for the possible recovery from unfavourable areas, while successful individuals adopt a more conservative strategy. At the same time, zebrafish will leave areas that do not yield any successful hunting opportunities and will become exploratory and continue searching for better alternatives.

These biological mechanisms motivate the integration of two key components to create ZFO, which includes fitness-based adaptation and the ability to switch between behaviours when they become stagnant, thereby improving the robustness of the system and improving long-term search efficiency.

2.3. *Mathematical Modelling of ZFO*

The optimization of ZFO starts by generating a population of candidate zebrafish (X) according to (1). This candidate population is produced in a random manner between the determined lower bounds lb and upper bounds ub of the specific optimization problems.

$$z = \begin{bmatrix} z_{1,1} & z_{1,2} & \cdots & z_{1,d} \\ z_{2,1} & z_{2,2} & \cdots & z_{2,d} \\ \vdots & \vdots & \ddots & \vdots \\ z_{n,1} & z_{n,2} & \cdots & z_{n,d} \end{bmatrix}, \tag{1}$$

where Z is the set of candidate zebrafish solutions to date, with $z_{i,j}$ indicating the location of the decision variable j for the zebrafish solution i. The population size is denoted by n, while the problem dimensionality is stated as d.

Every zebrafish's initial value is defined as a randomly generated number from the uniform distribution as expressed:

$$z_{i,j} = lb_j + \text{rand} \cdot (ub_j - lb_j). \tag{2}$$

In (2) rand is a randomly generated number obtained from the uniform distribution within the range of [0, 1], with lb_j representing the lower limit and ub_j denoting the upper limit of decision variable j.

At each iteration, the best solution found so far is defined as the zebrafish solution whose associated fitness value is the smallest. The best solution becomes the global best and will guide the fish in hunting and shoaling in the next iteration.

2.3.1. *Exploration Phase*

The earlier stages of optimization for zebrafish are primarily characterized by exploratory responses. In the absence of known food sources, the exploratory response is analogous to a random search for potential locations of food. During this phase, an agent's movements will be random throughout the search space to attempt to find areas to discover unexplored regions.

The update rule for position on exploration, for the ith zebrafish at time (t), is represented by:

$$Z_i^{t+1} = Z_i^t + \alpha(t)(1 + \phi_i)R, \tag{3}$$

where R follows a Gaussian distribution $N(0, 1)$ in (3) and (4) $\alpha(t)$ is representation of time-varying exploration factor, defined by:

$$\alpha(t) = \alpha_{\text{max}}(1 - \tau) + \alpha_{\text{min}}\tau, \quad \tau = \frac{t}{T}, \tag{4}$$

where $\tau \in [0, 1]$ is the normalized step of iteration progress (value from 0 to 1), t is the current iteration, T is the maximum number of iterations, α_{max} and α_{min} are the upper and lower bounds of the exploration step size respectively.

In (5) represents term ϕ_i called the fairness factor, which encourages greater activity in poorly performing candidate solutions, is defined as:

$$\phi_i = \max\left(0, \frac{F_{\text{median}} - F_{\text{best}} + \varepsilon}{F_i - F_{\text{best}}}\right). \tag{5}$$

This exploration method reduces the risk of premature convergence, as well as increasing the area coverage of the search space. Where F_{median} is the median fitness value of the population, $F_{\text{best}} + \varepsilon$ is the best fitness value encountered so far and is a tiny positive value 10^{-10} to avoid division by zero. The operator $\max(0, \cdot)$ ensures that is ϕ_i non-negative.

2.3.2. *Shoaling Behaviour*

Zebrafish exhibit shoaling as a mechanism for social regrouping. Shoaling allows individuals to swim together while keeping safe distances apart from one another. Through shoaling behaviour, information can be shared between neighbouring zebrafish. It also allows the zebrafish to continue to move together as they migrate from one location to the next.

The local centre associated with the ith zebrafish is calculated using (6):

$$C_i = \frac{1}{k} \sum_{j \in N_i} X_j. \tag{6}$$

Zebrafish update their position according to:

$$X_i^{t+1} = X_i^t + \beta(1 + 0.2\phi_i)\big(C_i - X_i^t\big) - \gamma S_i + \delta_s R. \tag{7}$$

As given in (7), $S_i = X_i^t - X_{\text{near},i}$, where k is the size of neighbourhood, N_{i} is the set of the k nearest neighbours of agent i and C_i is the local shoal centroid of agent i and X is the separation vector from nearest neighbour of the agent i to avoid congestion, β is the weight of shoaling attraction, γ is the weight of repulsion (separation), δ_s is the weight of random perturbation and R is a vector of random numbers sampled from $N(0, 1)$. S_i prevents zebrafish from clumping on each other β, γ and δ_s are parameters used to define attraction, repulsion and random perturbation weight, respectively. Shoaling behaviour reflects the shift from exploration or exploitation behaviour to cohesion behaviour of the swarm.

2.3.3. *Hunting Phase*

After identifying areas with significant prey abundance, zebrafish begin to hunt by moving towards those areas in a directed manner. During this hunting process, zebrafish can use both local and global information to help improve their potential prey location.

$D_g = Z_{\text{best}} - Z_i$ is the global direction vector in (8) toward population best position Z_{best} and $D_l = Z_{\text{lbest}} - Z_i$ is the local direction vector toward local best position.

$$D_g = Z_{\text{best}} - Z_i, \qquad D_l = Z_{\text{lbest}} - Z_i. \tag{8}$$

These two hunting directions will be combined to form the hunting direction:

$$D_h = w_g D_g + (1 - w_g) D_l. \tag{9}$$

As given in (9), D_h represents the holistic hunting direction and w is the adaptive global influence weight within range $w_g[0, 1]$.

The global influence weight w_g is defined in (10), where $\tau = \frac{t}{T}$ is the normalised iteration progress. When $\tau = 0$ (early iterations), $w_g = 0.5$ global and local guidance contribute equally to D_h, where $D_h = 0.5D_g + 0.5D_l$, maintaining balance between local and global search. As $\tau = 1$ (late iterations), $w_g = 1.0$, means $D_h = w_g D_g + (1 - w_g) D_l$. $D_l = 1.0D_g + 0D_l = D_g$. And hence D_g, the direction is then dictated totally by the global best. If $\tau = 0$ (early iterations), then $w_g = 0.5$ and so the global and local directions equally contribute to the initial exploration:

$$w_g = 0.5 + 0.5\frac{t}{T}. \tag{10}$$

The new hunting direction of zebrafish will be updated according to (11), where η is the hunting step scale factor, ξ is the hunting perturbation weight and R is a random vector drawn from $N(0, 1)$. The term $(1 - \tau)$ makes the hunting step get smaller with each iteration, enabling fine-grained exploitation at later iterations:

$$Z_i^{t+1} = Z_i^t + \eta(1 + \phi_i)(1 - \tau) D_h + \xi R. \tag{11}$$

This feature enables zebrafish to exploit promising regions more thoroughly in later iterations while retaining mobility for lower-fitness individuals.

2.3.4. *Stagnation-Aware Behavioural Adaptation*

The natural behaviour of zebrafish also exhibits a periodicity in their hunting patterns. ZFO implements a mechanism that forces a zebra to reperform line searches on areas that are deemed to have experienced more than a fixed number of unsuccessful hunting attempts.

$$Z_i^{t+1} = LB + R \odot (UB - LB), \tag{12}$$

where LB and UB are the lower and upper bound vectors of the search space in (2), $R \sim U(0, 1)$ is a vector of uniformly distributed random numbers and $\odot$ denotes element-wise (Hadamard) multiplication. In (12) presents the vectorized form of initialization. This type of adaptive reset allows the algorithm the opportunity to recover from getting stuck in a local solution (e.g. local search) and continue its effective long-term search. An adaptive stagnation threshold $S_{th} = \max(3, \text{round}(0.05 \times T))$, where T is the number of iterations. It refers to the number of unsuccessful search steps allowed consecutively before reinitialization. It is scaled dynamically according to optimization horizon and bounded from below by 3. This condition for reset is designed such that it triggers early enough for detecting premature stagnation in short time periods and sustained enough for detection of stagnation in long time periods before reset.

2.3.5. *Behavioural Transition Strategy*

Exploration, shoaling and hunting are selected according to a probability-related method which relates to the interdependence of an individual agent's fitness at a specific time and the agent's progress through the optimization process:

$$P_{\text{explore}} = 1 - \tau + 0.5\phi_i, \tag{13}$$

$$P_{\text{hunt}} = \tau + 0.3\phi_i, \tag{14}$$

$$P_{\text{shoal}} = 1 - (P_{\text{explore}} + P_{\text{hunt}}), \tag{15}$$

where $\tau = \frac{t}{T}$ is the normalised iteration progress and φ_{i} is the individual fairness value defined by (13), (14) and (15). P_{explore} are the respective probabilities for entry into exploration, P_{hunt} is the hunting and remaining shoaling state P_{shoal}, which is normalized to unity before making state selection. The value of P_{explore} is monotonically decreasing with progress, which implies that later iterations favour the exploration state. On the other hand, P_{hunt} increases with both τ and φ_{i}, so the high-fitness agent in later iteration has the biggest probability of directed hunting. During each iteration, the above three states probabilities are separately sampled by every agent to make the behaviour choices and generate a heterogeneous population that exploration, shoaling and hunting exist at the same time among the swarm. This individual-level adaptation strategy of phase-switching provides ZFO to maintain an everlasting compromise between global search and local exploitation during the process.

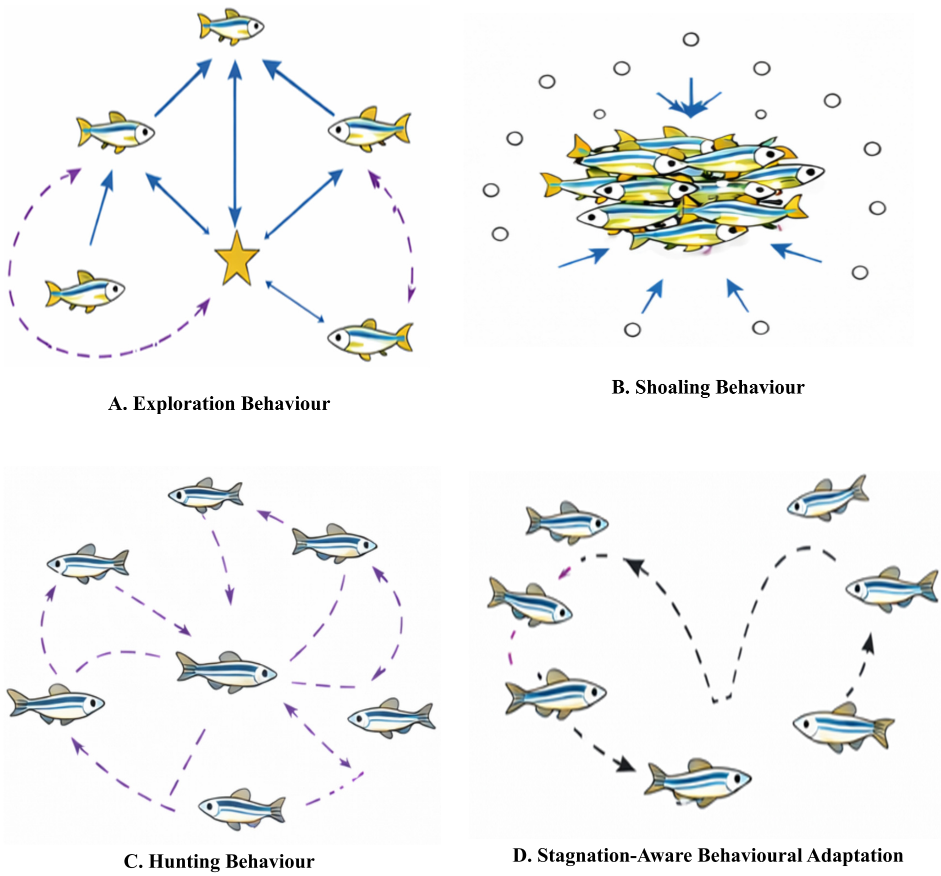

Fig. 1. Behavioural modelling of Zebrafish Optimization Algorithm (ZFO).

2.4. *Computational Complexity*

The Zebrafish Optimization (ZFO) algorithm has a computational complexity that is directly linked to three important parameters: Population size N, dimension d and maximum iterations $\text{Max}_{\text{iteration}}$, that will be conducted with the algorithm. The computational cost of creating the initial zebrafish population and assessing their fitness during initialization requires $\mathcal{O}(N \times d)$ time. The calculation of population statistics such as highest median fitness during each iteration has an $\mathcal{O}(N \log N)$ cost associated. Finally, the most computationally expensive part of ZFO is calculating the distances among zebrafish; this most commonly occurs when identifying the neighbours closest to a given zebrafish and the independent zebrafish sends its updates in isolation from other zebrafish, the method proposed is ideally suited for parallel and distributed implementations.

Figure 1 shows various behaviours of Zebrafish shown in A, B, C and D. The exploration behaviour is illustrated in figure A, which depicts how individual zebrafish will disperse outwards in random directions through the search space looking for food (the goal) and creating a global exploration of the environment while preventing premature or simultaneous convergence to food at the same time. The shoaling behaviour is illustrated in figure B, which shows how zebrafish use the same movement direction together in groups and communicate with one another, allowing them to better utilize their time by sharing information and creating a cohesive population. This shoaling behaviour would allow zebrafish to move toward more productive parts of the overall environment, enabling them to exploit these specific areas more efficiently. The hunting behaviour is shown in figure C, which illustrates a group of zebrafish that will aggressively come together in a concentrated manner in order to capture their target or prey. This illustrates how the hunting behaviour corresponds to the intensified search localized around the best solution found to date. The stagnation-aware behaviour shown in figure D allows zebrafish to modify their movements if they sense there is no further progress being made. When this occurs, the movement patterns change, allowing for re-establishment of diversity in their search process and the ability to jump out of any local optimum situation. With the use of zebrafish images, the biological feasibility of the ZFO is greatly enhanced visually, thereby reinforcing the correlation between the behaviours of natural zebrafish and the respective algorithmic representation of these behaviours.

