Enhanced Pied Kingfisher Optimization Algorithm with Hovering Scouts and Foraging Flocks Mechanisms

Aarhus Dela Cruz
Mathematics Department-College of Science,
Bulacan State University,
Bulacan, Philippines 3000

***Abstract*—Population-based metaheuristic algorithms are widely used for solving nonlinear, nonconvex, constrained, and high-dimensional optimization problems. However, many swarm-based optimizers still suffer from premature convergence, loss of population diversity, and weak exploitation of multiple promising regions. To address these limitations, this study proposes the Hovering Scouts and Foraging Flocks Pied Kingfisher Optimizer (HSFFPKO), an enhanced variant of the Pied Kingfisher Optimizer (PKO). The proposed method introduces two complementary mechanisms. The Hovering Scouts mechanism applies scale-aware Gaussian probing with weak global-best guidance to restore local diversity and reduce stagnation, while the Foraging Flocks mechanism organizes the population into temporary subgroups guided by leader–centroid targets under a shrinking search radius. The performance of HSFFPKO was evaluated on the CEC 2017 benchmark suite and nine constrained engineering design problems. In the ablation study at $D = 10$, HSFFPKO achieved the best average rank of 1.67 and obtained 20 wins out of 30 benchmark functions. In the scalability analysis, HSFFPKO remained the best-ranked PKO variant at $D = 30$, $D = 50$, and $D = 100$, with average ranks of 1.37, 1.33, and 1.27, respectively. In the broad comparison with recent optimizers at $D = 30$, HSFFPKO obtained the best overall average rank of 2.60 and the highest number of function wins, with 11 wins out of 30 functions. The Nemenyi post-hoc test showed that HSFFPKO was statistically comparable with the strongest competitors, including BWSMA and GPSOM, while significantly outperforming several other methods. Engineering results further confirmed that HSFFPKO is highly competitive for continuous constrained design problems, although its performance was weaker on the discrete gear-train problem. These results indicate that HSFFPKO is a scalable and competitive PKO variant for continuous numerical and engineering optimization.**

***Keywords*—Exploration-exploitation balance; metaheuristic optimization; numerical optimization; Pied Kingfisher Optimizer; swarm intelligence**

I. Introduction

Optimization problems in computer science, machine learning, operations research, and engineering design are often nonlinear, nonconvex, constrained, and high-dimensional. In many practical cases, gradient information is unavailable, expensive to compute, or unreliable because the objective function is noisy, discontinuous, or treated as a black box. For this reason, population-based metaheuristic algorithms have become widely used alternatives for solving difficult real-parameter optimization problems. Classical and recent swarm-based methods, including Particle Swarm Optimization (PSO), Grey Wolf Optimizer (GWO), Whale Optimization Algorithm (WOA), Salp Swarm Algorithm (SSA/SSO), Bat Algorithm (BA), and other animal-inspired optimizers, have demonstrated strong potential because they combine stochastic exploration with iterative improvement of candidate solutions [1], [2], [3], [4], [5], [6], [7].

Despite these advantages, the design of an effective metaheuristic remains challenging. The no-free-lunch theorem indicates that no single optimization algorithm can be universally superior over all possible problem classes [8]. Therefore, algorithmic improvement usually depends on identifying search weaknesses in a particular optimizer and introducing mechanisms that better control exploration, exploitation, diversity preservation, and convergence stability. In swarm intelligence, excessive exploration may delay convergence, whereas excessive exploitation may cause premature convergence to local optima. This trade-off becomes more difficult in hybrid and composition functions, high-dimensional search spaces, and constrained engineering design problems, where the landscape may contain variable interaction, deceptive basins, and narrow feasible regions [9], [10], [11].

Recent studies show that improving a metaheuristic often requires more than simply changing its biological metaphor. Effective improvements usually introduce specific search mechanisms such as adaptive perturbation, random-walk modification, subgroup cooperation, diversity recovery, leader-guided movement, or local-search refinement. For example, the original Bat Algorithm was later modified through torus-walk-based movement to improve local search and high-dimensional performance [12]. A modified controlled Bat Algorithm further used torus-walk concepts to improve diversity and convergence in numerical optimization and neural-network training [13]. Similarly, an improved Bat Algorithm was applied to neural-network optimization for data classification to reduce local-minimum trapping and improve convergence accuracy [14]. These developments highlight an important trend in recent metaheuristic research: successful variants are usually justified by the search behavior they add, not only by the natural behavior they imitate.

The Pied Kingfisher Optimizer (PKO) is a recent bio-inspired algorithm that models the hunting behavior of the pied kingfisher, *Ceryle rudis*, through perching, hovering, diving, and commensal association [15]. These behaviors provide a useful balance between observation, exploratory movement,

elite-guided exploitation, and cooperative perturbation. However, the baseline PKO can still suffer from two important limitations. First, population diversity may decrease too early when many agents are attracted toward similar regions of the search space. Second, the algorithm does not explicitly organize the population into cooperative subgroups that can search several promising basins in parallel. These limitations can reduce robustness on multimodal, hybrid, and composition landscapes, especially when the search space becomes more complex.

To address these gaps, this study proposes the Hovering Scouts and Foraging Flocks Pied Kingfisher Optimizer (HSFFPKO), an enhanced PKO variant with two complementary mechanisms. The first mechanism, Hovering Scouts (HS), periodically selects a subset of agents and applies scale-aware Gaussian perturbations with a weak directional bias toward the current global best solution. This mechanism is intended to restore individual-level diversity and help agents escape stagnation near local optima. The second mechanism, Foraging Flocks (FF), partitions the population into temporary subgroups and moves each member toward a blended target formed from the flock leader and flock centroid under a shrinking radius. This mechanism is intended to promote coordinated basin-level exploitation while preserving multiple search directions. Thus, HS operates mainly as an individual diversity-recovery mechanism, whereas FF provides group-level coordination and exploitation.

The main contributions of this study are summarized as follows:

- A new PKO variant, called HSFFPKO, is proposed by integrating Hovering Scouts and Foraging Flocks into the baseline PKO search cycle.
- The Hovering Scouts mechanism introduces scale-aware Gaussian probing with weak global-best guidance to improve local diversity and reduce premature convergence.
- The Foraging Flocks mechanism introduces leader–centroid subgroup search to improve coordinated exploitation around promising regions while maintaining population diversity.
- Ablation and scalability analyses are conducted to isolate the individual and combined effects of HS and FF against the baseline PKO at $D = 10$, $D = 30$, $D = 50$, and $D = 100$.
- The proposed method is evaluated on the CEC 2017 benchmark suite and on constrained engineering design problems to assess numerical accuracy, robustness, and practical applicability.
- Nonparametric statistical tests are used to support a cautious comparison with recent metaheuristic optimizers, emphasizing both performance gains and cases where differences from close competitors are not statistically significant.

The remainder of this study is organized as follows: Section II reviews related work on metaheuristic optimization, PKO, and diversity-preserving search mechanisms. Section III presents the baseline PKO and its main search operators. Section IV describes the proposed HSFFPKO, including the Hovering Scouts and Foraging Flocks mechanisms. Section V explains the experimental protocol, benchmark functions, engineering design problems, and statistical procedures. Section VI reports and discusses the ablation results, comparative performance, historical search behavior, and engineering optimization results. Finally, Section VII concludes the study and outlines limitations and future research directions.

II. Related Work

Metaheuristic optimization has developed rapidly because many real-world problems are nonlinear, nonconvex, constrained, and difficult to solve using exact or gradient-based techniques. In such problems, the objective function may be expensive, discontinuous, noisy, or unavailable in closed form. Population-based metaheuristics address this difficulty by maintaining several candidate solutions and improving them through stochastic search rules. However, according to the no-free-lunch theorem, no single optimizer can perform best on all possible optimization problems [8]. This motivates the continuous development of new algorithms and the improvement of existing ones through more effective exploration, exploitation, diversity preservation, and convergence-control mechanisms.

A. Swarm-Based and Bio-Inspired Optimization Algorithms

Swarm intelligence algorithms are among the most widely used metaheuristics because they are simple to implement, derivative-free, and adaptable to many optimization problems. Particle Swarm Optimization (PSO) models social learning among particles and updates candidate solutions using individual and global best information [1]. Grey Wolf Optimizer (GWO) simulates the leadership hierarchy and cooperative hunting process of grey wolves, where alpha, beta, and delta wolves guide the rest of the pack toward promising regions [2]. Whale Optimization Algorithm (WOA) models the bubble-net hunting behavior of humpback whales and alternates between encircling, spiral movement, and global exploration [3]. Salp Swarm Algorithm (SSA), also referred to in some studies as Salp Swarm Optimization (SSO), imitates the chain-like movement of salps and uses leader–follower dynamics to guide the population [4]. The Bat Algorithm (BA) is another important bio-inspired method that uses echolocation-inspired frequency, loudness, and pulse-rate control to balance local and global search [5].

Although these algorithms have been successfully applied to many benchmark and engineering problems, they share common limitations. PSO may converge prematurely when particles rapidly collapse toward the current best position. GWO and WOA can lose population diversity in later iterations because the search becomes increasingly leader-dominated. SSA/SSO may suffer from weak diversity when the chain structure becomes too dependent on the leader. BA may also become trapped in local optima if its local-search process is not sufficiently diversified. These limitations are frequently observed in multimodal, hybrid, composition, and high-dimensional functions, where the search landscape contains many deceptive basins and strong variable interactions [9], [10], [11].

B. Recent Improvements in Metaheuristic Search Mechanisms

Recent metaheuristic research shows that performance improvement usually comes from adding concrete search mechanisms rather than merely proposing a new natural metaphor. Common improvement strategies include adaptive parameter control, chaotic or opposition-based initialization, Gaussian or Levy perturbation, restart strategies, subgroup search, multi-population cooperation, leader–centroid attraction, mutation, and local-search hybridization. These strategies are designed to address two central issues: maintaining enough diversity to avoid premature convergence and increasing exploitation pressure once promising regions have been identified.

Several recent Bat Algorithm variants illustrate this trend. Bangyal *et al.* proposed a modified Bat Algorithm with torus walk to improve global optimization performance, particularly by enhancing the movement strategy and helping the algorithm avoid premature convergence in difficult landscapes [12]. A modified controlled Bat Algorithm later used torus-walk concepts to improve diversity and convergence in numerical optimization and neural-network training [13]. Similarly, an improved Bat Algorithm was used to optimize neural-network training for data classification, showing that algorithmic refinement can improve convergence speed and reduce local-minimum trapping in machine-learning applications [14]. These works are relevant to the present study because the proposed HSFFPKO also improves a baseline bio-inspired optimizer by adding explicit mechanisms for diversity recovery and guided exploitation.