2.5. *Algorithm Framework*

The ZFO Algorithm 1 flowchart is illustrated in Fig. 2. From an initialization of a zebrafish population (within established search bounds), fitness evaluations take place to ascertain which individuals represent the "global best". In each iteration, each individual zebrafish will compute a "fairness factor" for itself based on its fitness relative to other members of its population; they will also determine their relative neighbourhoods. If an individual achieves a greater degree of fitness than its neighbours during the current iteration, that would indicate that it occupies the region of the "local centre" as well as the "local best". A stagnation counter will be continually updated to monitor the extent of one's progress from one iteration to the next; if, within a given period, no improvement has occurred, the stagnation counter will reach its "stagnation threshold". When that occurs, the zebrafish that has become stagnant is forced into a state of exploration, which serves to help it escape from being trapped in a local optimum. All members of the population have equal and independent probabilistic activations of states associated with the three aforementioned behaviours: exploration, shoaling and hunting; the behaviours of shoaling and hunting can assist zebrafish in rapidly changing their positions during position updates while ensuring a balance between reliable global exploration and effective local exploitation. After boundary control, the fitness values of all members will be reevaluated to identify and document the global best candidate for the current iteration. This process will continue until the designated maximum iteration number is reached, at which time the convergence history will be documented for further analysis.

3. Experimental Analysis

The proposed ZFO algorithm is evaluated through experimental assessment by analysing its effectiveness, robustness and convergence speed to an optimal solution and compared

Algorithm 1 Pseudo code of Proposed Zebrafish Optimization (ZFO)

Input: Population size N, maximum iterations T, lower bound lb, upper bound ub, problem dimension d, objective function $f(\cdot)$
Output: Best fitness value *Best_ score*, best solution *Best_pos*, convergence curve

1. Initialize population $X_i^{t+1} = LB + R \odot (UB - LB), i = 1, 2, \ldots, N$
2. Evaluate fitness $F_i = f(X_i)$
3. Identify global best $Best\_score$ and $Best\_pos$
4. Initialize behavioural states randomly (Exploration or Hunting)
5. Initialize stagnation counter $stag_i = 0$
6. for $t = 2$ to T do
7. Compute normalized time $\tau = \frac{t}{T}$
8. Update exploration step size using (4) annealing
9. Compute population statistics: $F_{\text{best}}, F_{\text{median}}$
10. For each zebrafish $i = 1$ to N do
11. Compute fairness factor

$$\phi_i = \max\left(0, \frac{F_{\text{median}} - F_{\text{best}} + \varepsilon}{F_i - F_{\text{best}}}\right)$$

12. Determine k nearest neighbours
13 Compute local shoal centre C_i and local best neighbour using (6)
14. Update stagnation counter
15. Compute behaviour probabilities using (13) to (15)

$$P_{\text{explore}} = 1 - \tau + 0.5\phi_i, \quad P_{\text{hunt}} = \tau + 0.3\phi_i$$

16. if $stag_i \geqslant stagnation\_limit$ then
17. State $\leftarrow$ Exploration
18. Else
19. Select state $\in$ {Exploration, Shoaling, Hunting} probabilistically
20. end if

State-Based Movement Update

21. if State = Exploration then using (3)
22. Generate random direction
23. Update position using adaptive exploration step
24. else if State = Shoaling then using (7)
25. Move toward local centre
26. Apply separation from close neighbours
27. Add small random perturbation
28. else if State = Hunting then
29. Compute global and local hunting directions using (8)
30. Combine directions using time-adaptive weight using (9)
31. Update position with fairness-scaled step using (11)
32. end if
33. Apply boundary control to updated position
34. end for
35. Evaluate fitness of updated population
36. Update stagnation counters
37. Replace old population with updated population
38. Update global best solution
39. Store best fitness in convergence curve
40. end for

Return *Best _ score*, *Best _ pos* and convergence curve

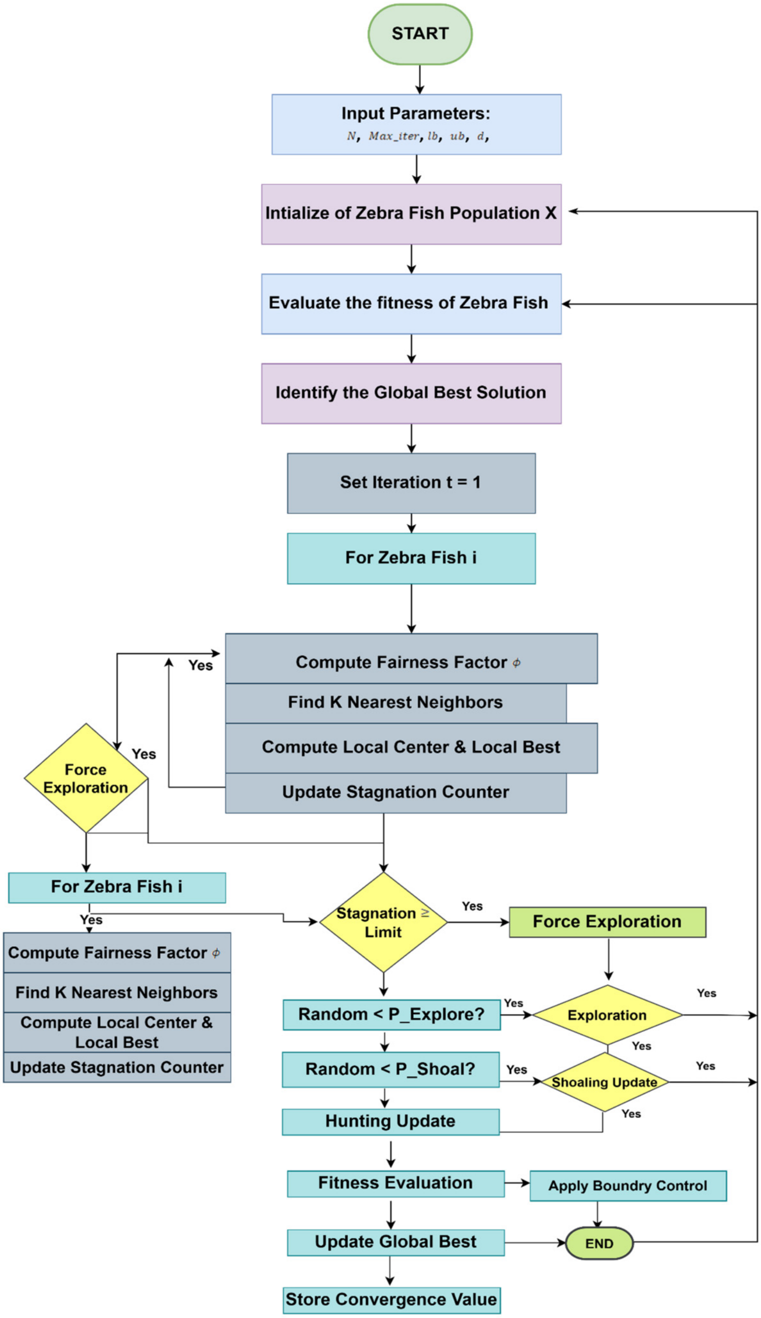

Fig. 2. Flowchart of proposed ZFO algorithm.

against other well-known metaheuristic approaches. All experiments were conducted under identical experimental conditions to fairness equality and reproducibility among experiments.

3.1. *Benchmark Functions*

The Zebrafish Optimization (ZFO) algorithm employed a number of commonly used benchmark functions. These types of benchmark functions are frequently used in optimization to evaluate an optimization algorithm's ability to explore and exploit the solution space and how it converges toward a solution based on different landscape structures; they have been used to compare and contrast other metaheuristic optimization algorithms based on their ability to adapt to different environmental conditions. The benchmark functions selected for use in the current study include categories of functions that contain unimodal, multimodal and fixed dimensional multimodal. There is a balanced and comprehensive test of all forms of benchmark function types available for use in the ZFO evaluation process.

The ZFO algorithm was benchmarked against three classes of classic benchmark problems: the unimodal functions (F1–F7), the multimodal functions (F8–F13) and the fixed dimension multimodal problems (F14–F23). Unimodal functions test exploration and convergence precision, multimodal functions test the problem of moving out of local minima and fixed-dimension multimodal functions have relatively low dimensionality and smoother landscapes. All benchmark problems could be properly characterized with mathematical definition, search boundaries and global optimum data taken from the comprehensive, standard library given by Surjanovic and Bingham (Surjanovic and Bingham, 2013): https://www.sfu.ca/~ssurjano/optimization.html. The source code of the ZFO algorithm together with all benchmark test functions needed for full reproducibility is publicly available at (Suruthi and Ganesh, 2026): https://github.com/suruthi-m/Zebr-Fish-Optimization.git.

All experiments were run on three different configurations of population size, number of operations and stopping condition. Each test problem instance ran over multiple runs to average out the random variations between runs and allow a better comparison between the solutions. Performance statistics of the fitness values produced over each benchmark include the minimum fitness achieved, the average and standard deviation of all of the fitness values computed over all the runs. The performance statistics will serve as a detailed basis for the examination of the advantages and disadvantages of the new ZFO algorithm and to produce well-justified comparisons with other optimisers within the discussion section.

3.2. *Comparison of ZFO with Some State-of-the-Art Algorithm Metaheuristic Algorithms*

The Zebrafish Optimization (ZFO) Algorithm was evaluated with different state-of-the-art Meta-Heuristic Algorithms, including the Cuttlefish Optimization Algorithm (CFO), Jellyfish Search Optimizer (JSO), Pufferfish Optimization Algorithm (POA) and the Krill

Herd Optimizer (KHO). Cuttlefish are aquatic animals that change their colour based on their environment, enabling them to camouflage themselves effectively, which makes the use of cuttlefish for optimization extremely effective. The Cuttlefish Optimization Algorithm (CFO), which is based on the colour-changing capabilities of cuttlefish, predominantly relies on the process of reflection and visibility operations to balance between exploration and exploitation of solution spaces. Although CFO has a strong capability for global exploration during the initial phase of each search, its capability to exploit locally optimal solutions tends to diminish in highly multimodal landscapes as a result of continually switching between exploration and exploitation phases. The Jellyfish Search Optimizer (JSO) mimics jellyfish move due to current influences and swarm behaviour. JSO employs both passive and active movement to properly explore the search space. JSO's performance can decline in late iterations due to not providing adequate exposure pressure. The Pufferfish Optimization Algorithm (POA) is based on the defensive and adaptive behaviours of pufferfish. POA has a high level of diversity, but due to the nature of the landscape it may experience oscillations in convergence to the optimum location. The Krill Herd Optimizer (KHO) uses how krill herd together and includes induced movement of individuals, foraging activity and physical movement around space (diffusion). KHO allows for continued diversity within the population, but due to multiple operators that interact, computing the optimum using KHO can increase its complexity. The parameter settings used for all benchmark problems are detailed in Table 1. These are control parameters and were found empirically and set at the same values for all problems to maintain fairness in comparison and ensure reproducibility.

3.3. *Qualitative Analysis*

ZFO performs can be viewed qualitatively from the Search history, Search path and Convergence charts in Figs. 3, 4 and 5. As shown in the search history chart, candidate solutions at the start of the optimization process are found over a large area of the solution space, indicating an excellent exploratory capacity, as well as their ability to cover/locate areas not previously searched. As the optimization process continues, candidate solutions cluster around identified promising solution areas, indicating a successful movement from Exploration to Exploitation. The search path chart exhibits the direction of movement of the best zebrafish toward the global optimum. This movement illustrates how effective the Hunting and Shoaling mechanisms direct the search process. Also, the Convergence curve exhibits a rapid decline in fitness from the start of the optimization process to the end of the optimization process, while the trend of average fitness confirms that there is consistent population-level convergence of candidate solutions toward the extremely competitive global optimum for the benchmark function. As shown, ZFO maintains a diverse population while achieving steady and reliable convergence to the benchmark function.

3.4. *Performance Analysis and Mechanistic Analysis*

In addition to providing evidence for the relative effectiveness of ZFO, this section seeks to explore at which problem instances and for which reasons the proposed ZFO model

Table 1
Parameter configuration for the Proposed Zebrafish Optimization Algorithm.

Parameter	Symbol	Value/Formula	Description
Population size	N	30	Number of Candidate Agents for the Zebrafish. It is set to 30–50.
Max iterations	T	500/1000	Max number of iterations
Search dimension	d	*Problem-dependent*	Dimensions of search space
Max exploration step	$\alpha_{\max}$	$0.3 \times$ range	Exploration step size max (lb, ub)
Min exploration step	$\alpha_{\min}$	$0.05 \times$ range	Exploration step size min
Fairness weight (explore)	w_{e}	1.0	Multiplier of fairness factor during exploration of step sizes
Separation weight	γ	$0.5 \times \beta$ ($\approx 0.05 \times$ range)	Repulsion/separation coefficient applied to S_i in the shoaling update, set as half the shoaling attraction step
Shoaling base step	β	$0.1 \times$ range	Attraction rate of the shoal centroid in movement during shoaling
Shoaling noise	δ_s	0.05	Stochastic component during shoaling movement
Neighbourhood size	k	$\max(2, \mathrm{round}(\sqrt{N}))$	Number of closest neighbours for local centroid movement
Hunting base step	η	$0.15 \times$ range	Attraction rate of prey in the movement of the hunter
Hunting noise	ξ	0.02	Stochastic component during hunter movement
Fairness weight (hunt)	w_{h}	0.5	Multiplier of fairness factor during hunting of step sizes
Global weight (hunting)	w_g	$0.5 + 0.5\tau$	Time variant influence between global search direction and local hunter movements
Exploration probability	P_{explore}	$(1-\tau) + 0.5\phi_i$	Probability of entering in to an exploratory movement
Hunting probability	P_{hunt}	$\tau + 0.3\phi_i$	Probability of hunting behaviour for a certain agent in the step of an iteration
Shoaling probability	P_{shoal}	$1 - (P_{\text{explore}} + P_{\text{hunt}})$	Probability of shoaling behaviour for a certain agent in the step of an iteration
Stagnation threshold	S_{th}	$\max(3, \mathrm{round}(0.05 \times T))$	Consecutive steps without improvement before a forced re-start
Fairness factor	ϕ_i	$\max\left(0, \frac{F_{\text{median}} - F_{\text{best}} + \varepsilon}{F_i - F_{\text{best}}}\right)$	Fairness weight function based on agent fitness [i is index of the agent, the best one gets $i = 0$]
Normalised time	τ	$\frac{t}{T}$	Progress over time of algorithm execution from 0 to 1

offers superior results in comparison to others. Emphasis is given to the individual role played by exploration, shoaling, hunting and stagnation detection and parameter adaptive mechanisms in establishing the optimal equilibrium between exploration and exploitation under varying scenarios.

3.4.1. *Performance Analysis on Unimodal Benchmark Functions*

The performance of the Zebrafish Optimization Algorithm (ZFO) was compared against four of the most recently developed metaheuristic algorithms (CUTTLE, JS, POA and KH) across seven unimodal benchmark functions (F1–F7). Table 2 shows that the algorithms were tested under exactly the same conditions (50 separate trials with 500 iterations) and Table 3 shows the algorithms were tested under exactly the same conditions (50 separate trials with 1000 iterations) and the best, average and standard deviation (STD) of each

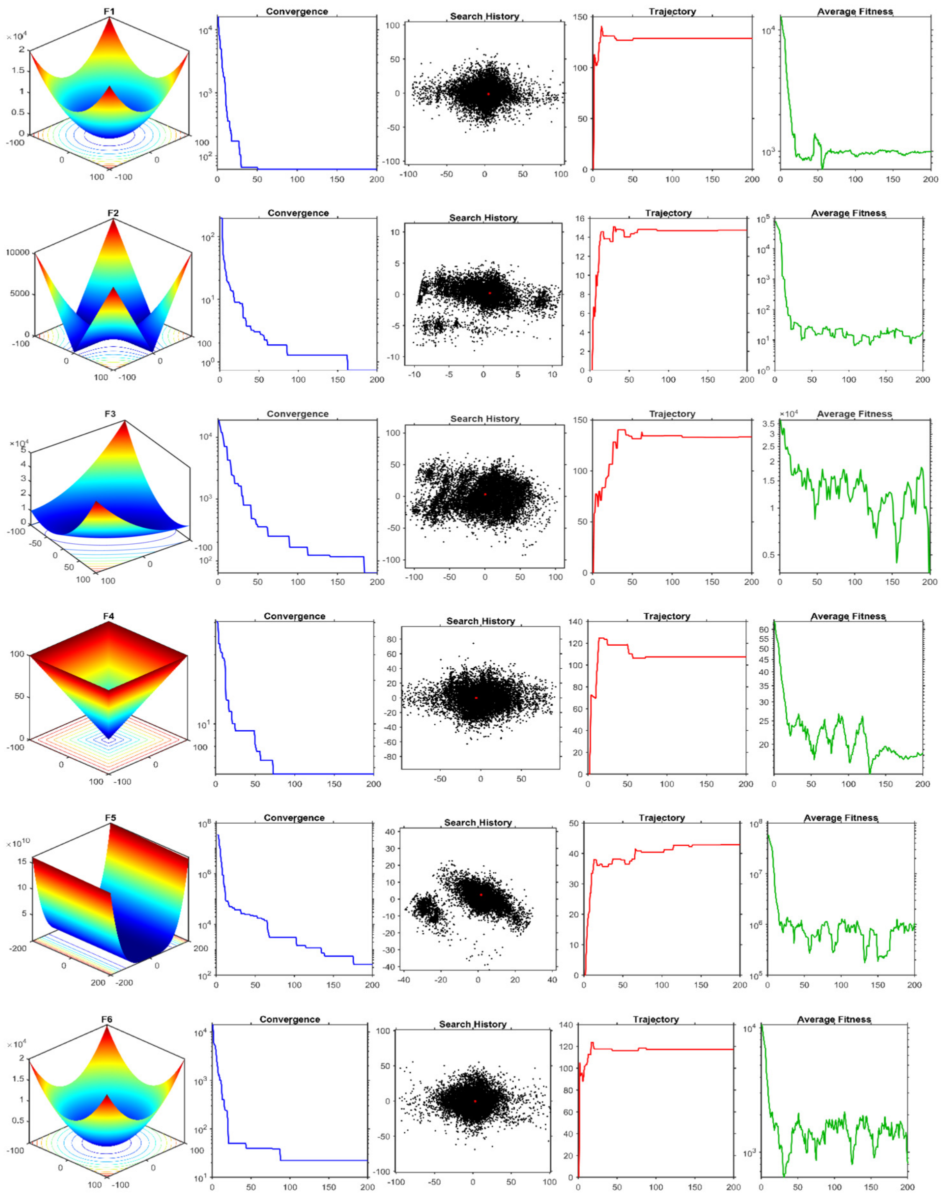

Fig. 3. Qualitative results of F1 to F7 ZFO on unimodal benchmark functions.

algorithm's fitness score were captured. Figures 6 and 7 show the convergence behaviour of the algorithms (50 separate trials with 500 iterations and 50 separate trials with 1000 iterations) compared on the unimodal benchmark functions, which is shown along with their optimization dynamics across the iteration process.

ZFO consistently outperforms the other tested algorithms on all unimodal test functions. Specifically, ZFO has lower mean fitness values and lower standard deviation val-

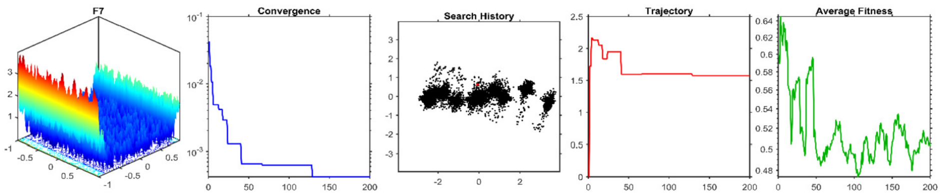

Fig. 3. (*continued*)

ues, indicating excellent exploitation and convergence behaviour. ZFO's low variability (i.e. low variance) in independent experimental runs also suggests that it can be relied upon to produce similar results regardless of the run.

For simpler unimodal functions like F1, F2 and F7, ZFO quickly approaches the global optimum and generates more accurate solutions than those created by CUTTLE, JS, POA and KH. The ZFO Algorithm's strength lies in its ability to perform intensive exploration of the more promising areas of the search space to find these points. The competing algorithms show higher mean errors and a greater spread, indicating they will take longer to converge and are more likely to find fewer optimum paths during their search process.

Additionally, for higher dimensional unimodal problems, ZFO continues to outperform the alternatives through its ability to honour the trade-off between encouraging movement toward the "optimal" solution while avoiding premature stagnation. The results for POA and CUTTLE were significantly poorer than ZFO, demonstrating relatively large mean fitness values and high standard deviations, indicating that they are less effective at exploiting the available resources within unimodal landscapes.

The extensive experiments with unimodal benchmark functions, the performance gained from using ZFO has been validated and is extremely effective, accurate and robust. ZFO is an extremely effective optimizer in terms of utilising its extreme capability for finding a singular, global optimum and is especially beneficial for solving optimization issues, including those with a singular optimizer.

3.4.2. *Performance Analysis on Multimodal Benchmark Functions*

A comparison between the ZFO algorithm and four other stochastic, multi-dimensional optimization methods is presented in Table 4 and 5. All methods were applied in three levels of testing, consisting of the recommended number of trials (50 with 500 iterations and 50 with 1000 iterations). The trial results were compiled into tables including the best trial results, average results and standard deviations for all trial results obtained for each level of testing. Additionally, Figs. 8 and 9 show the convergence behaviour of the algorithms (50 separate trials with 500 iterations and 50 separate trials with 1000 iterations) compared on the multimodal benchmark functions, which is shown along with their optimization dynamics across the iteration process.

Data was collected across all run sizes and iteration limits to develop the ZFO algorithm's ability to do better than, or at least as well as the benchmark algorithms. The results indicate that ZFO has superior exploration capability in the early stages of the search process and gradually increases its ability to exploit as the number of iterations increases.

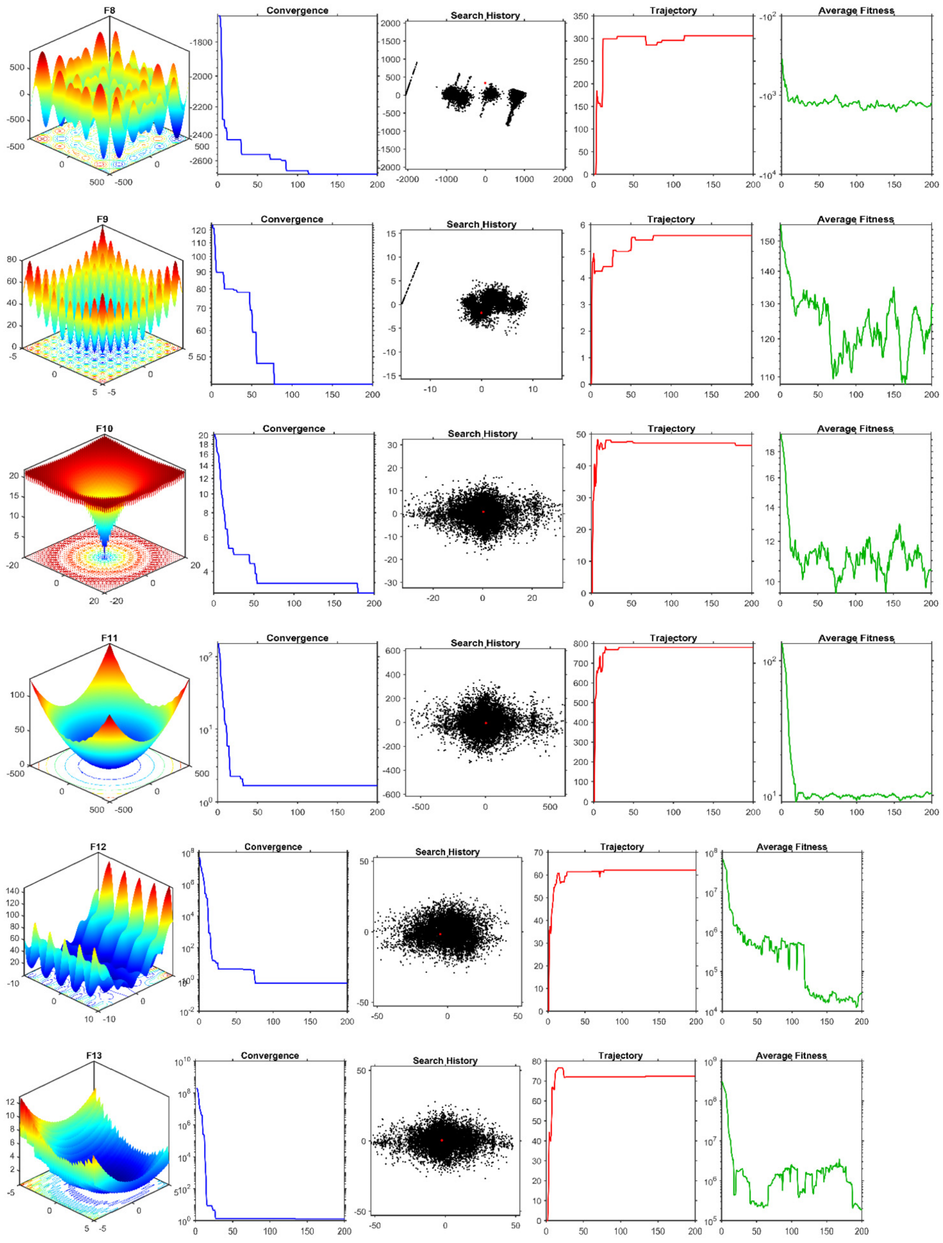

Fig. 4. Qualitative results of F8 to F13 ZFO on multimodal benchmark functions.

Through this process, ZFO is able to traverse multi-modal optimally and prevents a premature fall back into local optima.