Other recent optimizers also use multi-strategy or population-structuring mechanisms. Manta Ray Foraging Optimization (MRFO) uses chain, cyclone, and somersault foraging behaviors to support exploration and exploitation [16]. Mating-based MRFO further incorporates a mating mechanism to create higher-quality offspring and improve the population during search [17]. Outpost-based Multi-population WOA (OM-WOA) improves WOA by combining outpost individuals with multiple subpopulations, thereby increasing search diversity and reducing the probability of stagnation [18]. Best–Worst Slime Mould Algorithm (BWSMA) improves the original SMA using adaptive greedy selection, best–worst management, and stagnant replacement mechanisms [19]. Group-based PSO with multiple strategies (GPSOM) divides the population into different sub-swarms responsible for exploration, exploitation, and balance between the two phases [20]. Hierarchical Multi-Step GWO (HMS-GWO) improves the standard GWO by assigning different search steps to different wolf roles, which enhances exploration, exploitation, and diversity preservation [21]. These methods show that modern swarm optimization increasingly relies on structured population management, adaptive search rules, and diversity-preserving mechanisms.

C. Pied Kingfisher Optimizer and its Limitations

The Pied Kingfisher Optimizer (PKO) is a recent swarm-based metaheuristic inspired by the hunting behavior and symbiotic relationships of the pied kingfisher, *Ceryle rudis* [15]. PKO models three main search behaviors: perching or hovering for prey, diving for prey, and commensal association. The perching and hovering behaviors support exploration and diversification, the diving behavior strengthens exploitation around promising solutions, and the commensal association introduces cooperative perturbation among agents. Because of this structure, PKO provides a useful balance between global search and local refinement.

However, the baseline PKO still has limitations that motivate further improvement. First, the population can lose diversity when too many individuals are attracted to similar promising regions. Second, the original algorithm does not explicitly maintain several coordinated subgroups that can explore different basins in parallel. Third, although the commensal association operator can introduce cooperative perturbation, it does not directly distinguish between individual-level diversity recovery and group-level basin exploitation. These limitations may reduce robustness on complex functions, especially hybrid and composition functions where the global optimum may be hidden among several deceptive local basins.

D. Position of the Proposed HSFFPKO

The proposed HSFFPKO is positioned as a mechanism-based improvement of PKO. It does not replace the baseline PKO framework; instead, it augments the original search cycle with two complementary mechanisms. Hovering Scouts (HS) introduce self-scaled Gaussian perturbations with weak global-best guidance. This mechanism is related to diversity-recovery and perturbation-based improvement strategies because it allows selected agents to probe nearby regions even after the population has started to converge. Foraging Flocks (FF) introduce temporary subgroup search based on a leader–centroid target and a shrinking radius. This mechanism is related to multi-population and group-based search because it enables several local neighborhoods to be exploited in parallel.

Compared with existing improvement strategies, HSFF-PKO has a distinct design purpose. HS acts at the individual level by reactivating selected agents and reducing local stagnation. FF acts at the subgroup level by coordinating movement around promising local leaders and flock centroids. Therefore, the two mechanisms are intended to be complementary rather than redundant. HS improves local diversity and escape capability, while FF improves coordinated exploitation and basin-level refinement. This two-scale structure differentiates HSFFPKO from single-mechanism variants that rely only on perturbation, only on leader attraction, or only on multi-population division.

Table I summarizes representative related methods and clarifies the position of HSFFPKO with respect to the literature.

Overall, the literature indicates three important research trends. First, recent metaheuristic variants increasingly focus on diversity preservation and stagnation avoidance rather than only faster convergence. Second, subgroup and multi-population strategies are becoming common because they allow different regions of the search space to be explored or exploited simultaneously. Third, perturbation-based mechanisms remain useful when they are controlled and integrated into the main search cycle. HSFFPKO follows these trends by combining scale-aware scout perturbation with temporary flock-based exploitation. This combination addresses the main weaknesses of baseline PKO and provides a clear justification for the proposed algorithm.

TABLE I. COMPARISON OF REPRESENTATIVE METAHEURISTIC ALGORITHMS AND THEIR RELEVANCE TO HSFFPKO

Algorithm	Inspiration	Main mechanism	Limitation / motivation	Relation to HSFFPKO
PSO [1]	Bird flocking	Individual and global best learning	May prematurely converge when particles collapse to the same region	HSFFPKO adds scout perturbation and flock search to preserve diversity
GWO [2]	Wolf hierarchy	Alpha, beta, and delta guided hunting	Leader dependence may reduce diversity	Uses local flock leaders instead of one global hierarchy
WOA [3]	Whale hunting	Encircling, spiral motion, and random search	Can stagnate on high-dimensional or deceptive functions	Uses shrinking flock radius and scout probes for refinement and escape
SSA/SSO [4]	Salp chain	Leader–follower movement	Search may become overly leader-dependent	Forms several temporary flocks instead of one chain
BA [5]	Bat echolocation	Frequency, loudness, pulse rate, and local walk	May require stronger diversity control	HS uses controlled Gaussian probing for local diversity
TW-BA [12]	Enhanced BA	Torus-walk movement	Improves BA for difficult global optimization	Supports modifying baseline algorithms through explicit search operators
MCBA [13]	Controlled BA	Torus-walk diversity and convergence control	Addresses premature convergence and weak local search	Motivates controlled perturbation mechanisms such as HS
IBANN [14]	Improved BA	BA-based neural-network weight optimization	Reduces local-minimum trapping in ANN training	Shows practical value of refining metaheuristics for applied tasks
MRFO [16]	Manta ray foraging	Chain, cyclone, and somersault foraging	Depends on balance among foraging phases	HSFFPKO separates individual scouting and group foraging
MMRFO [17]	Manta ray mating	Mating-based offspring generation	Improves population quality	HSFFPKO improves quality through scouts and leader–centroid flocks
OMWOA [18]	Whale with outposts	Outpost and multi-population search	Designed for high-dimensional search and stagnation reduction	Similar to FF, but implemented inside the PKO framework
BWSMA [19]	Slime mould	Best–worst management and stagnant replacement	Improves convergence and avoids local optima	Strong comparator because it explicitly manages stagnation
GPSOM [20]	Group-based PSO	Sub-swarms for exploration, exploitation, and balance	Addresses high-dimensional complexity and diversity loss	Closely related to FF, but HSFFPKO also adds scout perturbation
HMS-GWO [21]	Hierarchical wolves	Multi-step role-based search	Addresses premature convergence and parameter sensitivity	HSFFPKO differentiates scouts and flocks instead of wolf roles
QSO [7]	Quokka behavior	Environment-driven search and exploitation mixing	Requires effective exploration–exploitation control	Serves as a recent animal-inspired comparator
RCCO [6]	Red-crowned crane	Foraging, roosting, and dance strategies	Performance depends on strategy switching	Serves as a recent biomimetic comparator
PKO [15]	Pied kingfisher	Perching/hovering, diving, and commensalism	May lose diversity and lacks explicit subgroup search	Direct baseline improved by HSFFPKO

III. BASELINE PKO

Since the proposed HSFFPKO is developed as an extension of the Pied Kingfisher Optimizer (PKO), this section briefly reviews the baseline PKO search process [15]. Consider a minimization problem:

$$\min_{X\in\Omega} f(X), \qquad \Omega = [\mathrm{LB}, \mathrm{UB}] \subset \mathbb{R}^D, \tag{1}$$

where, f is the objective function, D is the number of decision variables, and LB and UB are the lower and upper bound vectors. PKO maintains a population of N candidate solutions, called kingfishers. The position of the i-th kingfisher at iteration t is denoted by X_i^t, and the best solution found so far is denoted by X^*.

A. Population Initialization

The initial population is generated uniformly within the feasible search domain:

$$X_{i,d}^0 = \ell_d + r_{i,d}(u_d - \ell_d), \quad i = 1, \ldots, N, \quad d = 1, \ldots, D, \tag{2}$$

where, $r_{i,d} \sim \mathcal{U}(0,1)$, and ℓ_d and u_d are the lower and upper bounds of the d-th variable, respectively. Uniform initialization provides an unbiased starting distribution over the feasible domain, which is important because the initial population can influence both convergence speed and the ability of the optimizer to avoid poor local optima [11].

B. Exploration Through Perching and Hovering

PKO alternates between exploratory and exploitative behaviors inspired by the hunting strategy of the pied kingfisher. During the exploration phase, an agent updates its position using a peer-guided displacement:

$$\widetilde{X}_i^t = X_i^t + \alpha_i^t \odot S_i^t \odot (X_j^t - X_i^t), \quad j \neq i, \tag{3}$$

where, X_j^t is a randomly selected peer, α_i^t is a stochastic scaling vector, $\odot$ denotes element-wise multiplication, and S_i^t is a time-dependent step factor. In PKO, this exploratory movement is implemented through perching and hovering modes. Perching encourages broader directional probing, whereas hovering provides a more localized peer-guided search around the current region.

A representative hovering step factor can be expressed as:

$$S_i^t = B_{i,j}^t - \left(\frac{t}{T}\right)^{1/BF}, \tag{4}$$

where, $B_{i,j}^t$ is a fitness-dependent beating-rate term, BF is the beating factor, and T is the maximum number of iterations. This term decreases the exploratory amplitude as the search progresses, allowing the population to gradually move from broad exploration toward more refined search.

C. Exploitation Through Diving

The diving phase is the main exploitation mechanism of PKO. It moves an agent toward a perturbed elite point associated with the current global best solution. The elite attractor is defined as:

$$b_i^t = X_i^t + [o(t)]^2 \zeta_i X^*, \tag{5}$$

where, $\zeta_i \sim \mathcal{N}(0,1)$ and $o(t)$ is a decreasing control function:

$$o(t) = \exp\left[-\left(\frac{t}{T}\right)^2\right]. \tag{6}$$

The diving update is then given by:

$$\widetilde{X}_i^t = X_i^t + H_i^t \, o(t) \, \alpha_i^t \odot (b_i^t - X^*), \tag{7}$$

where, H_i^t is a hunting-ability coefficient derived from the relative quality of the current agent. This mechanism increases exploitation pressure around the best solution while preserving stochastic perturbation through α_i^t, ζ_i, and the elite attractor b_i^t.