As the number of runs increasing from 25 to 50, the performance of ZFO is more consistent with lower values of standard deviation over all multimodal functions. These results demonstrate the robustness and reliability of the proposed algorithm when subjected

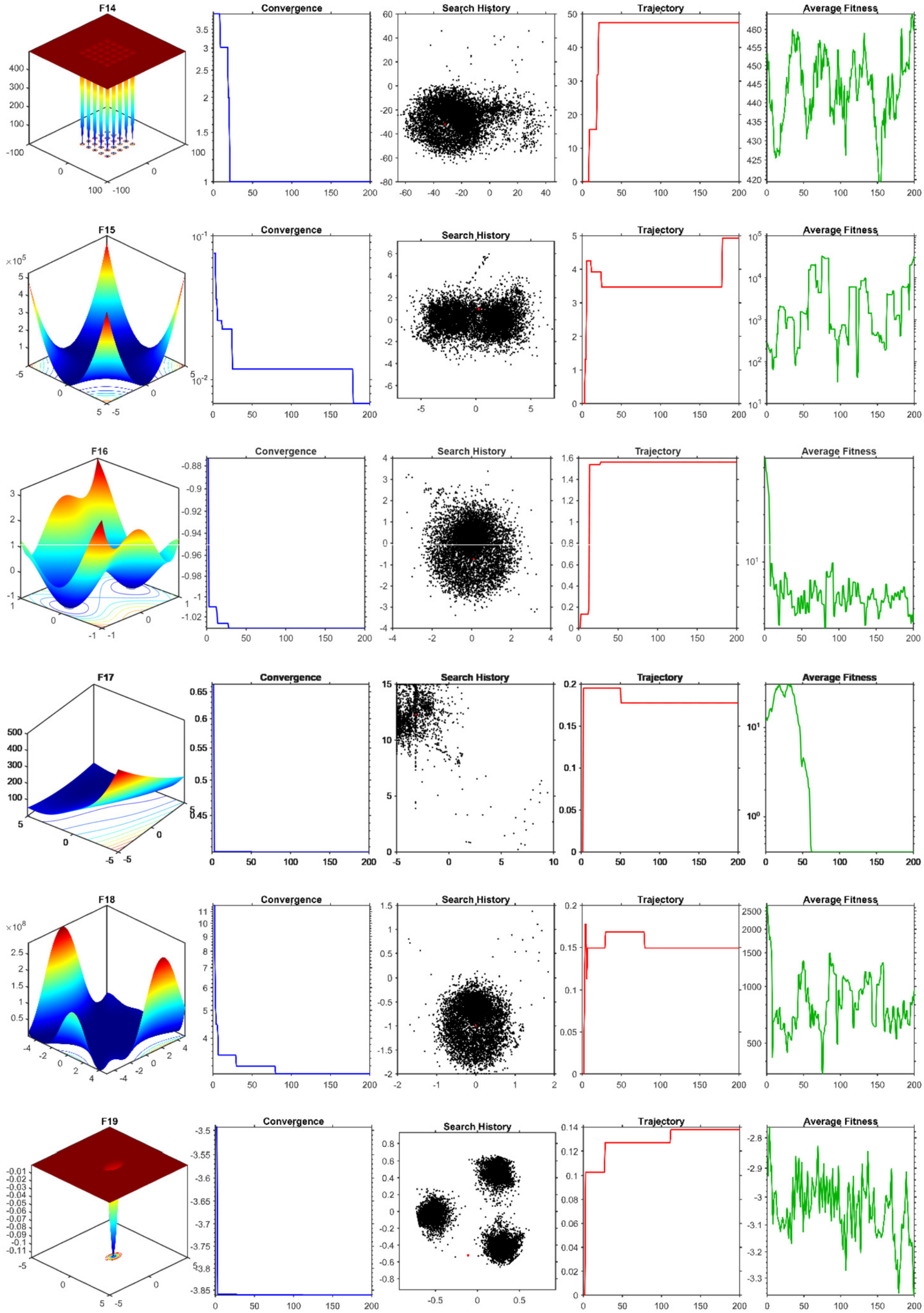

Fig. 5. Qualitative results of F14 to F23 ZFO on fixed dimension multimodal benchmark functions.

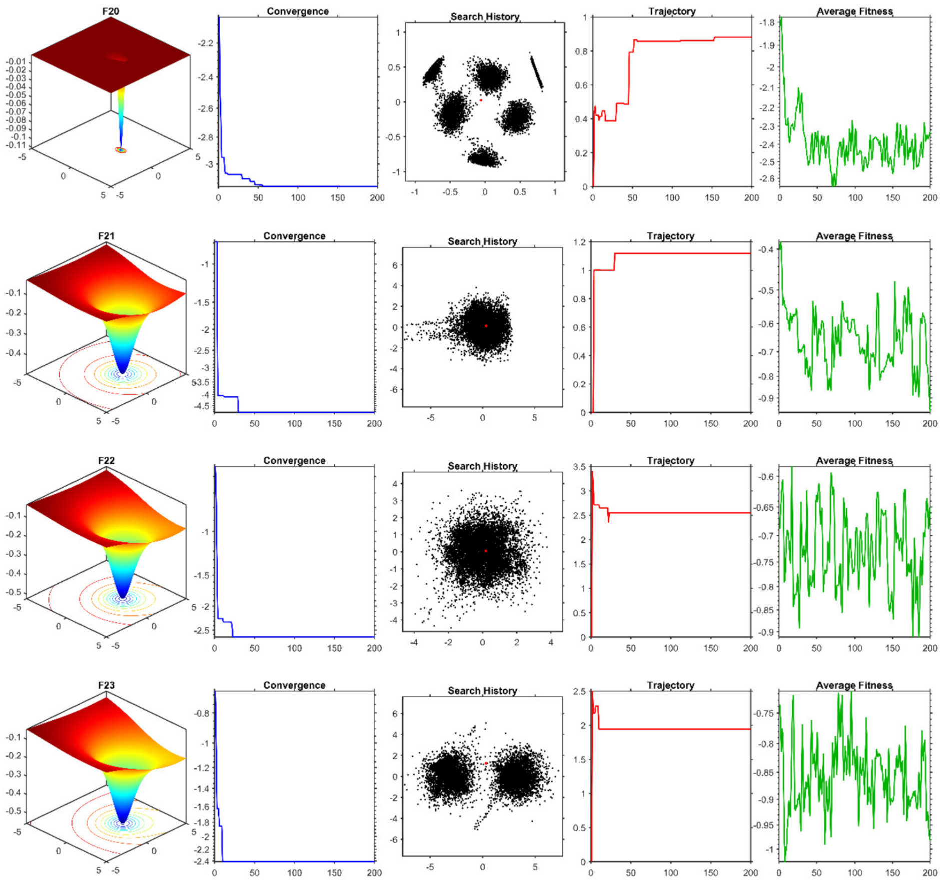

Fig. 5. (*continued*)

to diverse stochasticity. On the other hand, other proposed algorithms, such as POA and KH demonstrate a high degree of performance variability, indicating that they are more susceptible to differences in both initialization and search randomness.

Additionally, by increasing the number of iterations from 500 to 1000 iterations, a greater number of iterations will allow ZFO to refine the quality of its solutions beyond their current mean fitness value without compromising the stability of their convergence rates. The ability of the algorithm to scale up in terms of both the number of runs and the number of iterations demonstrates that the algorithm can effectively utilize more computer resources to improve the accuracy of its optimization results.

The results of the experiments conducted in multiple runs, iterations and other configurations validate the capability, dependability and capacity of ZFO to solve multimodal optimization problems. These experimental configurations provide a consistent performance improvement across all trial types and substantiate the validity of ZFO as an appropriate tool for dealing with complex, high dimensional optimization problems that are characterized by multiple local optima.

Table 2
Result for unimodal F1–F7 with 50 runs and 500 iterations.

Function	ZFO			CUTTLE			JS	
	BEST	MEAN	STD	BEST	MEAN	STD	BEST	MEAN
F1	9.98E+02	1.23E+03	1.38E+02	2.62E+04	3.32E+04	2.53E+03	5.02E+03	1.03E+04
F2	1.81E+01	2.28E+01	2.21E+00	2.49E+02	3.89E+08	1.17E+09	4.19E+01	5.88E+01
F3	6.75E+03	1.31E+04	2.99E+03	5.04E+04	7.90E+04	8.32E+03	6.70E+03	1.76E+04
F4	1.65E+01	2.78E+01	4.64E+00	6.52E+01	7.58E+01	4.04E+00	2.39E+01	3.58E+01
F5	5.60E+04	9.35E+04	2.30E+04	3.44E+07	5.39E+07	8.57E+06	1.64E+06	5.33E+06
F6	9.85E+02	1.23E+03	1.33E+02	2.06E+04	3.19E+04	2.81E+03	5.65E+03	1.09E+04
F7	1.89E−01	3.47E−01	8.95E−02	2.71E+01	4.14E−01	7.15E+00	1.36E+00	3.73E+00

Function	POA			KH		
	STD	BEST	MEAN	STD	BEST	MEAN
F1	2.43E+03	3.91E+04	5.30E+04	4.77E+03	1.35E+04	3.14E+04
F2	8.51E+00	1.28E+05	4.13E+11	1.95E+12	3.18E+01	4.56E+01
F3	5.73E+03	7.31E+04	9.91E+04	1.05E+04	5.14E+03	1.02E+04
F4	3.99E+00	7.12E+01	7.89E+01	2.82E+00	5.77E+01	7.38E+01
F5	2.30E+06	7.97E+07	1.23E+08	2.51E+07	1.15E+06	8.23E+06
F6	2.65E+03	3.67E+04	5.18E+04	5.77E+03	1.32E+04	2.84E+04
F7	1.43E+00	5.22E+01	8.86E+01	1.68E+01	7.01E−02	2.39E−01

Table 3
Result for unimodal F1–F7 with 50 runs and 1000 iterations.

Function	ZFO			CUTTLE			JS		
	BEST	MEAN	STD	BEST	MEAN	STD	BEST	MEAN	STD
F1	7.39E+02	1.04E+03	1.13E+02	2.11E+04	2.57E+04	1.77E+03	3.20E+03	1.05E+04	2.27E+03
F2	1.72E+01	2.45E+01	2.43E+01	1.37E+02	1.76E+07	5.05E+07	4.03E+01	5.99E+01	8.46E+00
F3	5.93E+03	9.86E+03	1.85E+03	5.82E+04	6.99E+04	5.64E+03	7.46E+03	1.88E+04	5.74E+03
F4	1.48E+01	2.13E+01	4.70E+00	5.77E+01	7.01E+01	4.82E+00	2.24E+01	3.35E+01	3.80E+00
F5	3.73E+04	6.50E+04	1.60E+04	2.20E+07	3.49E+07	4.85E+06	1.54E+06	4.83E+06	2.16E+06
F6	8.00E+02	1.03E+03	1.09E+02	2.15E+04	2.56E+04	1.84E+03	5.05E+03	1.03E+04	2.48E+03
F7	1.07E−01	2.45E−01	7.13E−02	1.49E+01	2.61E+01	4.58E+00	5.89E−01	4.12E+00	1.86E+00

Function	POA			KH		
	BEST	MEAN	STD	BEST	MEAN	STD
F1	2.06E+04	3.11E+04	3.72E+03	1.42E+04	3.00E+04	1.04E+04
F2	1.54E+02	1.42E+08	5.22E+08	3.25E+01	4.55E+01	4.96E+00
F3	6.56E+04	7.92E+04	6.65E+03	4.31E+03	9.74E+03	5.22E+03
F4	6.84E+01	7.41E+01	2.67E+00	4.16E+01	7.22E+01	1.01E+01
F5	2.92E+07	5.36E+07	1.06E+07	1.17E+06	5.73E+06	4.99E+06
F6	2.28E+04	3.14E+04	3.85E+03	1.57E+04	3.01E+04	8.91E+03
F7	2.37E+01	3.84E+01	6.91E+00	5.82E−02	1.43E−01	5.76E−02

3.4.3. *Performance Analysis on Fixed Dimension Multimodal Functions*

The performance of ZFO is compared with CUTTLE, JS, POA and KH by evaluating each algorithm when solving benchmark functions that have several variables and are multimodal. These problems were carefully designed so that each has a fixed number of decision variables and represent very complex optimization problems because many of the

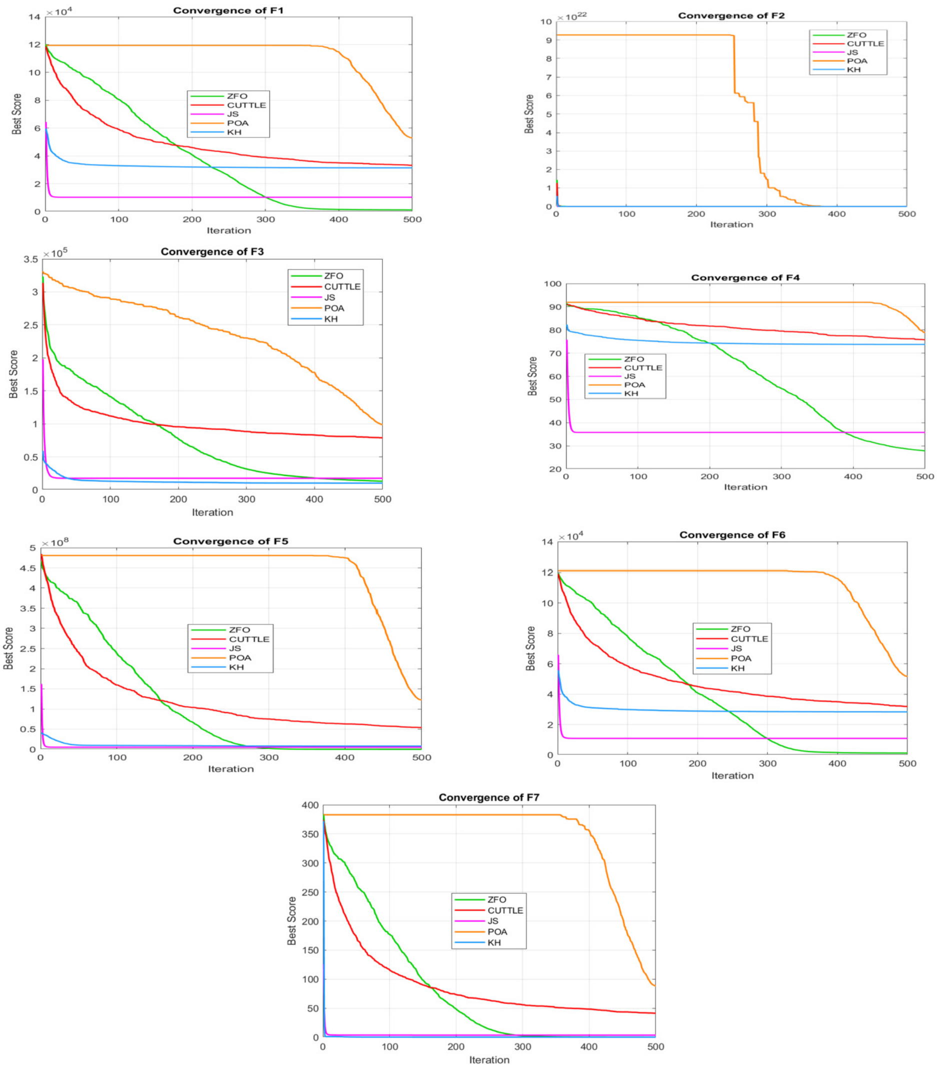

Fig. 6. Convergence curve for unimodal F1–F7 with 50 runs and 500 iterations.

variables have many local minima and they require very different exploitation strategies than traditional optimization problems. They provide an excellent means for testing an algorithm's robustness, exploration capability and convergence properties.