D. Commensal Association

After the main exploratory or exploitative movement, PKO applies a commensal association operator to maintain population diversity. For each agent, a cooperative trial position is generated with probability $PE(t)$:

$$\widetilde{X}_i^t = X_m^t + o(t)\alpha_i^t \odot |X_i^t - X_n^t|, \tag{8}$$

where, X_m^t and X_n^t are randomly selected peers. The probability of applying this operator decreases linearly as:

$$PE(t) = PE_{\max} - (PE_{\max} - PE_{\min})\frac{t}{T}. \tag{9}$$

Thus, commensal association is more active in the early and middle stages of the search and gradually weakens as the algorithm approaches the final exploitation phase.

All candidate solutions generated by PKO are clipped to the feasible bounds. A greedy selection rule is then applied:

$$X_i^{t+1} = \begin{cases} \widetilde{X}_i^t, & \text{if } f(\widetilde{X}_i^t) < f(X_i^t), \\ X_i^t, & \text{otherwise.} \end{cases} \tag{10}$$

This acceptance rule ensures that each accepted movement improves the current solution of the corresponding agent.

The baseline PKO flow in Algorithm 1 shows where the original exploration, exploitation, and cooperative adjustment operators occur. The proposed HSFFPKO preserves this structure but inserts two additional mechanisms, Hovering Scouts and Foraging Flocks, to strengthen individual-level diversity recovery and group-level basin exploitation.

Algorithm 1 Baseline Pied Kingfisher Optimizer

Require: Population size N, maximum iterations T, bounds LB, UB, objective function f
Ensure: Best solution X^*
Initialize N kingfishers uniformly using (2)
Evaluate all agents and set the best solution X^*
for $t = 1$ to T **do**
  Compute the control factor $o(t)$
  for each agent $i = 1, \ldots, N$ **do**
    if $\mathrm{rand} < 0.8$ **then**
      Select a random peer $j \neq i$
      Generate an exploratory trial solution using perching or hovering
    else
      Generate an exploitative trial solution using diving
    Clip the trial solution to [LB, UB]
    Apply greedy selection and update X_i^t if improved
  for each agent $i = 1, \ldots, N$ **do**
    if $\mathrm{rand} < PE(t)$ **then**
      Generate a commensal association trial solution
      Clip the trial solution to [LB, UB]
      Apply greedy selection and update X_i^t if improved
  Update the global best solution X^*
return X^*

IV. The Proposed HSFFPKO

The proposed Hovering Scouts and Foraging Flocks Pied Kingfisher Optimizer (HSFFPKO) is designed to address two limitations of the baseline PKO. First, the population may lose diversity when many agents move toward similar promising regions. Second, the original PKO does not explicitly organize the population into cooperative subgroups that can exploit several promising basins in parallel. To address these issues, HSFFPKO inserts two additional mechanisms into the baseline PKO cycle: Hovering Scouts (HS) and Foraging Flocks (FF). HS acts at the individual level by perturbing selected agents to recover local diversity, whereas FF acts at the subgroup level by coordinating movement around local flock leaders and flock centroids.

A. Hovering Scouts

The Hovering Scouts mechanism is motivated by the idea that a small number of agents can perform exploratory probing while the rest of the population continues the regular PKO search cycle. At iteration t, let $S_t \subseteq \{1, \ldots, N\}$ denote the set of selected scouts:

$$|S_t| = \max(1, \lfloor \gamma N \rfloor), \tag{11}$$

where, $\gamma \in (0,1)$ is the scout rate. Thus, at least one scout is activated at each scouting event, and the number of scouts increases proportionally with the population size.

For each selected scout $i \in S_t$, HS first generates a self-scaled Gaussian probe:

$$Y_i^t = X_i^t + X_i^t \odot G_i^t, \qquad G_i^t \sim \mathcal{N}(0, I_D), \tag{12}$$

where, G_i^t is a D-dimensional Gaussian random vector and $\odot$ denotes element-wise multiplication. This operation makes the perturbation scale-aware because variables with larger magnitudes receive proportionally larger exploratory variation. To prevent the scout from becoming purely random, a weak best-referenced directional component is then added:

$$\widetilde{X}_i^t = Y_i^t + \eta\,\sigma_i^t \odot (X^* - Y_i^t), \tag{13}$$

where, $\eta > 0$ is the scout bias coefficient, X^* is the current global best solution, and $\sigma_i^t \in \{-1, +1\}^D$ is a random sign vector. The sign vector randomizes the coordinate-wise direction of the best-referenced movement, while η controls its magnitude.

The candidate scout position is clipped to the search bounds and accepted using greedy selection:

$$X_i^{t+1} = \begin{cases} \widetilde{X}_i^t, & \text{if } f(\widetilde{X}_i^t) \le f(X_i^t), \\ X_i^t, & \text{otherwise.} \end{cases} \tag{14}$$

The HS operator has three intended effects. First, it reintroduces local diversity around selected agents after the population has started to contract. Second, it helps revive stagnant trajectories by allowing scouts to probe nearby unexplored regions. Third, the weak best-referenced term helps maintain search orientation without forcing all agents to move directly toward the same solution. In this study, HS is applied every iteration with scout rate $\gamma = 0.10$ and bias coefficient $\eta = 0.05$.

B. Foraging Flocks

The Foraging Flocks mechanism is introduced to provide coordinated subgroup search. Instead of moving the whole population toward a single global attractor, FF temporarily divides the population into smaller flocks so that different regions can be searched in parallel. Every T_f iterations, the population is randomly partitioned into K non-overlapping flocks:

$$\{1, \ldots, N\} = \bigcup_{k=1}^{K} F_k, \qquad F_k \cap F_{k'} = \varnothing \quad (k \neq k'). \tag{15}$$

For each flock F_k, the centroid is computed as:

$$C_k^t = \frac{1}{|F_k|} \sum_{i \in F_k} X_i^t. \tag{16}$$

The local flock leader is the best member of the flock:

$$q_k = \arg\min_{i \in F_k} f(X_i^t), \qquad L_k^t = X_{q_k}^t. \tag{17}$$

The flock then forms a blended target:

$$T_k^t = \phi L_k^t + (1 - \phi) C_k^t, \qquad 0 \le \phi \le 1, \tag{18}$$

where, ϕ controls the exploitation strength. A larger ϕ places more emphasis on the best member of the flock, whereas a smaller ϕ gives more influence to the centroid and therefore encourages broader group-level exploration.

For each member $i \in F_k$, FF generates a trial position:

$$\widetilde{X}_i^t = X_i^t + \xi_i^t \odot (T_k^t - X_i^t) + R_i^t, \tag{19}$$

where, $\xi_i^t \sim \mathcal{U}(0,1)^D$ is a random scaling vector and

$$R_i^t \sim \mathcal{N}(0, \rho_m^2 I_D) \tag{20}$$

is a Gaussian local-search term. The foraging radius decreases after each flocking event according to:

$$\rho_m = \rho_0 \delta^m, \qquad 0 < \delta < 1, \tag{21}$$

where, ρ_0 is the initial radius, δ is the radius decay factor, and m is the number of flocking events already performed.

The FF candidate is clipped to the feasible bounds and accepted if it improves the current solution. This mechanism acts on a larger scale than HS. In the early stage, the larger radius allows different flocks to explore several basins. In the middle stage, flock leaders and centroids guide coordinated exploitation of promising regions. In the late stage, the shrinking radius supports more precise local refinement. In this study, FF uses $K = 5$ flocks, is applied every $T_f = 5$ iterations, uses exploitation weight $\phi = 0.60$, initial radius fraction $\rho_0 = 0.10$, and radius decay $\delta = 0.95$.

C. Integrated Search Process

HSFFPKO preserves the main PKO cycle and inserts HS and FF as additional refinement stages. Each iteration begins with the original PKO exploration or exploitation update, followed by commensal association. The global best solution is then updated. After that, HS is applied to selected scouts, and FF is applied periodically to temporary flocks. Each generated candidate solution is clipped to the feasible domain and accepted using the same greedy rule used in PKO.

Algorithm 2 shows that HSFFPKO is a layered extension of PKO rather than a replacement for it. The original perching, hovering, diving, and commensal association operators remain unchanged. HS and FF are inserted after the baseline PKO update to provide additional search pressure at two different levels: individual-level perturbation and subgroup-level coordination.

D. Computational Complexity

Let C_f denote the computational cost of one objective-function evaluation. The baseline PKO evaluates N agents over T iterations and performs vector operations of dimension D. Therefore, its computational cost can be expressed as:

$$\mathcal{O}\big(TN(C_f + D)\big). \tag{22}$$

HS adds a scout update for approximately γN agents per iteration. Each scout update requires D-dimensional vector operations and one additional objective-function evaluation. Thus, the HS overhead is:

$$\mathcal{O}\big(T\gamma N(C_f + D)\big). \tag{23}$$

Algorithm 2 Hovering Scouts and Foraging Flocks Pied Kingfisher Optimizer

Require: Population size N, maximum iterations T, bounds LB, UB, objective function f
Require: Scout rate γ, scout bias η, number of flocks K, flocking interval T_f, exploitation weight ϕ, initial radius ρ_0, radius decay δ
Ensure: Best solution X^*
Initialize N kingfishers uniformly within [LB, UB]
Evaluate all agents and identify the initial best solution X^*
Set $m = 0$, where m counts the number of flocking events
for $t = 1$ to T **do**
  for each agent $i = 1, \ldots, N$ **do**
    if $\mathrm{rand} < 0.8$ **then**
      Generate a PKO exploratory trial solution using perching or hovering
    else
      Generate a PKO exploitative trial solution using diving
    Clip the trial solution to [LB, UB]
    Apply greedy selection
  Apply PKO commensal association and update X^*
  Select S_t with $|S_t| = \max(1, \lfloor \gamma N \rfloor)$
  for each scout $i \in S_t$ **do**
    Generate a scout candidate using HS
    Clip the candidate to [LB, UB]
    Apply greedy selection
  if $t \bmod T_f = 0$ **then**
    Randomly partition the population into K flocks
    Set $\rho_m = \rho_0 \delta^m$
    for each flock F_k **do**
      Compute centroid C_k^t and local leader L_k^t
      Compute blended target $T_k^t = \phi L_k^t + (1 - \phi) C_k^t$
      for each member $i \in F_k$ **do**
        Generate a flock candidate using FF
        Clip the candidate to [LB, UB]
        Apply greedy selection
    Set $m = m + 1$
  Update the global best solution X^*
return X^*

FF is applied every T_f iterations. At each flocking event, the population is partitioned into K flocks, and centroids, local leaders, and flock candidate positions are computed. Since each agent belongs to exactly one flock, the total vector-operation cost per flocking event is linear in ND, and the objective-function cost is linear in NC_f. Therefore, the FF overhead is:

$$\mathcal{O}\left(\frac{T}{T_f} N(C_f + D)\right). \tag{24}$$

Combining the baseline PKO, HS, and FF components gives:

$$\mathcal{O}\left(TN(C_f + D) + T\gamma N(C_f + D) + \frac{T}{T_f} N(C_f + D)\right). \tag{25}$$

Since γ and T_f are fixed control parameters, the asymptotic complexity remains:

$$\mathcal{O}\big(TN(C_f + D)\big), \tag{26}$$

which is the same order as the baseline PKO. Therefore, HSFFPKO does not change the asymptotic complexity class of PKO. Its additional cost is a constant-factor overhead caused by scout probing and periodic flock coordination. The memory complexity also remains:

$$\mathcal{O}(ND), \tag{27}$$

because the algorithm stores the population, fitness values, and a small number of auxiliary vectors for scout and flock operations.

V. Experimental Protocol

The proposed HSFFPKO was evaluated using two complementary test sets: the IEEE CEC 2017 single-objective real-parameter benchmark suite and nine constrained engineering design problems. The CEC 2017 suite contains 30 benchmark functions grouped into unimodal, multimodal, hybrid, and composition categories, as summarized in Table II. These categories provide different levels of difficulty: unimodal functions mainly evaluate convergence accuracy, multimodal functions test the ability to escape local optima, hybrid functions assess performance under mixed variable interactions, and composition functions evaluate robustness against shifted, rotated, and deceptive landscapes [22].

TABLE II. CEC 2017 Benchmark Categories and Experimental Setting.

Category	Functions	Domain	Main challenge
Unimodal	F1–F4	$[-100, 100]^D$	convergence speed and numerical precision
Multimodal	F5–F10	$[-100, 100]^D$	avoidance of many local optima
Hybrid	F11–F20	$[-100, 100]^D$	variable interaction and mixed landscape components
Composition	F21–F30	$[-100, 100]^D$	shifted, rotated, and deceptive global structures
Runs	30 independent runs per function		
Budget	$1000D$ function evaluations		
Statistics	Friedman test followed by Nemenyi post-hoc analysis		

For the CEC 2017 experiments, the search domain was fixed to $[-100, 100]^D$, and the maximum number of function evaluations was set to $1000D$ for each algorithm. Each algorithm was executed for 30 independent runs on each benchmark function to account for stochastic variability. The ablation study was first conducted at $D = 10$ to isolate the individual and combined contributions of Hovering Scouts and Foraging Flocks under a controlled computational setting. To further verify scalability, the four PKO-based variants were also evaluated at $D = 30$, $D = 50$, and $D = 100$. The broader comparison with recent optimizers was emphasized at $D = 30$, which provides a more demanding setting for assessing the overall competitiveness of HSFFPKO against state-of-the-art algorithms.

To ensure a fair comparison, all algorithms were evaluated using the same benchmark functions, dimensionality, search bounds, maximum function-evaluation budget, and number of independent runs. For HSFFPKO, the parameter values were fixed across all CEC 2017 functions: scout rate $\gamma = 0.10$, scout bias coefficient $\eta = 0.05$, number of flocks $K = 5$, flocking interval $T_f = 5$, exploitation weight $\phi = 0.60$, initial radius fraction $\rho_0 = 0.10$, and radius decay factor $\delta = 0.95$. The parameters of the competing algorithms were set according to their original studies or commonly recommended values to avoid problem-specific tuning bias.

Performance was assessed using the mean best objective value, standard deviation, and mean error with respect to the known optimum of each CEC 2017 function. Since metaheuristic algorithms are stochastic and their performance distributions are not guaranteed to satisfy normality assumptions, nonparametric statistical tests were used for algorithm comparison [23]. First, the Friedman test was applied to determine whether significant differences existed among the compared algorithms across the benchmark functions. When the Friedman test indicated significant differences, the Nemenyi post-hoc test was used to compare the average ranks and identify statistically meaningful pairwise differences. The statistical analysis was used not only to identify the best-ranked algorithm but also to determine whether HSFFPKO was significantly better than, or statistically comparable with, its closest competitors.

The engineering test set included nine constrained design problems: cantilever beam, gear train, I-beam, pressure vessel, speed reducer, tension/compression spring, three-bar truss, tubular column, and welded beam design. These problems complement the CEC 2017 suite because they involve nonlinear constraints, different numbers of decision variables, different sensitivity levels, and, in the case of the gear train problem, discrete integer structure. Thus, the engineering experiments provide a second-stage assessment of whether the proposed optimizer remains effective beyond synthetic benchmark functions and can be applied to practical constrained optimization problems [24].

For the engineering problems, the same general evaluation logic was used: each algorithm was run independently multiple times under a fixed evaluation budget, and performance was compared using solution quality, robustness, and runtime behavior. The best, mean, and standard deviation of the objective values were recorded. The results were then interpreted together with the CEC 2017 findings to determine whether HSFFPKO provides consistent practical advantages or whether its performance depends on the structure of the optimization problem.

VI. Results and Discussion

A. Ablation Analysis at $D = 10$

The ablation study was conducted to determine the individual and combined effects of the two proposed mechanisms. Four variants were compared: the baseline PKO, PKO with Hovering Scouts only (HSPKO), PKO with Foraging Flocks only (FFPKO), and the full HSFFPKO. This design makes it possible to examine whether HS and FF provide complementary contributions or whether one mechanism dominates the performance improvement.

The Friedman test produced $\chi^2 = 27.72$ with $p = 4.16 \times 10^{-6}$, indicating statistically significant differences among the four variants. As summarized in Table III, HSFFPKO obtained the best average rank of 1.67 and achieved the highest number of function wins, with 20 wins out of 30 CEC 2017 functions. HSPKO ranked second with an average rank of 2.27 and six wins, followed by the baseline PKO and FFPKO. These results indicate that the full integration of HS and FF provides the strongest overall performance among the tested variants.

The distribution of wins provides further insight into the role of each mechanism. HSFFPKO achieved the strongest performance on the unimodal, multimodal, and hybrid categories, suggesting that the combination of scout-based perturbation and flock-based coordination improves both convergence accuracy and robustness on complex landscapes. HSPKO achieved its strongest results on several composition functions, which indicates that the Hovering Scouts mechanism is especially useful for reactivating local diversity in deceptive search spaces. In contrast, FFPKO alone produced fewer wins, suggesting that flock-based exploitation is most effective when combined with the diversity-recovery ability of HS.

TABLE III. Ablation Summary on CEC 2017 at $D = 10$. Category Counts Record how many Function Wins were Obtained in each Family.

Algorithm	Avg. rank	Wins	U	M	H	C
HSFFPKO	**1.67**	**20**	**3**	**6**	**8**	3
HSPKO	2.27	6	0	0	2	**4**
PKO	2.70	3	0	0	0	3
FFPKO	3.37	1	1	0	0	0

The Nemenyi post-hoc test further clarifies these findings. HSFFPKO was significantly better than the baseline PKO ($p = 0.010$) and FFPKO ($p = 2.0 \times 10^{-6}$), but it was not significantly different from HSPKO ($p = 0.273$). This result should be interpreted carefully. It shows that HS is the dominant contributor to the improvement over baseline PKO, while FF provides complementary stability and additional exploitation capability. Therefore, the contribution of FF is not that it always dominates as a stand-alone mechanism, but that it strengthens the full HSFFPKO framework when combined with HS.

Function-level results provide more detailed evidence. On the unimodal functions, HSFFPKO obtained the best results on F1–F3, showing strong convergence precision. On the hybrid functions, HSFFPKO achieved the best result on most cases, including F11, F12, F14–F18, and F20. This is important because hybrid functions combine different base landscapes and therefore require a balance between exploration and exploitation. The main exceptions are F13 and F19, where HSPKO obtained the lowest mean error. These exceptions show that the proposed full variant is not uniformly dominant on every function, but it remains the most reliable variant across the whole benchmark set (see Table IV to Table VI).

The composition-function results reveal an important pattern. HSFFPKO remained competitive and obtained the best result on F23, F26, and F30, but HSPKO and PKO were better on some other composition functions. This suggests that the added flocking mechanism improves robustness on some complex landscapes but may occasionally introduce additional local contraction that is less beneficial when the landscape requires broader exploration. This observation is consistent with the no-free-lunch principle and supports a cautious interpretation of the ablation results: HSFFPKO provides the best overall balance among the tested variants, although some specific functions may favor simpler or HS-only search behavior.

B. Scalability Analysis at $D = 30$, $D = 50$, and $D = 100$

To further address the scalability of the proposed mechanisms, the four PKO-based variants were evaluated on higher-

TABLE IV. ABLATION STUDY ON $D = 10$ – UNIMODAL FUNCTIONS (F1–F4).

Func.	Algorithm	Mean ± Std	Error
F1	PKO	17417.6 ± 56536.6	17317.6
	HSPKO	7308.75 ± 10802.6	7208.75
	FFPKO	5818.79 ± 4072.93	5718.79
	HSFFPKO	**1052.16** ± 1116.73	952.158
F2	PKO	307.622 ± 309.408	107.622
	HSPKO	609.48 ± 1292.54	409.48
	FFPKO	210.575 ± 28.5371	10.5753
	HSFFPKO	**200.012** ± 0.062968	0.0119784
F3	PKO	315.799 ± 40.822	15.7985
	HSPKO	304.669 ± 9.56007	4.66858
	FFPKO	300.004 ± 0.0158136	0.00395122
	HSFFPKO	**300** ± 1.14421e-06	2.12218e-06
F4	PKO	405.349 ± 1.34174	5.34882
	HSPKO	405.613 ± 0.87472	5.61319
	FFPKO	**405.049** ± 1.86189	5.0488
	HSFFPKO	405.551 ± 0.645628	5.55107

TABLE V. ABLATION STUDY ON $D = 10$ – HYBRID FUNCTIONS (F11–F20).