To allow for a fair and statistically significant comparison between the algorithms through multiple experiments, the following factors were taken into account: the number of independent runs per algorithm. Table 6 and 7 show the algorithms were tested under exactly the same conditions (50 separate trials with 500 iterations and 50 separate trials with 1000 iterations). After completing the experiments for each of these variations, the algorithms' real-time fitness values were analysed based on: best fitness value, mean fitness value and standard deviation (STD) for each configuration. Additionally, Figs. 10

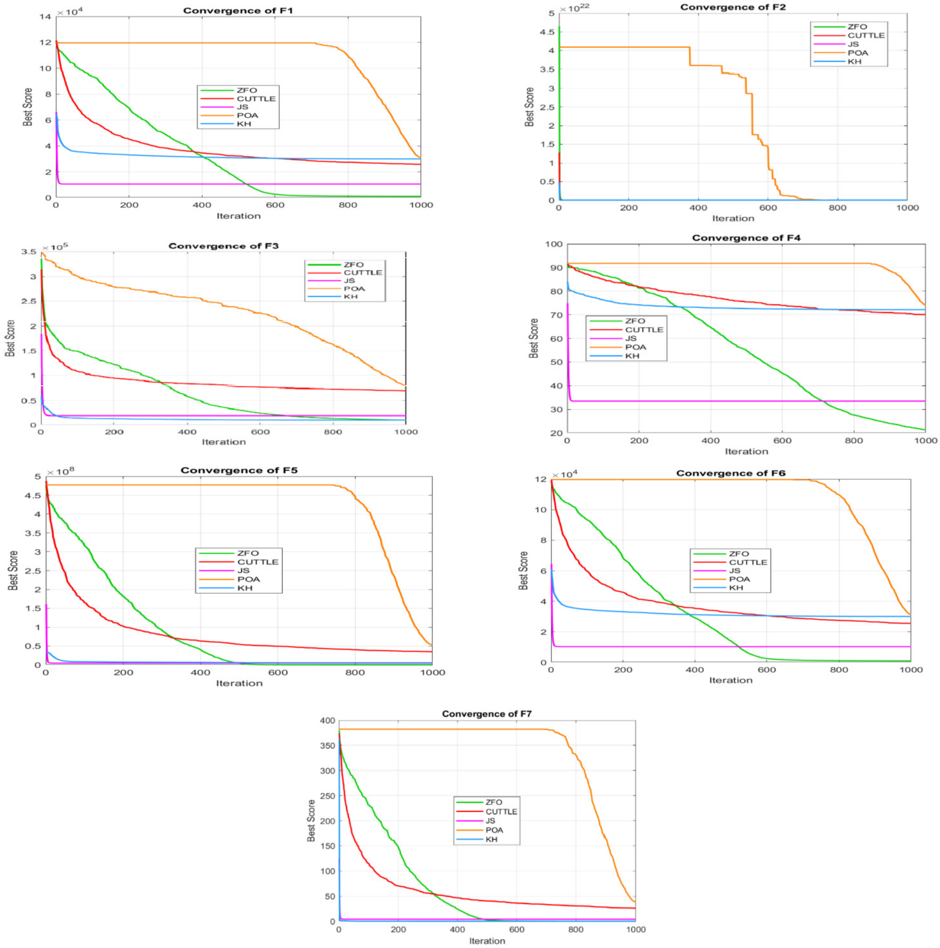

Fig. 7. Convergence curve for unimodal F1–F7 with 50 runs and 1000 iterations.

and 11 show the convergence behaviour of the algorithms (50 separate trials with 500 iterations and 50 separate trials with 1000 iterations) compared on the fixed dimensional multimodal benchmark functions, which is shown along with their optimization dynamics across the iteration process of 50. Each independent run was completed with 500 iterations or 1000 iterations.

ZFO performed better than any other algorithm for the entire fixed-dimensional multimodal function suite. More importantly, ZFO is consistently able to avoid the pitfalls of local optima while maintaining a diverse population. This is evidenced by ZFO's lower and more consistent average fitness over repeated runs. Therefore, ZFO has effectively found a good balance between exploration and exploitation when operating in a multimodal environment.

An increased run count of 50 demonstrated improved stability of ZFO as indicated by consistently lower standard deviation and it shows greater stability in the presence

Table 4
Result for multimodal F8–F13 with 50 runs and 500 iterations.

Function	ZFO			CUTTLE			JS		
	BEST	MEAN	STD	BEST	MEAN	STD	BEST	MEAN	STD
F8	−1.14E+04	−9.30E+03	9.13E+02	−6.29E+03	−5.54E+03	2.98E+02	−7.58E+03	−5.84E+03	7.26E+02
F9	3.21E+02	3.88E+02	3.01E+01	5.43E+02	5.84E+02	2.11E+01	2.43E+02	2.99E+02	2.52E+01
F10	6.73E+00	7.47E+00	4.10E−01	1.76E+01	1.86E+01	3.27E−01	1.11E+01	1.33E+01	7.60E−01
F11	8.74E+00	1.19E+01	1.32E+00	2.22E+02	2.95E+02	3.20E+01	5.41E+01	9.53E+01	2.09E+01
F12	7.98E+00	1.07E+02	4.71E+02	3.08E+07	7.41E+07	2.50E+07	1.51E+03	5.50E+05	5.71E+05
F13	1.53E+02	6.11E+03	1.00E+04	1.10E+08	2.08E+08	3.51E+07	5.36E+05	6.75E+06	5.17E+06

Function	POA			KH		
	BEST	MEAN	STD	BEST	MEAN	STD
F8	−8.00E+03	−6.49E+03	5.36E+02	−9.64E+03	−9.29E+03	2.73E+02
F9	5.64E+02	6.16E+02	2.28E+01	1.89E+01	4.46E+01	1.64E+01
F10	2.01E+01	2.02E+01	4.39E−02	1.18E+01	1.44E+01	1.35E+00
F11	3.65E+02	4.75E+02	5.11E+01	3.03E+02	7.87E+02	2.56E+02
F12	8.04E+07	2.11E+08	4.62E+07	2.25E+05	7.64E+06	1.21E+07
F13	2.66E+08	5.16E+08	1.07E+08	3.29E+06	3.71E+07	3.66E+07

Table 5
Result for multimodal F8–F13 with 50 runs and 1000 iterations.

Function	ZFO			CUTTLE			JS		
	BEST	MEAN	STD	BEST	MEAN	STD	BEST	MEAN	STD
F8	−1.17E+04	−1.01E+04	7.10E+02	−7.41E+03	−5.91E+03	4.20E+02	−7.93E+03	−5.98E+03	7.32E+02
F9	3.02E+02	3.74E+02	3.44E+01	4.91E+02	5.68E+02	2.14E+01	2.23E+02	2.82E+02	2.54E+01
F10	6.42E+00	6.94E+00	2.56E−01	1.72E+01	1.78E+01	2.55E−01	1.17E+01	1.35E+01	8.38E−01
F11	8.35E+00	1.00E+01	9.54E−01	1.96E+02	2.33E+02	1.62E+01	6.02E+01	9.38E+01	1.88E+01
F12	8.15E+00	1.58E+01	5.57E+00	1.58E+07	3.61E+07	1.10E+07	2.46E+03	4.73E+05	4.47E+05
F13	6.55E+01	1.91E+03	4.44E+03	3.45E+07	1.12E+08	2.66E+07	1.13E+06	8.29E+06	6.58E+06

Function	POA			KH		
	BEST	MEAN	STD	BEST	MEAN	STD
F8	−9.03E+03	−7.11E+03	5.12E+02	−9.62E+03	−9.33E+03	1.13E+02
F9	5.01E+02	5.75E+02	2.54E+01	2.09E+01	4.35E+01	1.58E+01
F10	2.00E+01	2.01E+01	3.64E−02	9.71E+00	1.40E+01	1.35E+00
F11	2.15E+02	2.85E+02	3.52E+01	4.25E+02	8.15E+02	1.56E+02
F12	8.55E+06	6.30E+07	2.22E+07	4.41E+04	4.86E+06	5.16E+06
F13	8.70E+07	1.71E+08	4.23E+07	3.41E+06	2.92E+07	4.27E+07

of stochastic variation. ZFO can further enhance the quality of the solution when run over 500–1000 iterations, improving convergence accuracy without negatively impacting stability.

In the search phase, POA and KH often substitute a restricted number of search patterns for a variety of fixed-dimensional multimodal functions, which causes premature convergence. On the other hand, compared to the other competing algorithms, the overall performance of CUTTLE and JS is more variable. In addition, evaluation shows that the new algorithm ZFO is better able to maintain diversities in the search process and provide steady convergence to high-quality solutions.

Overall, the experimentation has demonstrated that ZFO is a high-performance method for Fixed-Dimension Multimodal Optimization Problems due to its superior robustness,

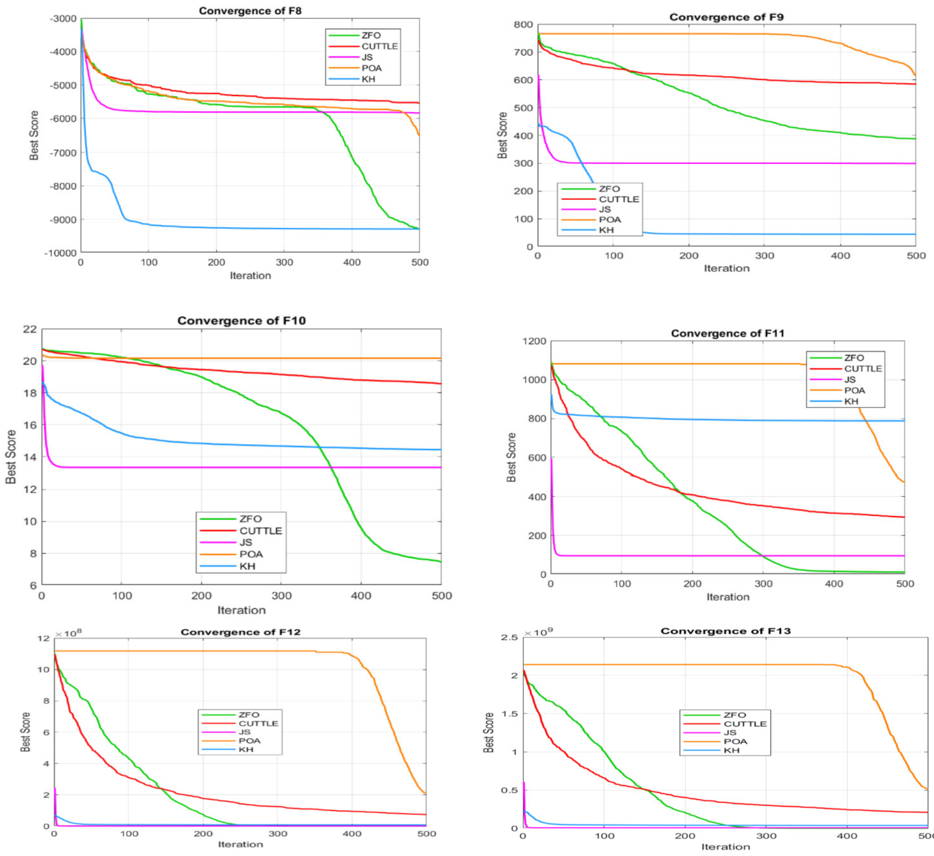

Fig. 8. Convergence curve for multimodal F8–F13 with 50 runs and 500 iterations.

scalability and convergence reliability and that the increases in performance seen across various run sizes and iteration limits support ZFO's efficacy with respect to the solution of difficult complex optimization problems characterized by rugged and deceptive search spaces.

3.5. *Statistical Analysis*

The performance of ZFO was rigorously validated through a statistical analysis of all four algorithms (Cuttle, JS, POA and KH) to compare the proposed ZFO proposal against other algorithms. Because metaheuristic algorithms operate by generating random outcomes, the results are non-normally distributed and do not meet the criteria for normality. Nonparametric statistical tests were the most effective way to evaluate the comparison and to provide a reliable, unbiased assessment of the performance of the algorithms.

3.5.1. *Friedman Test*

The Friedman test was conducted independently for each of the benchmark functions to test the algorithms' abilities. The Friedman test was performed on three sets of data – the

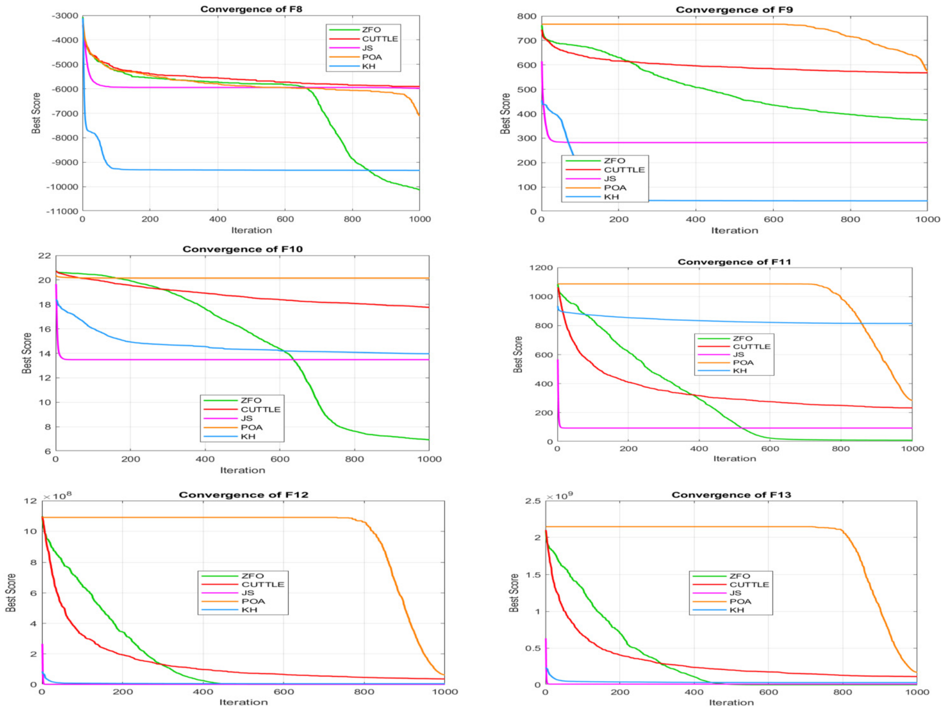

Fig. 9. Convergence curve for multimodal F8–F13 with 50 runs and 1000 iterations.

results of 25, 30 and 50 independent trials with 500 and 1000 iterations and a significance level of $\alpha = 0.05$. The Friedman test provided mean rankings for the algorithms and the p-value associated with the mean. The results of the Friedman test showed ZFO had the lowest (and best) ranking among most unimodal and multimodal functions and fixed-dimensional multimodal functions. Additionally, the p-values generated by the Friedman test for many of the test functions indicated that the differences between the algorithms' performance were statistically significant, thus indicating no random variability would account for the observed differences in performance.