Func.	Algorithm	Mean ± Std	Error
F11	PKO	1104.03 ± 1.37893	4.03043
	HSPKO	1104.04 ± 1.73024	4.03600
	FFPKO	1116.72 ± 35.0349	16.7229
	HSFFPKO	**1103.54** ± 1.19288	3.54384
F12	PKO	50679.1 ± 13201.8	49479.1
	HSPKO	46992.4 ± 18392.6	45792.4
	FFPKO	39395.6 ± 22806.4	38195.6
	HSFFPKO	**29184.7** ± 15420.2	27984.7
F13	PKO	2179.94 ± 238.866	879.939
	HSPKO	**2090.08** ± 348.25	790.085
	FFPKO	2596.49 ± 2122.84	1296.49
	HSFFPKO	2489.22 ± 2292.19	1189.22
F14	PKO	1467.18 ± 29.3744	67.1786
	HSPKO	1469.19 ± 28.8256	69.1924
	FFPKO	1535.58 ± 90.5509	135.579
	HSFFPKO	**1452.59** ± 38.769	52.5937
F15	PKO	2594.11 ± 1542.44	1094.11
	HSPKO	2272.43 ± 1014.08	772.434
	FFPKO	3394.45 ± 2034.35	1894.45
	HSFFPKO	**1597.46** ± 77.6397	97.4553
F16	PKO	1614.09 ± 25.4375	14.0942
	HSPKO	1608.83 ± 11.2774	8.83311
	FFPKO	1646.31 ± 59.4109	46.3078
	HSFFPKO	**1601.44** ± 2.15571	1.4431
F17	PKO	1720.46 ± 11.1111	20.4567
	HSPKO	1716.91 ± 11.1617	16.9054
	FFPKO	1729.65 ± 13.2253	29.6493
	HSFFPKO	**1711.96** ± 9.53097	11.9635
F18	PKO	10529.9 ± 9266.26	8729.91
	HSPKO	7543.23 ± 4118.77	5743.23
	FFPKO	7712.72 ± 5124	5912.72
	HSFFPKO	**4669.6** ± 5438.7	2869.6
F19	PKO	2722.85 ± 1597.53	822.85
	HSPKO	**2596.03** ± 1174.8	696.033
	FFPKO	7548.22 ± 9779.13	5648.22
	HSFFPKO	3787.92 ± 2793.19	1887.92
F20	PKO	2011.44 ± 9.01475	11.435
	HSPKO	2010.85 ± 9.81497	10.8532
	FFPKO	2027.18 ± 26.0327	27.1775
	HSFFPKO	**2010.14** ± 10.6252	10.1425

dimensional CEC 2017 benchmark functions at $D = 30$, $D = 50$, and $D = 100$. This experiment was designed to determine whether the performance advantage observed in the $D = 10$ ablation study remains stable as the dimensionality of the search space increases. The same experimental protocol was used, including the search range $[-100, 100]^D$, the evaluation budget of $1000D$ function evaluations, and 30 independent runs per function.

Table VII shows that the Friedman test detected statistically significant differences among the four PKO variants at all tested dimensions. In each case, HSFFPKO obtained the best average rank. Its rank improved from 1.37 at $D = 30$ to 1.33 at $D = 50$ and 1.27 at $D = 100$, indicating that the combined Hovering Scouts and Foraging Flocks mechanisms remain effective as the dimensionality of the search space increases.

The win-count results in Table VIII further support the scalability of HSFFPKO. At $D = 30$, HSFFPKO obtained the best result on 24 out of 30 functions. At $D = 50$, it remained best on 23 functions, and at $D = 100$, it achieved the best result on 25 functions. In contrast, the baseline PKO obtained two wins at $D = 30$ but no wins at $D = 50$ and $D = 100$. HSPKO and FFPKO occasionally produced the best result on a small number of functions, but neither variant matched the consistency of the full HSFFPKO.

These results indicate that the two proposed mechanisms are not only useful in low-dimensional settings but also remain effective under higher-dimensional search conditions. The Hovering Scouts mechanism helps restore diversity when the search space becomes larger, while the Foraging Flocks mechanism supports coordinated exploitation of multiple promising regions. Their combination allows HSFFPKO to maintain strong rank performance and high function-win counts as dimensionality increases.

Overall, the scalability analysis strengthens the ablation findings. The $D = 10$ experiment shows that HS and FF improve the baseline search process, while the $D = 30$, $D = 50$, and $D = 100$ experiments show that the full HSFFPKO remains reliable as the dimensionality of the problem increases. Therefore, the proposed mechanisms address not only mechanism-level improvement but also high-dimensional robustness.

C. Comparison with Recent Optimizers at $D = 30$

The broader comparison evaluated HSFFPKO against nine recent metaheuristic optimizers: BWSMA, GPSOM, HMS-GWO, MLSO [25], MMRFO, OMWOA, QSO, RCCO, and SSO. The compared methods and their main design features are summarized in Table IX. This comparison was conducted at $D = 30$ to evaluate the competitiveness of HSFFPKO under a more demanding dimensional setting than the ablation study.

The Friedman test across the 30 CEC 2017 functions yielded $\chi^2 = 138.92$ with $p = 1.727 \times 10^{-25}$, confirming that the rank differences among the compared optimizers are statistically significant. As shown in Table X, HSFFPKO obtained the best overall average rank of 2.60 and the highest number of

TABLE VI. ABLATION STUDY ON $D = 10$ – COMPOSITION FUNCTIONS (F21–F30).

Func.	Algorithm	Mean ± Std	Error
F21	PKO	2270.13 ± 53.5929	170.13
	HSPKO	**2251.16** ± 54.7097	151.161
	FFPKO	2267.15 ± 58.6099	167.149
	HSFFPKO	2257.36 ± 53.4223	157.357
F22	PKO	2296.84 ± 18.291	96.8393
	HSPKO	**2294.76** ± 21.4413	94.7557
	FFPKO	2295.58 ± 20.0678	95.5791
	HSFFPKO	2300.11 ± 0.210192	100.115
F23	PKO	2611.98 ± 3.56988	311.976
	HSPKO	2610.38 ± 3.03165	310.385
	FFPKO	2619.95 ± 7.78134	319.953
	HSFFPKO	**2607.2** ± 2.30879	307.202
F24	PKO	2733.76 ± 44.3326	333.76
	HSPKO	**2721.83** ± 75.146	321.828
	FFPKO	2742.25 ± 46.4699	342.245
	HSFFPKO	2735.41 ± 2.64616	335.408
F25	PKO	**2909.41** ± 20.3816	409.406
	HSPKO	2912.64 ± 21.8685	412.643
	FFPKO	2909.98 ± 20.5569	409.977
	HSFFPKO	2926.68 ± 23.3775	426.678
F26	PKO	2916.01 ± 28.4976	316.008
	HSPKO	2910.85 ± 23.2286	310.852
	FFPKO	2948.6 ± 34.5955	348.602
	HSFFPKO	**2901.57** ± 8.60817	301.572
F27	PKO	3089.62 ± 0.385193	389.616
	HSPKO	**3089.54** ± 0.5413	389.538
	FFPKO	3090.47 ± 1.43742	390.469
	HSFFPKO	3089.56 ± 0.393655	389.555
F28	PKO	**3169.12** ± 64.1953	369.123
	HSPKO	3170.41 ± 47.6545	370.407
	FFPKO	3188.87 ± 72.3728	388.869
	HSFFPKO	3176.17 ± 96.8557	376.171
F29	PKO	**3147.1** ± 11.3148	247.103
	HSPKO	3147.85 ± 14.2279	247.854
	FFPKO	3169.52 ± 27.3461	269.521
	HSFFPKO	3148.92 ± 11.3444	248.915
F30	PKO	484720 ± 194045	481720
	HSPKO	435978 ± 202781	432978
	FFPKO	368805 ± 242669	365805
	HSFFPKO	**339458** ± 240082	336458

TABLE VII. FRIEDMAN TEST RESULTS FOR SCALABILITY EVALUATION OF PKO VARIANTS

D	Friedman χ^2	p-value	Best algorithm	Best rank
30	35.64	8.92×10^{-8}	HSFFPKO	1.37
50	34.64	1.45×10^{-7}	HSFFPKO	1.33
100	39.76	1.20×10^{-8}	HSFFPKO	1.27

function wins, with 11 wins out of 30 functions. GPSOM and BWSMA were the closest competitors, with average ranks of 2.93 and 3.00, respectively. BWSMA achieved eight function wins, while GPSOM achieved six.

The category-level results show that HSFFPKO was the only algorithm with wins across all four benchmark families. It achieved one win on unimodal functions, three wins on multimodal functions, three wins on hybrid functions, and four wins on composition functions. This distribution suggests that HSFFPKO is not specialized for only one type of landscape. Instead, its strength comes from consistent performance across different search conditions. BWSMA was particularly strong on hybrid functions, while GPSOM was highly competitive on hybrid and composition functions. These results indicate that the leading algorithms form a close top-performing group rather than a single method dominating all functions.

The function-level results in Table XI, Table XII, and Table XIII support this interpretation. HSFFPKO achieved the best result on representative functions such as F1, F21, F25, F27, and F29, and remained competitive on several other functions. BWSMA performed best on functions such as F5, F23, and F24, while GPSOM produced the best results on F22 and F30. Therefore, the comparative results show that HSFFPKO has the strongest overall rank and broadest category coverage, but the performance advantage is function-dependent.

The representative numerical results show that HSFFPKO often combines low mean error with competitive standard deviation. However, the results also show that the advantage is not uniform across all functions. Some functions favor BWSMA, GPSOM, MLSO, or QSO. This confirms that HSFFPKO should be interpreted as a strong and balanced optimizer across the suite rather than as an algorithm that dominates every individual test case.

Table XIV gives the Nemenyi post-hoc comparison between HSFFPKO and each competitor. The results show that HSFFPKO is not statistically significantly different from BWSMA, GPSOM, and MLSO at the adopted significance level. In contrast, HSFFPKO is significantly better than QSO, MMRFO, HMSGWO, SSO, RCCO, and OMWOA. This distinction is important. HSFFPKO should not be described as universally superior to all competing algorithms. A more accurate conclusion is that HSFFPKO achieved the best overall rank and function-win count, while remaining statistically comparable to the strongest competitors, especially BWSMA and GPSOM.

Thus, the supported claim is that HSFFPKO is a top-tier and well-balanced optimizer for the tested CEC 2017 setting. Its main advantage is not absolute dominance on every function, but strong and stable behavior across unimodal, multimodal, hybrid, and composition landscapes.

D. Historical Search Behavior

To further examine the search dynamics of HSFFPKO, historical search visualizations were generated for four representative CEC 2017 functions at $D = 30$: F1 as a unimodal function, F5 as a multimodal function, F13 as a hybrid function, and F23 as a composition function. For each function, the visualization includes the landscape view, the projected search distribution, the relative discrepancy curve, and the mean fitness curve. These plots provide qualitative evidence of how the population moves, contracts, and refines solutions during the optimization process.

Fig. 1 shows the behavior of HSFFPKO on the unimodal function F1. The population rapidly moves toward the optimal region, and the discrepancy curve decreases smoothly. The mean fitness curve also drops sharply in the early iterations before gradually flattening, indicating fast convergence followed

TABLE VIII. SUMMARY OF PKO-VARIANT PERFORMANCE ACROSS HIGHER DIMENSIONS ON THE CEC 2017 BENCHMARK SUITE

Algorithm	$D = 30$		$D = 50$		$D = 100$	
	Avg. rank	Wins	Avg. rank	Wins	Avg. rank	Wins
HSFFPKO	**1.37**	**24**	**1.33**	**23**	**1.27**	**25**
HSPKO	2.67	2	2.67	4	2.60	2
PKO	2.67	2	2.87	0	2.93	0
FFPKO	3.30	2	3.13	3	3.20	3

TABLE IX. COMPARED ALGORITHMS AND THE MAIN DESIGN FEATURES EMPHASIZED IN THE EXPERIMENTS

Algorithm	Core search idea	Key settings
HSFFPKO	PKO augmented with Hovering Scouts and Foraging Flocks for local diversity and group-level exploitation	scout rate 0.10, scout bias 0.05, five flocks, flocking every five iterations, exploitation weight 0.60, radius fraction 0.10, radius decay 0.95
BWSMA	best–worst population partitioning with adaptive greedy and stagnation-reset mechanisms	reset probability 0.03, best/worst grouping, covariance-based dominant-set search
GPSOM	structured multi-swarm optimizer with explicit exploration, exploitation, and equilibrium blocks	three sub-swarms (20%/20%/60%) with time-varying exploration and exploitation schedules
HMSGWO	Grey Wolf variant with hierarchical approval, omega radius, and alpha/beta schedules	canonical GWO control $a(t)$ plus multiple adaptive coefficients
MLSO	lion-inspired search with mutation and roulette-like interaction	mutation probability decays from 0.25 to 0.02 with local king-step scaling
MMRFO	manta-ray based foraging with chain, cyclone, mating, and $\mu + \lambda$ selection	stochastic chain/cyclone steps with mating-based offspring generation
OMWOA	Whale Optimization with outpost scouting and multi-population organization	WOA coefficient schedule, outpost perturbation, axis nudges, multiple subpopulations
QSO	quokka-inspired search driven by environmental and drought parameters	environment factors and exploitation mixing tied to normalized time t/T
RCCO	red-crowned crane strategy switching between foraging, roosting, and dance behaviors	strategy toggle probability with time-varying long-distance and roost steps
SSO	salp-based search strengthened by stagnation handling and local search	SSA control schedule, stagnation trigger, and local recovery attempts

TABLE X. COMPARATIVE SUMMARY ON CEC 2017 AT $D = 30$

Algorithm	Avg. rank	Wins	U	M	H	C
HSFFPKO	**2.60**	**11**	**1**	**3**	3	**4**
GPSOM	2.93	6	**1**	0	3	2
BWSMA	3.00	8	0	2	**4**	2
MLSO	4.87	2	1	0	0	1
QSO	5.37	2	1	0	0	1
MMRFO	5.67	0	0	0	0	0
HMSGWO	6.63	1	0	1	0	0
SSO	7.20	0	0	0	0	0
RCCO	8.10	0	0	0	0	0
OMWOA	8.63	0	0	0	0	0

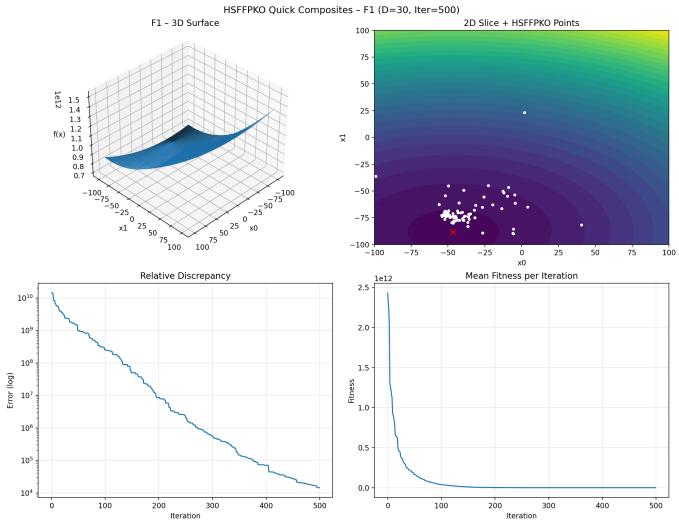

Fig. 1. Historical search on the unimodal function F1 at $D = 30$.

by local refinement. This behavior is expected for a unimodal landscape, where the main requirement is convergence speed and numerical precision.

Fig. 2 presents the behavior on the multimodal function F5. Compared with F1, convergence is slower and more stepwise because the landscape contains many local optima. The search points remain more dispersed during the earlier iterations, which suggests that the algorithm preserves exploration before committing to a promising basin. The subsequent reduction in discrepancy and mean fitness indicates that HSFFPKO can gradually escape poorer regions and refine the population around better solutions.

Fig. 3 shows the search behavior on the hybrid function F13. The population remains partially dispersed, reflecting the difficulty of optimizing a function composed of different landscape components. The mean fitness improves rapidly at first, while the discrepancy curve decreases more gradually. This suggests that HSFFPKO can identify promising regions early, but final refinement is more difficult when several landscape structures interact.

Fig. 4 illustrates the behavior on the composition function F23. The population moves toward the promising region, but some dispersion remains because the function contains shifted

TABLE XI. COMPARATIVE RESULTS ON CEC 2017 FUNCTIONS ($D = 30$) – F1–F5.

Func.	Algorithm	Mean ± Std	Error
F1	HSFFPKO	**2.002e+03** ± 1.746e+03	1.902e+03
	MLSO	1.152e+06 ± 1.512e+06	1.152e+06
	MMRFO	1.575e+06 ± 1.523e+06	1.575e+06
	OMWOA	3.839e+08 ± 4.157e+08	3.839e+08
	QSO	3.177e+05 ± 1.242e+05	3.176e+05
	RCCO	3.430e+07 ± 7.328e+07	3.430e+07
	SSO	5.002e+04 ± 5.853e+04	4.992e+04
	BWSMA	1.350e+04 ± 4.262e+04	1.340e+04
	GPSOM	6.277e+03 ± 5.774e+03	6.177e+03
	HMSGWO	1.911e+05 ± 4.127e+04	1.910e+05
F2	HSFFPKO	1.058e+13 ± 2.861e+13	1.058e+13
	MLSO	5.088e+12 ± 1.630e+13	5.088e+12
	MMRFO	1.074e+16 ± 2.339e+16	1.074e+16
	OMWOA	1.623e+21 ± 8.405e+21	1.623e+21
	QSO	**5.010e+11** ± 1.325e+12	5.010e+11
	RCCO	3.590e+21 ± 1.863e+22	3.590e+21
	SSO	3.691e+19 ± 1.798e+20	3.691e+19
	BWSMA	3.451e+13 ± 1.106e+14	3.451e+13
	GPSOM	1.435e+13 ± 7.097e+13	1.435e+13
	HMSGWO	7.198e+16 ± 1.208e+17	7.198e+16
F3	HSFFPKO	3.604e+04 ± 9.288e+03	3.574e+04
	MLSO	**5.786e+02** ± 2.966e+02	2.786e+02
	MMRFO	1.026e+04 ± 3.383e+03	9.959e+03
	OMWOA	2.823e+04 ± 8.063e+03	2.793e+04
	QSO	1.095e+04 ± 4.654e+03	1.065e+04
	RCCO	9.166e+03 ± 4.978e+03	8.866e+03
	SSO	2.065e+04 ± 7.625e+03	2.035e+04
	BWSMA	2.896e+03 ± 2.087e+03	2.596e+03
	GPSOM	1.057e+04 ± 4.663e+03	1.027e+04
	HMSGWO	3.600e+03 ± 1.692e+03	3.300e+03
F4	HSFFPKO	5.068e+02 ± 1.330e+01	1.068e+02
	MLSO	4.941e+02 ± 3.230e+01	9.408e+01
	MMRFO	5.242e+02 ± 3.759e+01	1.242e+02
	OMWOA	5.484e+02 ± 4.310e+01	1.484e+02
	QSO	5.005e+02 ± 2.071e+01	1.005e+02
	RCCO	5.186e+02 ± 2.675e+01	1.186e+02
	SSO	5.352e+02 ± 2.477e+01	1.352e+02
	BWSMA	5.060e+02 ± 2.797e+01	1.060e+02
	GPSOM	**4.866e+02** ± 2.586e+01	8.656e+01
	HMSGWO	5.201e+02 ± 2.557e+01	1.201e+02
F5	HSFFPKO	5.440e+02 ± 1.312e+01	4.402e+01
	MLSO	6.506e+02 ± 2.919e+01	1.506e+02
	MMRFO	6.602e+02 ± 3.029e+01	1.602e+02
	OMWOA	6.752e+02 ± 3.847e+01	1.752e+02
	QSO	6.504e+02 ± 3.350e+01	1.504e+02
	RCCO	7.345e+02 ± 6.103e+01	2.345e+02
	SSO	6.924e+02 ± 4.331e+01	1.924e+02
	BWSMA	**5.431e+02** ± 1.128e+01	4.308e+01
	GPSOM	5.858e+02 ± 1.781e+01	8.580e+01
	HMSGWO	6.237e+02 ± 2.059e+01	1.237e+02

TABLE XII. COMPARATIVE RESULTS ON CEC 2017 FUNCTIONS ($D = 30$) – F21–F25.