3.5.2. *Wilcoxon Signed Rank Test*

The results of the Wilcoxon signed-rank test demonstrate that there are statistically significant differences in performance between the ZFO algorithm and four other competing algorithms for each of the twenty three benchmark functions when using the best fitness values across independent runs of the ZFO and each of the competing algorithms. To complete the statistical test, the results obtained from each of the four competing algorithms were treated as a "reference" algorithm against which to compare ZFO. Before performing the Wilcoxon signed-rank test, all comparisons with missing values and identical fitness values were excluded to ensure the validity of the comparisons performed. The results obtained from conducting the Wilcoxon signed-rank test resulted in the conclusion that ZFO outperformed the competing algorithms in a statistically significant manner for most

Table 6
Result for fixed dimension multimodal F14–F23 with 50 runs and 500 iterations.

Function	ZFO			CUTTLE			JS		
	BEST	MEAN	STD	BEST	MEAN	STD	BEST	MEAN	STD
F14	9.98E−01	9.98E−01	1.59E−04	9.98E−01	1.00E−00	5.92E−03	9.98E−01	4.57E+00	2.87E+00
F15	6.77E−04	1.09E−03	2.27E−04	6.43E−04	1.09E−03	2.48E−04	3.99E−04	5.06E−03	9.57E−03
F16	−1.03E+00	−1.03E+00	2.54E−04	−1.03E+00	−1.03E+00	3.18E−04	−1.03E+00	−1.03E+00	3.52E−04
F17	3.98E−01	3.98E−01	4.52E−04	3.98E−01	3.98E−01	3.80E−04	3.98E−01	4.02E−01	1.19E−02
F18	3.00E+00	3.00E+00	3.30E−03	3.00E+00	3.00E+00	5.50E−03	3.00E+00	3.10E+00	5.41E−01
F19	−3.86E+00	−3.86E+00	8.97E−05	−3.86E+00	−3.86E+00	5.52E−04	−3.86E+00	−3.84E+00	2.66E−02
F20	−3.32E+00	−3.23E+00	5.62E−02	−3.31E+00	−3.29E+00	1.20E−02	−3.32E+00	−3.06E+00	2.24E−01
F21	−1.01E+01	−8.34E+00	2.26E+00	−9.69E+00	−8.02E+00	9.29E−01	−1.02E+01	−5.57E+00	3.48E+00
F22	−1.04E+01	−9.66E+00	1.21E+00	−9.83E+00	−8.31E+00	9.32E−01	−1.04E+01	−6.61E+00	3.24E+00
F23	−1.05E+01	−9.82E+00	1.01E+00	−9.85E+00	−8.25E+00	8.58E−01	−1.05E+01	−6.18E+00	3.29E+00

Function	POA			KH		
	BEST	MEAN	STD	BEST	MEAN	STD
F14	9.98E−01	9.99E−01	4.10E−03	1.02E+00	1.05E+01	8.73E+00
F15	9.67E−04	1.31E−03	1.85E−04	3.07E−04	4.28E−03	5.92E−03
F16	−1.03E+00	−1.03E+00	1.63E−04	−1.03E+00	−1.03E+00	1.69E−07
F17	3.98E−01	3.98E−01	1.36E−04	3.98E−01	3.98E−01	2.41E−06
F18	3.00E+00	3.00E+00	1.25E−03	3.00E+00	3.00E+00	7.95E−08
F19	−3.86E+00	−3.86E+00	1.57E−04	−3.86E+00	−3.82E+00	4.26E−02
F20	−3.32E+00	−3.31E+00	8.95E−03	−3.32E+00	−3.20E+00	1.31E−01
F21	−1.01E+01	−9.37E+00	5.25E−01	−1.02E+01	−7.26E+00	3.61E+00
F22	−1.03E+01	−9.75E+00	4.30E−01	−1.04E+01	−8.40E+00	3.28E+00
F23	−1.05E+01	−9.94E+00	4.30E−01	−1.05E+01	−7.22E+00	3.71E+00

Table 7
Result for fixed dimension multimodal F14–F23 with 50 runs and 1000 iterations.

Function	ZFO			CUTTLE			JS		
	BEST	MEAN	STD	BEST	MEAN	STD	BEST	MEAN	STD
F14	9.98E−01	9.98E−01	1.72E−05	9.98E−01	9.98E−01	9.48E−05	9.98E−01	4.46E+00	3.07E+00
F15	6.84E−04	1.12E−03	2.13E−04	6.13E−04	9.24E−04	1.46E−04	3.33E−04	4.72E−03	9.47E−03
F16	−1.03E+00	−1.03E+00	1.28E−04	−1.03E+00	−1.03E+00	1.91E−04	−1.03E+00	−1.03E+00	5.93E−05
F17	3.98E−01	3.98E−01	3.25E−04	3.98E−01	3.98E−01	2.81E−04	3.98E−01	4.29E−01	1.03E−01
F18	3.00E+00	3.00E+00	1.69E−03	3.00E+00	3.00E+00	2.59E−03	3.00E+00	3.61E+00	3.69E+00
F19	−3.86E+00	−3.86E+00	5.18E−05	−3.86E+00	−3.86E+00	2.31E−04	−3.86E+00	−3.84E+00	2.34E−02
F20	−3.32E+00	−3.26E+00	6.00E−02	−3.32E+00	−3.30E+00	8.78E−03	−3.32E+00	−3.07E+00	1.94E−01
F21	−1.01E+01	−8.88E+00	1.86E+00	−9.81E+00	−8.66E+00	7.04E−01	−1.02E+01	−5.58E+00	3.34E+00
F22	−1.03E+01	−9.88E+00	1.02E+00	−1.01E+01	−9.07E+00	5.64E−01	−1.04E+01	−5.91E+00	3.25E+00
F23	−1.05E+01	−1.01E+01	7.42E−01	−1.03E+01	−8.76E+00	8.20E−01	−1.05E+01	−5.67E+00	3.05E+00

Function	POA			KH		
	BEST	MEAN	STD	BEST	MEAN	STD
F14	9.98E−01	9.98E−01	3.41E−07	9.98E−01	1.22E+01	2.11E+01
F15	6.69E−04	1.05E−03	1.77E−04	3.08E−04	3.40E−03	6.35E−03
F16	−1.03E+00	−1.03E+00	3.05E−05	−1.03E+00	−1.03E+00	3.20E−06
F17	3.98E−01	3.98E−01	4.90E−05	3.98E−01	3.98E−01	4.20E−06
F18	3.00E+00	3.00E+00	3.46E−04	3.00E+00	3.00E+00	1.19E−08
F19	−3.86E+00	−3.86E+00	4.89E−05	−3.86E+00	−3.85E+00	1.42E−02
F20	−3.32E+00	−3.32E+00	2.70E−03	−3.32E+00	−3.28E+00	7.16E−02
F21	−1.01E+01	−9.94E+00	1.59E−01	−1.02E+01	−7.50E+00	3.37E+00
F22	−1.04E+01	−1.02E+01	9.73E−02	−1.04E+01	−7.34E+00	3.66E+00
F23	−1.05E+01	−1.04E+01	8.08E−02	−1.05E+01	−7.83E+00	3.67E+00

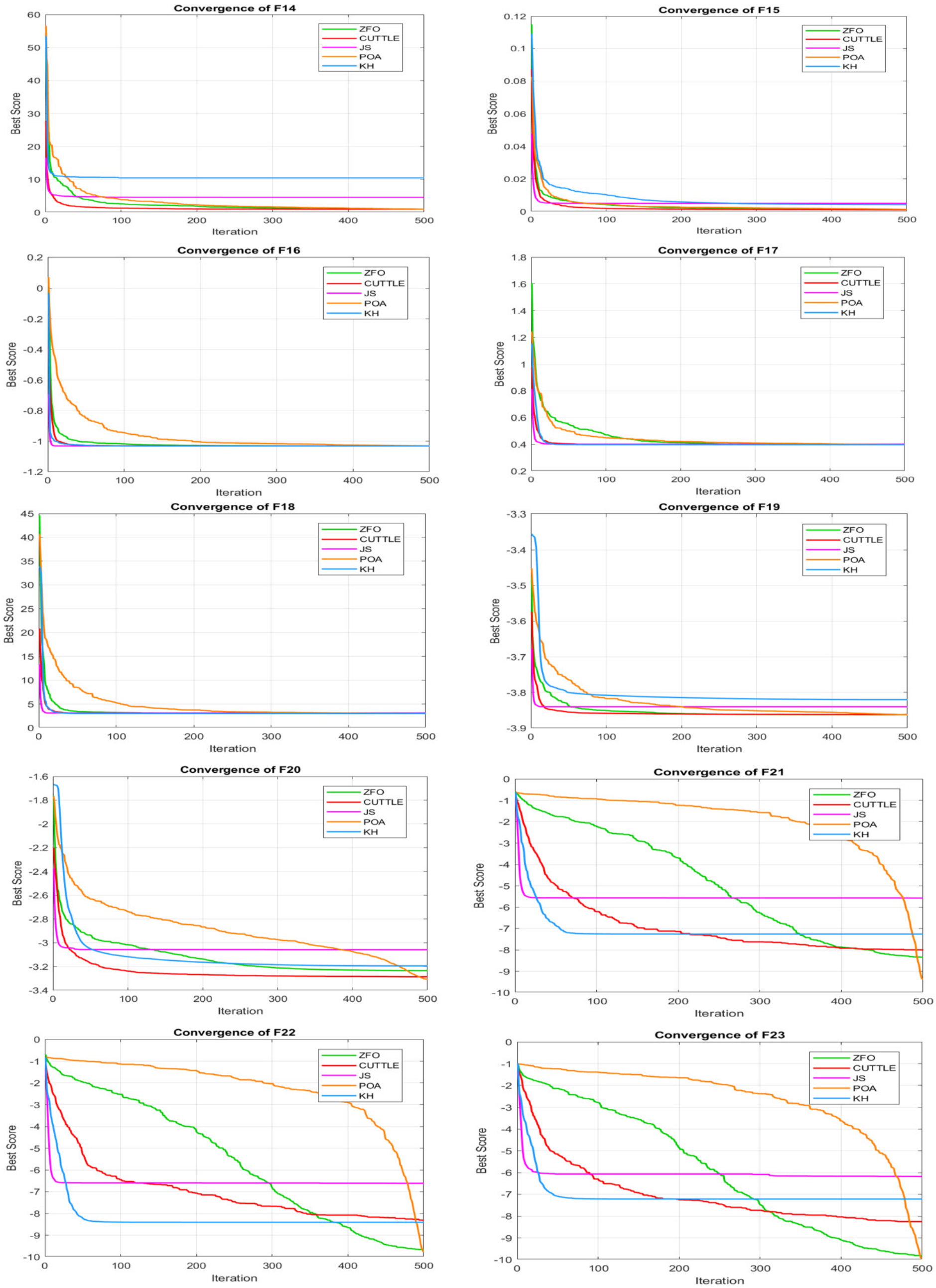

Fig. 10. Convergence curve for fixed dimension multimodal F14–F23 with 50 runs and 500 iterations.

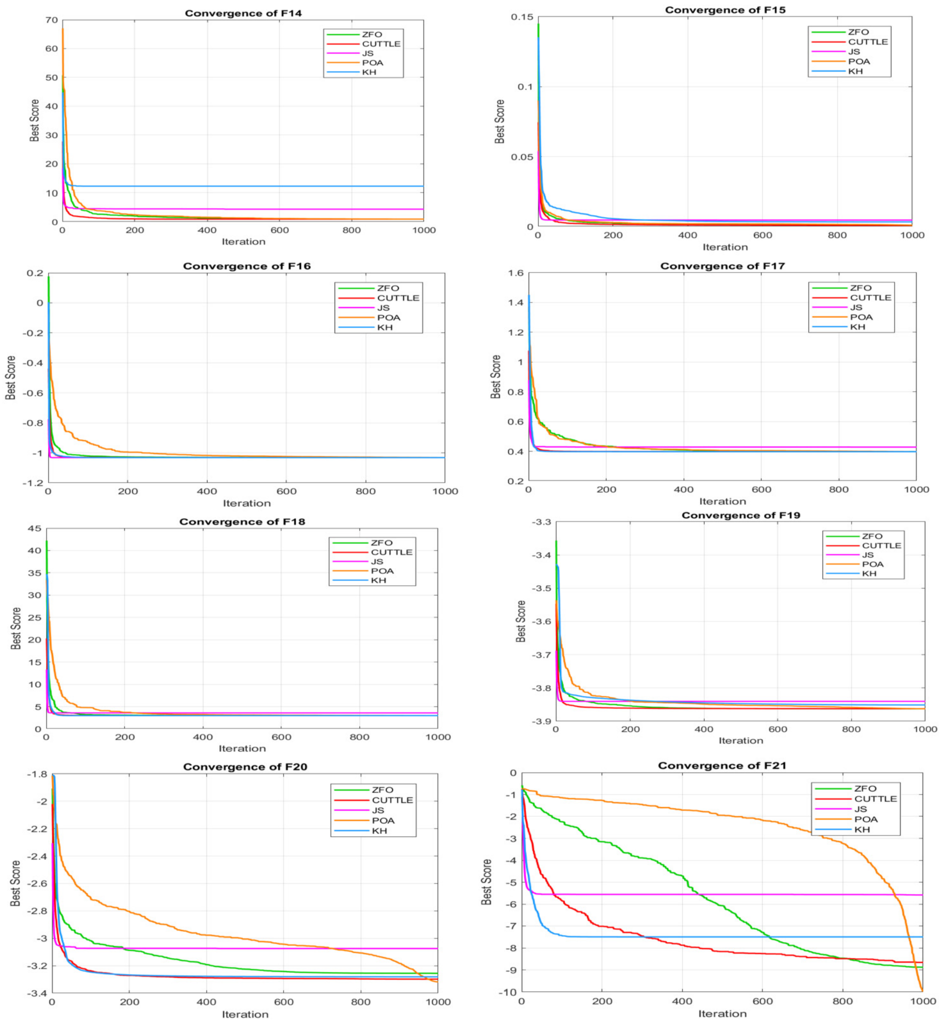

Fig. 11. Convergence curve for fixed dimension multimodal F14–F23 with 50 runs and 1000 iterations.

benchmark functions. In two specific cases, multimodal and fixed-dimension multimodal problems, the results of the Wilcoxon signed-rank test confirm that the statistically significant performance improvements exhibited by the ZFO algorithm are due to its consistently superior performance rather than due to any form of stochastic variation of performance. Table 8 summarizes the p-value result for the Wilcoxon signed-rank test for statistically significant differences of performance between the algorithms.Across all the benchmark functions a trend was observed ($p < 0.05$) on statistically significance of the ZFO performance.