Func.	Algorithm	Mean ± Std	Error
F21	HSFFPKO	**2.343e+03** ± 9.733e+00	2.430e+02
	MLSO	2.444e+03 ± 3.062e+01	3.439e+02
	MMRFO	2.439e+03 ± 2.969e+01	3.392e+02
	OMWOA	2.468e+03 ± 3.726e+01	3.679e+02
	QSO	2.463e+03 ± 6.412e+01	3.632e+02
	RCCO	2.532e+03 ± 5.973e+01	4.321e+02
	SSO	2.485e+03 ± 4.375e+01	3.848e+02
	BWSMA	2.345e+03 ± 8.588e+00	2.452e+02
	GPSOM	2.368e+03 ± 1.755e+01	2.679e+02
	HMSGWO	2.419e+03 ± 3.571e+01	3.192e+02
F22	HSFFPKO	3.574e+03 ± 1.884e+03	1.374e+03
	MLSO	3.273e+03 ± 1.978e+03	1.073e+03
	MMRFO	2.951e+03 ± 1.705e+03	7.512e+02
	OMWOA	4.327e+03 ± 2.411e+03	2.127e+03
	QSO	4.366e+03 ± 1.918e+03	2.166e+03
	RCCO	4.593e+03 ± 2.636e+03	2.393e+03
	SSO	3.116e+03 ± 1.850e+03	9.160e+02
	BWSMA	3.769e+03 ± 1.463e+03	1.569e+03
	GPSOM	**2.306e+03** ± 4.093e+00	1.058e+02
	HMSGWO	3.290e+03 ± 2.003e+03	1.090e+03
F23	HSFFPKO	2.695e+03 ± 1.472e+01	3.947e+02
	MLSO	2.943e+03 ± 6.527e+01	6.434e+02
	MMRFO	2.905e+03 ± 7.519e+01	6.051e+02
	OMWOA	2.914e+03 ± 6.321e+01	6.135e+02
	QSO	2.846e+03 ± 5.182e+01	5.457e+02
	RCCO	2.935e+03 ± 1.072e+02	6.354e+02
	SSO	2.915e+03 ± 8.099e+01	6.145e+02
	BWSMA	**2.689e+03** ± 1.119e+01	3.892e+02
	GPSOM	2.771e+03 ± 3.716e+01	4.714e+02
	HMSGWO	2.979e+03 ± 1.266e+02	6.786e+02
F24	HSFFPKO	2.864e+03 ± 1.309e+01	4.644e+02
	MLSO	3.100e+03 ± 1.107e+02	6.998e+02
	MMRFO	3.040e+03 ± 7.865e+01	6.397e+02
	OMWOA	3.066e+03 ± 6.193e+01	6.658e+02
	QSO	2.971e+03 ± 4.651e+01	5.711e+02
	RCCO	3.060e+03 ± 6.946e+01	6.595e+02
	SSO	3.103e+03 ± 8.684e+01	7.034e+02
	BWSMA	**2.860e+03** ± 1.082e+01	4.603e+02
	GPSOM	2.911e+03 ± 2.004e+01	5.107e+02
	HMSGWO	3.123e+03 ± 9.966e+01	7.235e+02
F25	HSFFPKO	**2.898e+03** ± 1.408e+01	3.976e+02
	MLSO	2.909e+03 ± 2.355e+01	4.088e+02
	MMRFO	2.939e+03 ± 2.944e+01	4.388e+02
	OMWOA	2.972e+03 ± 4.229e+01	4.718e+02
	QSO	2.903e+03 ± 1.870e+01	4.031e+02
	RCCO	2.930e+03 ± 2.657e+01	4.298e+02
	SSO	2.930e+03 ± 2.675e+01	4.298e+02
	BWSMA	2.917e+03 ± 1.947e+01	4.167e+02
	GPSOM	2.902e+03 ± 1.899e+01	4.020e+02
	HMSGWO	2.947e+03 ± 2.154e+01	4.466e+02

and combined components. The staircase-like improvement in the discrepancy and mean fitness curves suggests that the algorithm continues to discover better basins during the search rather than converging immediately to a single region.

Overall, the historical search results support the quantitative findings. HSFFPKO converges efficiently on simpler landscapes and maintains enough diversity to continue improving on more deceptive functions. However, the visualizations also show that final localization remains more challenging on hybrid and composition functions, which motivates future work on adaptive parameter control or stronger terminal local search.

E. Engineering Design Problems

The engineering design experiments were conducted to determine whether the benchmark performance of HSFFPKO transfers to constrained real-world optimization problems. The test set includes continuous constrained problems, low-dimensional design problems, and one discrete integer problem. This allows the practical behavior of HSFFPKO to be evaluated under different types of engineering constraints.

As summarized in Table XV, HSFFPKO achieved the best standalone result on the pressure vessel problem and tied for the best result on the I-beam, speed reducer, spring, three-bar

TABLE XIII. COMPARATIVE RESULTS ON CEC 2017 FUNCTIONS ($D = 30$) – F26–F30.

Func.	Algorithm	Mean ± Std	Error
F26	HSFFPKO	4.122e+03 ± 2.045e+02	1.522e+03
	MLSO	5.818e+03 ± 1.685e+03	3.218e+03
	MMRFO	5.706e+03 ± 1.636e+03	3.106e+03
	OMWOA	5.953e+03 ± 1.121e+03	3.353e+03
	QSO	**3.705e+03** ± 1.077e+03	1.105e+03
	RCCO	5.629e+03 ± 1.869e+03	3.029e+03
	SSO	5.908e+03 ± 1.824e+03	3.308e+03
	BWSMA	3.902e+03 ± 2.575e+02	1.302e+03
	GPSOM	4.543e+03 ± 1.133e+03	1.943e+03
	HMSGWO	5.945e+03 ± 1.768e+03	3.345e+03
F27	HSFFPKO	**3.215e+03** ± 1.164e+01	5.155e+02
	MLSO	3.308e+03 ± 6.936e+01	6.084e+02
	MMRFO	3.300e+03 ± 5.981e+01	5.998e+02
	OMWOA	3.332e+03 ± 7.076e+01	6.320e+02
	QSO	3.273e+03 ± 3.527e+01	5.734e+02
	RCCO	3.295e+03 ± 6.296e+01	5.948e+02
	SSO	3.331e+03 ± 7.868e+01	6.314e+02
	BWSMA	3.233e+03 ± 1.399e+01	5.331e+02
	GPSOM	3.268e+03 ± 2.717e+01	5.685e+02
	HMSGWO	3.497e+03 ± 8.938e+01	7.969e+02
F28	HSFFPKO	3.237e+03 ± 2.193e+01	4.373e+02
	MLSO	**3.217e+03** ± 1.811e+01	4.170e+02
	MMRFO	3.272e+03 ± 3.604e+01	4.724e+02
	OMWOA	3.324e+03 ± 3.746e+01	5.240e+02
	QSO	3.227e+03 ± 2.124e+01	4.273e+02
	RCCO	3.276e+03 ± 3.275e+01	4.758e+02
	SSO	3.292e+03 ± 3.555e+01	4.915e+02
	BWSMA	3.252e+03 ± 2.996e+01	4.519e+02
	GPSOM	3.222e+03 ± 2.631e+01	4.225e+02
	HMSGWO	3.293e+03 ± 3.350e+01	4.935e+02
F29	HSFFPKO	**3.509e+03** ± 9.501e+01	6.085e+02
	MLSO	4.143e+03 ± 2.108e+02	1.243e+03
	MMRFO	4.208e+03 ± 2.453e+02	1.308e+03
	OMWOA	4.480e+03 ± 3.005e+02	1.580e+03
	QSO	4.036e+03 ± 2.376e+02	1.136e+03
	RCCO	4.496e+03 ± 4.185e+02	1.596e+03
	SSO	4.089e+03 ± 2.949e+02	1.189e+03
	BWSMA	3.561e+03 ± 1.725e+02	6.607e+02
	GPSOM	4.007e+03 ± 2.400e+02	1.107e+03
	HMSGWO	4.433e+03 ± 2.356e+02	1.533e+03
F30	HSFFPKO	1.916e+04 ± 3.481e+04	1.616e+04
	MLSO	1.897e+04 ± 3.760e+04	1.597e+04
	MMRFO	3.449e+04 ± 4.007e+04	3.149e+04
	OMWOA	2.615e+06 ± 4.927e+06	2.612e+06
	QSO	9.322e+06 ± 7.515e+06	9.319e+06
	RCCO	4.720e+05 ± 1.606e+06	4.690e+05
	SSO	4.644e+04 ± 3.933e+04	4.344e+04
	BWSMA	4.964e+04 ± 6.979e+04	4.664e+04
	GPSOM	**1.408e+04** ± 1.741e+04	1.108e+04
	HMSGWO	1.239e+07 ± 6.408e+06	1.239e+07

truss, and tubular column problems. On the cantilever beam and welded beam problems, BWSMA produced slightly better results, although HSFFPKO remained highly competitive. The main exception is the gear train problem, where HMSGWO obtained a better objective value than HSFFPKO, likely because this benchmark has a discrete integer search structure that is less naturally aligned with the continuous perturbation and flocking operators used in HSFFPKO.

The pressure vessel result is particularly important because it indicates that HSFFPKO can be effective on nonlinear constrained continuous design problems. The tied-best results on several other engineering problems further show that the proposed algorithm is not limited to synthetic benchmark functions. However, the gear train result reveals a limitation of the current design. Since the gear train problem has a discrete integer structure, the continuous perturbation and flocking mechanisms of HSFFPKO may be less naturally suited to the

TABLE XIV. NEMENYI POST-HOC COMPARISON BETWEEN HSFFPKO AND EACH COMPETITOR AT $D = 30$.

Comparator	p-value vs. HSFFPKO
BWSMA	1.000
GPSOM	1.000
MLSO	1.056×10^{-1}
QSO	1.467×10^{-2}
MMRFO	3.485×10^{-3}
HMSGWO	1.097×10^{-5}
SSO	1.791×10^{-7}
RCCO	8.924×10^{-11}
OMWOA	5.322×10^{-13}

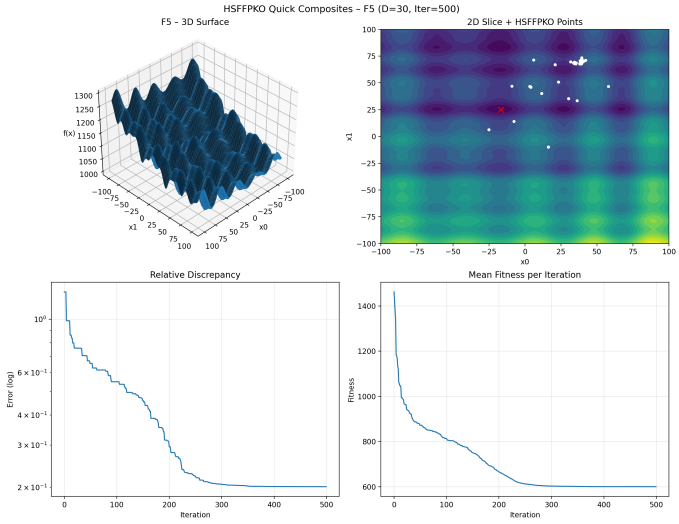

Fig. 2. Historical search on the multimodal function F5 at $D = 30$.