3.5.3. *Overall Ranking Analysis*

This overall ranking considers all of the functions possible within a benchmark configuration, using averaged Friedman rankings to see how the different configurations stack up

Table 8
Statistical significance analysis using Wilcoxon signed-rank test for ZFO under multiple run.

	Runs:25,500 Iteration	Runs:30,500 Iteration	Runs:50,500 Iteration	Runs:25,1000 Iteration	Runs:30,1000 Iteration	Runs:50,1000 Iteration
Function	p_value	p_value	p_value	p_value	p_value	p_value
F1	1.15E−19	1.77E−23	1.58E−39	1.03E−17	2.10E−21	1.02E−35
F2	1.33E−19	2.70E−22	1.51E−39	1.10E−19	1.01E−23	5.14E−39
F3	3.04E−18	3.74E−22	2.93E−37	2.02E−18	2.66E−21	7.76E−38
F4	8.63E−17	8.49E−20	3.88E−34	7.73E−17	3.54E−20	5.62E−34
F5	6.87E−20	3.53E−24	3.19E−40	1.14E−19	2.07E−23	2.98E−39
F6	3.19E−19	2.83E−23	1.02E−39	1.52E−17	3.55E−21	4.90E−36
F7	4.72E−20	1.55E−24	1.39E−40	1.33E−19	6.45E−24	1.75E−40
F8	1.10E−17	1.51E−20	1.26E−35	1.46E−18	3.03E−22	5.26E−37
F9	4.72E−20	2.75E−24	2.23E−41	1.03E−19	1.02E−23	5.30E−40
F10	8.30E−20	7.35E−24	8.93E−41	1.10E−19	1.68E−23	3.19E−40
F11	2.77E−20	7.81E−25	1.42E−40	2.02E−20	1.11E−24	5.54E−42
F12	2.02E−20	1.11E−24	1.15E−41	1.29E−19	1.55E−23	1.81E−40
F13	4.43E−20	2.48E−24	3.04E−41	2.95E−20	5.03E−24	2.71E−39
F14	2.15E−14	1.86E−15	1.86E−30	1.12E−14	2.01E−15	2.28E−29
F15	2.63E−05	8.87E−02	8.84E−06	2.85E−03	1.84E−04	1.40E−02
F16	4.53E−14	9.88E−19	1.01E−28	3.76E−14	5.11E−18	1.53E−31
F17	5.18E−10	1.17E−10	1.66E−20	3.17E−10	2.99E−11	4.99E−19
F18	1.36E−09	9.75E−11	8.98E−22	1.57E−09	5.46E−12	2.42E−21
F19	2.81E−12	1.10E−19	4.35E−30	1.31E−14	3.63E−13	1.79E−21
F20	1.16E−08	8.60E−09	3.50E−13	3.81E−10	1.50E−14	1.01E−19
F21	5.45E−04	2.91E−03	4.28E−06	3.14E−06	6.18E−08	2.03E−12
F22	7.08E−04	3.49E−05	3.72E−11	3.64E−10	3.15E−08	2.49E−11
F23	2.54E−02	1.96E−09	2.14E−10	4.04E−07	1.09E−06	6.60E−18

against one another based on the results of the global performance evaluation. All benchmarks, as expected, performed extremely well, with ZFO finishing at the top of the list each time. The results from the experimental configurations provided conclusive evidence that the overall performance of ZFO shows high levels of robustness in varying sizes of evaluations and budgets of evaluations (or iteration budgets) while allowing for generalising across other types of optimization landscapes. Averages of the Friedman rankings and overall ranks for all algorithms with different runs and iterations show that ZFO is consistently achieving the lowest average rank and demonstrating that it has superior and robust performance overall in Table 9.

3.6. *Analysis of Phase Contributions and Performance Drivers*

The benchmark results, as described in Section 3.5.1 and 3.5.3, show that ZFO outperforms or performs competitively to CFO, JSO, POA, KHO in all three benchmark categories. However, performing well **compared to** other algorithms does not alone guarantee algorithmic quality, determining where and why ZFO performs well (which phase, which biological mechanism and which mechanism driven by parameter is responsible for which improvement of benchmark function classes) is a crucial step. The performance drivers of ZFO were analysed as follow.

Table 9
Friedman average ranks and overall ranking of algorithms under different runs and iterations.

Run	Iteration	Algorithm	AvgRank	Overall rank
25	500	ZFO	1.98E+00	**1**
		CUTTLE	3.65E+00	4
		JS	2.93E+00	3
		POA	3.78E+00	5
		KH	2.65E+00	2
30	500	ZFO	1.98E+00	**1**
		CUTTLE	3.68E+00	4
		JS	2.91E+00	3
		POA	3.79E+00	5
		KH	2.64E+00	2
50	500	ZFO	2.00E+00	**1**
		CUTTLE	3.71E+00	4
		JS	2.84E+00	3
		POA	3.80E+00	5
		KH	2.65E+00	2
25	1000	ZFO	2.10E+00	**1**
		CUTTLE	3.79E+00	5
		JS	2.98E+00	3
		POA	3.54E+00	4
		KH	2.58E+00	2
30	1000	ZFO	2.08E+00	**1**
		CUTTLE	3.76E+00	5
		JS	2.98E+00	3
		POA	3.58E+00	4
		KH	2.60E+00	2
50	1000	ZFO	2.09E+00	**1**
		CUTTLE	3.77E+00	5
		JS	2.94E+00	3
		POA	3.57E+00	4
		KH	2.63E+00	2

3.6.1. *Unimodal Benchmarks (F1–F7): Hunting-Driven Exploitation*

For unimodal F1–F7 (with single global and no local optimum) performance is driven by Hunting Phase (Section 2.3.3). Update (11) includes global (to population best) and local (to neighbours best) directional vector, modulated by time-dependent factor $\tau = \frac{t}{T}$. When small local exploration dominates, to keep population coverage wide, whilst as grows global direction dominates and it increasingly targets the optimum. This temporal dependency fits well with convex, gradient-consistent landscapes such as F1 (Sphere), F2 (Schwefel 2.22) and F4 (Rosenbrock).

This Biological Inspiration directly relates to leaderless group predation observed in zebrafish; the intensity of the attack increases with the group as it closes in on the prey target. In contrast, the cuttlefish (CFO) use their camouflage-based search operator with a static, iteration independent, pull toward the best and pufferfish (POA) with their puffing mechanism use a radius search that increases equally regardless of distance to optimum.

CFO and POA are unable to adapt directional search influence and therefore present higher mean values and a larger spread on F1–F7 due to the difference mechanism described. Fairness factor ϕ_i (5) is also directly inspired by leaderless, individual-centric, foraging behaviour: we make a direct connection to zebrafish schooling where less capable individuals are allowed greater exploratory ranges and this is achieved at each hunting step, where weak agents (lower fitness) receive disproportionately larger hunting steps than strong (higher fitness) agents due to the fairness factor ϕ_i.

3.6.2. *Multimodal Benchmarks (F8–F13): Fairness-Weighted Exploration and Stagnation Recovery*

ZFO is able to outperform on F8–F13 where, due to the multitude of local optima, the two working principles are responsible for the gain: the Fairness-Weighted Exploration Phase (Section 2.3.1) and the Stagnation-Aware Reinitialization (Section 2.3.4). The adaptable step t (4) is controlled by max and min and it remains sufficiently large to maintain spread of population in initial iterations, where the risk of being attracted by local optima is the highest, whilst the fairness coefficient ϕ_i enables individuals trapped around a suboptimal local optimum to have larger exploratory movements in proportion to the severity of their trapping and therefore be capable of being kicked out of the basins of attraction. When they fail over iterations of escape, the stagnation count up forces a complete reinitialization to another random point (12), thus forcibly ensuring diversity at the individual level. It is the combination of this two-tiered system of step amplification, which accounts for the unfairest of being trapped around poor local optima and reinitialization after a defined threshold, that is key to ZFO being able to find high-quality solutions in difficult functions such as F8 (Schwefel 2.26), F9 (Rastrigin) and F12 (Penalised).

The fairness coefficient ϕ_i serves the biologically inspired element, directly drawn from the concept of equal access foraging observed in zebrafish where the worse-off individuals exhibit a larger relative movement range than fitter individuals within the shoal, a component not present in jellyfish (JSO) which passively moves with currents and employs a dynamic, population-based state-transition between drifting and foraging or krill (KHO), which uses diffusive, random walks and has no agent-specific state of stagnation. The reinitialization based on the stagnation count up is inspired by the empirical observation of zebrafish abandoning unfavourable foraging sites and redistributing itself and this is again an agent-specific change in state, unlike the two algorithms. It is the lack of these two components which is the cause for the higher mean fitness values and variance produced by JSO and KHO for F8–F13.

3.6.3. *Fixed-Dimension Multimodal Benchmarks (F14–F23): Shoaling-Based Coordination and Diversity*

For highly non-linear, deceptive, multi-modal fixed-dimension benchmark functions F14–F23, it is the Shoaling Phase (Section 2.3.2) which predominantly drives performance. The shoaling update (7) drives each individual towards the centroid of its k closest neighbours, as opposed to towards global best, facilitating rich neighbourhood-level information exchange and allowing the population to simultaneously explore and converge on distinct promising sub-regions. Minimum distance between individuals is encouraged through the separation parameter γ (S_i in 7), preventing population collapse and keeping distinct basins of attraction explored simultaneously. The stochastic perturbation parameter δ_s (R in (7)) helps maintain local diversity in each neighbourhood and allows individuals to move through the sharp ridges found in F14 (Langermann), F16 (Six-Hump Camelback) and F18 (Goldstein-Price).

The neighbourhood-centroid based shoaling update is biologically inspired by the very real active spatial regrouping based on proximity in shoals of zebrafish, which continually reform their position based on local cues and shared gradient information. It is distinctly different from the passive, drift-driven positioning of jellyfish (JSO) and diffusion-based herding of krill (KHO). In the case of the zebrafish shoaling mechanism, γ and δ_s contribute to its ability to efficiently navigate sharp, tricky terrain which is strongly linked to its unique neighbourhood-centroid style update combined with intelligent control of both separation between individuals and internal diversity. CFO's performed inconsistent behaviour across these specific F14–F23 problems, as compared to JSO, is that neither these specific kinds of spatial search mechanics nor active local grouping behaviours are present in CFO and hence CFO's performance has the largest variance of all.

3.6.4. *Role of the Adaptive Behavioural Transition Strategy and Parameter Interactions*

The most unique parameter among the 4 comparison algorithms and ZFO is the adaptive behavioural transition strategy (13)–(15). The adaptive behavioural transition strategy assigned each agent with their own, fitness rank-dependent probability of moving in the direction of exploration, shoaling or hunting at each iteration. This strategy of assigning individual, phase-switching probabilities depended on the interplay of the normalized iteration progress $\tau = \frac{t}{T}$, which skewed the population toward hunting over exploration as optimization progressed, and the individual fairness factor ϕ_i, which modified the individual agent's personal phase probabilities based on their individual fitness rank. The higher is an agent's fitness rank, the greater its P hunt, the greater its probability of exploiting the environment intensely, whereas low rank promoted greater P explore. This effectively caused each iteration to maintain a heterogeneous population stratified by fitness level and simultaneous exploration, shoaling and hunting occur within the swarm at varying times. This corresponds to the simultaneous occurrence of these three states at all times in a school of zebrafish.

The probabilities of phase transition are predicated on an empirically proven fitness-dependent zebrafish phase switching behaviour; individuals of low fitness will behave in an explorative or dispersive manner while those with high fitness will maintain the directed hunting behaviour. ZFO is the only model that incorporates an individual, fitness-dependent phase-switching mechanism. CFO has one visibility function-based operator, JSO has a population wide, time-dependent probability, POA utilizes a threshold-based mode switching between attraction and puffing and KHO is designed around static, pre-set weights on three distinct movement components without an individual assigned state.

α_{max}, α_{min} which govern the step range of the exploratory movement, β (shoaling attraction parameter), δ_h (hunting step size) and the stagnation parameter S_{th} (controls the rate of reinitialization) are the parameter combination that characterizes the time and spatial profile of search by ZFO. When combined, these parameters determine the temporal/individual characteristic of search performed by ZFO and are confirmed to yield stable and effective search within all three categories of benchmarks without function-specific parameter adjustment, suggesting generality of the biologically-motivated design.

3.6.5. *Summary of Performance Drivers*

A summary of the phase-to-benchmark mapping as defined in this work, listing the core driver for each benchmark class, biological inspiration, underlying ZFO mechanism, dominant parameter and the relevant algorithmic shortcoming of comparison algorithms for each benchmark class.

Unimodal (F1-F7): Dominant driver Hunting Phase (11); Biological basis zebrafish leaderless time-intensified predation; Dominant parameter time-adaptive and fairness i and the reason for competitors fail is that CFO and POA utilize iteration-independent static attraction that is incapable of increasing directionality over time. Multimodal (F8–F13): Dominant driver Exploration Phase (4), Stagnation Reinitialization (12) zebrafish fairness-weighted dispersal and patch-abandonment and the Dominant parameters – $\alpha_{\max}$, $\alpha_{\min}$, ϕ_i, stagnation $S_{th} = \max(3, \mathrm{round}(0.05 \times T))$ where T is the number of iterations. It refers to the number of unsuccessful search steps allowed consecutively before reinitialization. It is scaled dynamically according to the optimization horizon and bounded from below by 3. This condition for reset is designed such that it triggers early enough for detecting premature stagnation in short time periods and sustained enough for detection of stagnation in long time periods before reset and the reason of competitor fails is that JSO and KHO are devoid of individual level stagnation detection and threshold based reinitialization. Fixed-dim Multimodal (F14–F23): Dominant driver Shoaling Phase (7); zebrafish proximity-driven fluid spatial regrouping; Dominant parameters β (attraction), γ (separation), δ_s (perturbation), k (neighbourhood size) and the reason for competitors fail is that CFO and JSO are devoid of structured neighbourhood-centroid movement including separation and perturbation.

In conclusion, the superior results demonstrated by ZFO cannot be attributed merely to favourable average benchmark statistics but are mechanistically explained by three distinct behavioural phases, each designed to address specific structural characteristics of optimization landscapes. The fairness factor, inspired by the egalitarian foraging behaviour of zebrafish and embedded across all behavioural phases and transition mechanisms acts as an overarching adaptive control mechanism at the individual-agent level. This mechanism helps overcome the limitations of CFO, JSO, POA and KHO, particularly in diversity preservation, stagnation avoidance and adaptive search regulation.

4. Conclusion And Future Work

The Zebrafish Optimization Algorithm (ZFO) is an innovative metaheuristic algorithm based on models of how zebrafish forage. The key components of ZFO are the exploration phase, where the individual fish explore all available search spaces, the shoaling phase, where the group of zebrafish gathers to assess how they can best hunt together, and the hunt phase, where they will hunt in a group. In the optimization process, ZFO utilizes both stagnation-aware behaviour adaptation and a fairness-driven movement mechanism to maintain an effective balance between global exploration and local exploitation of resources.

The performance of ZFO was tested on a wide variety of benchmark optimization problems ranging from unimodal, multimodal and fixed dimension multimodal functions. ZFO was also compared to multiple state-of-the-art metaheuristic algorithms including the Cuttlefish Optimization Algorithm (CUTTLE), Jellyfish Search Optimizer (JS), Pufferfish Optimization Algorithm (POA) and the Krill Herd Optimizer (KH). Most of the results of the comparative experiments demonstrate ZFO produces solutions of a higher or equal quality, converges faster and is more robust than the others on most benchmark functions. The qualitative evaluation of the convergence curves, search histories/trajectories and the evolution of the population fitness of the ZFO all provide continued support for the efficacy of the behaviour-based methods employed to guide the ZFO search towards potentially promising areas and away from premature convergence.

ZFO's biologically interpretable architecture and small number of control parameters make it straightforward to develop and implement, greatly reducing tuning time. The use of fairness-based movement allows less capable agents to have meaningful involvement in the search for optimal solutions, stagnation-aware reinitialization gives ZFO the ability to escape local minima. These features provide ZFO with the flexibility and efficiency needed to solve various types of continuous optimization problems.

Although ZFO has shown strong promise, it may exhibit slower convergence when initialized near optimal regions of the search space due to the method's emphasis on maintaining a diverse population as a means for exploration. Future studies will examine various adaptive parameter control techniques to further enhance exploitation in later stages of the optimization process. Combining ZFO with local search techniques or other population-based metaheuristics may further enhance convergence precision and efficiency. Expanding this framework into multi-objective, discrete and real-world constrained optimization domains may also be significant future work.

References

Abualigah, L., Diabat, A., Mirjalili, S., Abd Elaziz, M., Gandomi, A.H. (2021). The arithmetic optimization algorithm. *Computer Methods in Applied Mechanics and Engineering*, 376, 113609. https://doi.org/10.1016/j.cma.2020.113609.

Abualigah, L., Elaziz, M.A., Khasawneh, A.M., Alshinwan, M., Ibrahim, R.A., Al-Qaness, M.A., Mirjalili, S., Sumari, P., Gandomi, A.H. (2022). Meta-heuristic optimization algorithms for solving real-world mechanical engineering design problems: A comprehensive survey, applications, comparative analysis and results. *Neural Computing & Applications*, 34, 4081–4110. https://doi.org/10.1007/s00521-021-06747-4.

Al-Baik, O., Alomari, S., Alssayed, O., Gochhait, S., Leonova, I., Dutta, U., Malik, O.P., Montazeri, Z., Dehghani, M. (2024). Pufferfish optimization algorithm: a new bio-inspired metaheuristic algorithm for solving optimization problems. *Biomimetics*, 9(2), 65. https://doi.org/10.3390/biomimetics9020065.

Beheshti, Z., Shamsuddin, S.M. (2013). A review of population-based meta-heuristic algorithms. *International Journal of Advances in Soft Computing and its Applications*, 5(1), 1–35. Available at: https://www.researchgate.net/publication/270750820_A_review_of_population-based_meta-SSheuristic_algorithm.

Chou, J.-S., Truong, D.-N. (2021). A novel metaheuristic optimizer inspired by behavior of jellyfish in ocean. *Applied Mathematics and Computation*, 389, 125535. https://doi.org/10.1016/j.amc.2020.125535.

Civicioglu, P. (2013). Backtracking search optimization algorithm for numerical optimization problems. *Applied Mathematics and Computation*, 219(15), 8121–8144. https://doi.org/10.1016/j.amc.2013.02.017.

Gad, A.G. (2022). Particle swarm optimization algorithm and its applications: a systematic review. *Archives of Computational Methods in Engineering*, 29(5), 2531–2561. https://doi.org/10.1007/s11831-021-09694-4.

Gandomi, A.H., Alavi, A.H. (2012). Krill herd: a new bio-inspired optimization algorithm. *Communications in Nonlinear Science and Numerical Simulation*, 17(12), 4831–4845. https://doi.org/10.1016/j.cnsns.2012.05.010.

Glover, F.W., Laguna, M. (1997). *Tabu Search*. Springer Science & Business Media, New York, NY, USA. https://doi.org/10.1007/978-1-4615-6089-0. (softcover reprint 1998).

Hashim, F.A., Mostafa, R.R., Hussien, A.G., Mirjalili, S., Sallam, K.M. (2023). Fick's law algorithm: a physical law-based algorithm for numerical optimization. *Knowledge-Based Systems*, 260, 110146. https://doi.org/10.1016/j.knosys.2022.110146.

Holland, J.H. (1992). *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence*. MIT Press, Cambridge, MA, USA. https://doi.org/10.7551/mitpress/1090.001.0001.

Huang, B. (2025a). Metaheuristic-based supply chain network optimization and inventory management using ant colony algorithm. *Informatica*, 49(7). https://doi.org/10.31449/inf.v49i7.6760.

Huang, B. (2025b). Hybrid particle swarm optimization and Q-learning for airport parking space allocation and scheduling. *Informatica*, 49(31). https://doi.org/10.31449/inf.v49i31.8352.

Kalpanarani, K., Hannah Grace, G. (2025). SMCFO: a novel cuttlefish optimization algorithm enhanced by simplex method for data clustering. *Frontiers in Artificial Intelligence*, 8, 1677059. https://doi.org/10.3389/frai.2025.1677059.

Kennedy, J., Eberhart, R. (1995). Particle swarm optimization. In: *Proceedings of ICNN'95-International Conference on Neural Networks*, Vol. 4, pp. 1942–1948. https://doi.org/10.1109/ICNN.1995.488968.

Kirkpatrick, S., Gelatt, C.D. Jr., Vecchi, M.P. (1983). Optimization by simulated annealing. *Science*, 220(4598), 671–680. https://doi.org/10.1126/science.220.4598.671.

Lourenço, H.R., Martin, O.C., Stützle, T. (2003). Iterated local search. In: Glover, F., Kochenberger, G.A. (Eds.), *Handbook of Metaheuristics*. International Series in Operations Research & Management Science, Vol. 57. Springer, Boston, MA, USA. https://doi.org/10.1007/0-306-48056-5_11.

Mladenović, N., Hansen, P. (1997). Variable neighbourhood search. *Computers and Operations Research*, 24(11), 1097–1100. https://doi.org/10.1016/S0305-0548(97)00031-2.

Mirjalili, S., Mirjalili, S.M., Lewis, A. (2014). Grey wolf optimizer. *Advances in Engineering Software*, 69, 46–61. https://doi.org/10.1016/j.advengsoft.2013.12.007.

Mirjalili, S., Lewis, A. (2016). The whale optimization algorithm. *Advances in Engineering Software*, 95, 51–67. https://doi.org/10.1016/j.advengsoft.2016.01.008.

Osaba, E., Villar-Rodríguez, E., Ser, J.D., Nebro, A.J., Molina, D., LaTorre, A., Suganthan, P.N., Coello, C.A.C., Herrera, F. (2021). A tutorial on the design, experimentation and application of metaheuristic algorithms to real-world optimization problems. *Swarm and Evolutionary Computation*, 64, 100888. https://doi.org/10.1016/j.swevo.2021.100888.

Pan, J.S., Hu, P., Snášel, V., Chu, S.-C. (2023). A survey on binary metaheuristic algorithms and their engineering applications. *Artificial Intelligence Review*, 56, 6101–6167. https://doi.org/10.1007/s10462-022-10328-9.

Price, K., Storn, R.M., Lampinen, J.A. (2006). *Differential Evolution: A Practical Approach to Global Optimization*. Springer Science & Business Media. https://doi.org/10.1007/3-540-31306-0.

Rao, R.V., Savsani, V.J., Vakharia, D. (2011). Teaching-learning-based optimization: a novel method for constrained mechanical design optimization problems. *Computer-Aided Design*, 43(3), 303–315. 303–315.https://doi.org/10.1016/j.cad.2010.12.015.

Rao, R. (2016). Jaya: a simple and new optimization algorithm for solving constrained and unconstrained optimization problems. *International Journal of Industrial Engineering Computations*, 7(1), 19–34. https://doi.org/10.5267/j.ijiec.2015.8.004.

Rashedi, E., Nezamabadi-Pour, H., Saryazdi, S. (2009). GSA: a gravitational search algorithm. *Information Sciences*, 179(13), 2232–2248. https://doi.org/10.1016/j.ins.2009.03.004.

Song, P.C., Chu, S.C., Pan, J.S., Yang, H.M. (2022). Simplified phasmatodea population evolution algorithm for optimization. *Complex & Intelligent Systems*, 8(4), 2749–2767. https://doi.org/10.1007/s40747-021-00402-0.

Sullivan, K.A., Jacobson, S.H. (2001). A convergence analysis of generalized hill climbing algorithms. *IEEE Transactions on Automatic Control*, 46(8), 1288–1293. https://doi.org/10.1109/9.940936.

Surjanovic, S., Bingham, D. (2013). *Virtual Library of Simulation Experiments: Test Functions and Datasets*. Simon Fraser University, Burnaby, Canada. https://www.sfu.ca/~ssurjano/optimization.html.

Suruthi, M., Ganesh, N. (2026). Zebrafish Optimization Algorithm source code repository. *GitHub*. https://github.com/suruthi-m/Zebr-Fish-Optimization.git.

Talbi, E.-G. (2009). *Metaheuristics: From Design to Implementation*. John Wiley & Sons, Hoboken, NJ, USA. https://doi.org/10.1002/9780470496916.

Vijaya, C., Srinivasan, P. (2024). Multi-objective meta-heuristic technique for energy efficient virtual machine placement in cloud data centers. *Informatica*, 48(6). https://doi.org/10.31449/inf.v48i6.5263.

Wu, X. (2026). A hybrid approach of metaheuristic algorithms and GMDH to predict source code testability. *Informatica*, 50(10). https://doi.org/10.31449/inf.v50i10.10680.

Zhan, Z.H., Shi, L., Tan, K.C., Zhang, J. (2022). A survey on evolutionary computation for complex continuous optimization. *Artificial Intelligence Review*, 55, 59–110. https://doi.org/10.1007/s10462-021-10042-y.

Zhao, W., Wang, L., Zhang, Z. (2019). Supply-demand-based optimization: a novel economics-inspired algorithm for global optimization. *IEEE Access*, 7, 73182–73206. https://doi.org/10.1109/ACCESS.2019.2918753.

M. Suruthi is a research scholar in the School of Computer Science and Engineering at Vellore Institute of Technology, Chennai, India. Her research spans towards mathematical optimization, engineering optimization, operations research, machine learning, AI-based optimization and computational methods, reflecting both theoretical rigour and practical applicability. She works across classical linear and nonlinear programming to modern metaheuristic and swarm intelligence techniques, addressing complex real-world challenges through innovative algorithmic approaches. Her contributions to nature inspired optimization bridge traditional mathematical methods with emerging intelligent computing paradigms. As an active and prolific researcher, she continues to advance multi-domain optimization research within the broader academic community.

N. Ganesh brings a wealth of experience to his role as a professor in the School of Computer Science and Engineering, Vellore Institute of Technology, Chennai, India. He is recognized as one of the Top 2% Scientists by Elsevier and Stanford University, USA. With a career spanning more than two decades in teaching, training and research, he has established himself as an authority in this field. His research interests are diverse and forward thinking, encompassing software engineering, agile software development, prediction and optimization techniques, deep learning, image processing and data analytics.