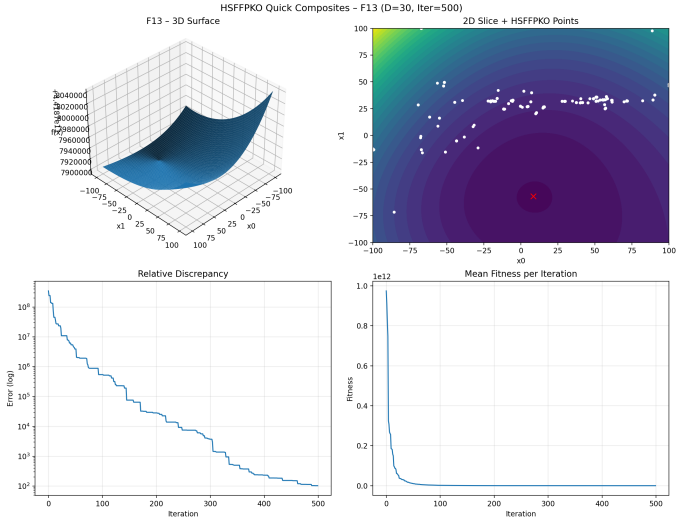

Fig. 3. Historical search on the hybrid function F13 at $D = 30$.

TABLE XV. CONDENSED ENGINEERING BENCHMARK RESULTS

Problem	Vars	Winning algorithm(s) and best mean	HSFFPKO outcome	Interpretation
Cantilever beam	5	BWSMA, 1.339956316	1.339957370; competitive	near-tie on a smooth constrained continuous problem
Gear train	4 int.	HMSGWO, 2.70×10^{-12}	5.70×10^{-11}; weaker	discrete ratio matching favors a different search dynamic
I-beam	4	GPSOM and HSFFPKO, 98.59387445	tied-best	stable convergence with negligible variance
Pressure vessel	4	HSFFPKO, 6049.858062	best overall	strongest exploitation under nonlinear constraints
Speed reducer	7	BWSMA, GPSOM, HSFFPKO, MLSO, 60503335.93	tied-best	several methods can reach the optimum on this case
Spring	3	GPSOM, HSFFPKO, RCCO, 0.001350022	tied-best	machine-precision convergence by multiple methods
Three-bar truss	2	all methods, 0.382842712	tied-best	problem is easy enough that ranking is not discriminative
Tubular column	2	BWSMA, HSFFPKO, MLSO, OMWOA, RCCO, SSO, 666.6666654	tied-best	another benchmark with many exact ties
Welded beam	4	BWSMA, 2.845786325	2.848179512; competitive	BWSMA holds a small edge on this nonlinear design

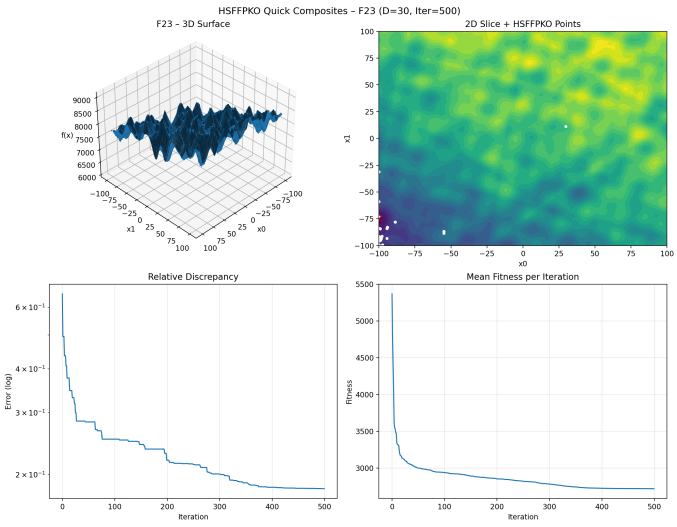

Fig. 4. Historical search on the composition function F23 at $D = 30$.

search space. In this case, the coordinated continuous basin search introduced by FF does not provide the same advantage that it provides on continuous problems.

The overall Friedman ranking on the engineering set places HSFFPKO first with an average rank of 2.67, followed closely by BWSMA with 2.72 and GPSOM with 3.72. This confirms that HSFFPKO is among the strongest methods for the engineering test set. Nevertheless, the results also support a cautious interpretation: HSFFPKO is highly competitive for continuous nonlinear constrained optimization, but specialized or more discrete-aware operators may be needed for integer or mixed-integer design problems.

F. Discussion

The results can be interpreted by viewing HSFFPKO as a layered search process. The baseline PKO already provides exploration through perching and hovering, exploitation through diving, and additional diversity through commensal association. However, the baseline method does not explicitly separate individual-level diversity recovery from subgroup-level basin exploitation. HS and FF were designed to address these two roles.

The ablation results show that HS is the dominant contributor to the performance gain. This is supported by the strong rank of HSPKO and the non-significant difference between HSPKO and HSFFPKO in the post-hoc test. The reason is that HS directly reactivates selected agents through scale-aware Gaussian probing, which helps reduce premature convergence and improves local diversity. FF, on the other hand, is more useful as a complementary mechanism because it organizes the population into temporary flocks and promotes coordinated exploitation around leader–centroid targets. The full HSFFPKO benefits from both mechanisms, but the improvement is not uniform across all function types.

The comparison with recent optimizers shows that HSFFPKO is a strong general-purpose optimizer. It achieved the best overall Friedman rank and the largest number of function wins at $D = 30$. However, the Nemenyi test shows that its differences from BWSMA and GPSOM are not statistically significant. Thus, HSFFPKO belongs to the top-performing group and offers the best overall balance in the present experiments.

G. Practical Implications

From a practical perspective, HSFFPKO has three main advantages. First, it is simple to integrate into the baseline PKO because HS and FF are inserted as additional operators without changing the main PKO structure. Second, it maintains the same asymptotic complexity class as PKO, which makes the additional search capability computationally manageable. Third, it performs well across both benchmark functions and constrained engineering problems, suggesting that it can be useful for practical optimization tasks where the search landscape is unknown.

The engineering results suggest that HSFFPKO is particularly suitable for continuous nonlinear constrained optimization problems. Its performance on the pressure vessel

problem and its tied-best results on several classical design problems indicate that the proposed mechanisms can improve practical search behavior. The algorithm is therefore promising for engineering applications where objective functions are expensive, nonlinear, or difficult to differentiate.

H. Limitations

Despite its strong overall performance, HSFFPKO has several limitations. First, although the scalability analysis shows that HSFFPKO remains effective at $D = 30$, $D = 50$, and $D = 100$, the full comparison with all recent optimizers was emphasized at $D = 30$. Future work should extend the broad multi-algorithm comparison to higher dimensions to determine whether the same relative ranking among HSFFPKO, BWSMA, GPSOM, and other competitors remains stable under larger-scale settings.

Second, the proposed mechanisms are mainly designed for continuous search spaces. The weaker result on the discrete gear train problem suggests that HSFFPKO may require discrete or mixed-integer adaptations for problems with integer variables. Third, the current parameter settings were fixed across all benchmark functions. Although this supports fairness and simplicity, adaptive control of scout rate, scout bias, flocking interval, exploitation weight, and radius decay may further improve robustness.

Finally, the historical search visualizations show that final refinement remains difficult on some hybrid and composition functions. This suggests that HSFFPKO could benefit from a stronger terminal local-search procedure, adaptive radius control, or hybridization with problem-specific refinement operators.

VII. Conclusion

This study presented HSFFPKO, an enhanced variant of PKO that integrates Hovering Scouts and Foraging Flocks to improve diversity preservation, coordinated basin search, and late-stage refinement. HS performs scale-aware local probing with a weak directional bias toward the current best solution, while FF creates temporary subgroups that search around leader–centroid targets under a shrinking radius. The proposed method preserves the simple structure of PKO while adding two complementary search mechanisms that operate at different scales.

The experimental results show that HSFFPKO provides strong and consistent performance across benchmark and engineering problems. In the ablation study at $D = 10$, HSFFPKO achieved the best average rank of 1.67 and obtained 20 wins out of 30 CEC 2017 functions. In the scalability analysis, HSFFPKO remained the best PKO-based variant at $D = 30$, $D = 50$, and $D = 100$, with average ranks of 1.37, 1.33, and 1.27, respectively. In the broad comparison with recent optimizers at $D = 30$, HSFFPKO obtained the best overall average rank of 2.60 and the largest number of function wins. However, the Nemenyi post-hoc test showed that HSFFPKO was statistically comparable with BWSMA and GPSOM, indicating that it should be interpreted as a top-tier optimizer rather than a universally dominant method.

The engineering results further confirmed the practical relevance of HSFFPKO. The proposed algorithm achieved the best result on the pressure vessel problem and tied for the best result on several other constrained design problems. Nevertheless, its weaker result on the discrete gear-train problem suggests that future work should develop discrete or mixed-integer variants of HSFFPKO. Other future directions include adaptive parameter control, stronger local exploitation on deceptive hybrid and composition functions, surrogate-assisted variants for expensive optimization, and extension to multi-objective optimization.

References

[1] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95 – International Conference on Neural Networks*, vol. 4, pp. 1942–1948, 1995, doi: 10.1109/ICNN.1995.488968.

[2] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey Wolf Optimizer," *Advances in Engineering Software*, vol. 69, pp. 46–61, 2014, doi: 10.1016/j.advengsoft.2013.12.007.

[3] S. Mirjalili and A. Lewis, "The Whale Optimization Algorithm," *Advances in Engineering Software*, vol. 95, pp. 51–67, 2016, doi: 10.1016/j.advengsoft.2016.01.008.

[4] S. Mirjalili, A. H. Gandomi, S. Z. Mirjalili, S. Saremi, H. Faris, and S. M. Mirjalili, "Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems," *Advances in Engineering Software*, vol. 114, pp. 163–191, 2017, doi: 10.1016/j.advengsoft.2017.07.002.

[5] X.-S. Yang, "A new metaheuristic bat-inspired algorithm," in *Nature Inspired Cooperative Strategies for Optimization (NICSO 2010)*, Studies in Computational Intelligence, vol. 284, J. R. González, D. A. Pelta, C. Cruz, G. Terrazas, and N. Krasnogor, Eds. Berlin, Heidelberg: Springer, 2010, pp. 65–74, doi: 10.1007/978-3-642-12538-6_6.

[6] J. Kang and Z. Ma, "Red-Crowned Crane Optimization: A novel biomimetic metaheuristic algorithm for engineering applications," *Biomimetics*, vol. 10, no. 9, p. 565, 2025, doi: 10.3390/biomimetics10090565.

[7] W. J. Al-Kubaisy and B. Al-Khateeb, "Quokka swarm optimization: A new nature-inspired metaheuristic optimization algorithm," *Journal of Intelligent Systems*, vol. 33, no. 1, 2024, doi: 10.1515/jisys-2024-0051.

[8] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 67–82, 1997, doi: 10.1109/4235.585893.

[9] D. Farinati and L. Vanneschi, "A survey on dynamic populations in bio-inspired algorithms," *Genetic Programming and Evolvable Machines*, vol. 25, no. 2, 2024, doi: 10.1007/s10710-024-09492-4.

[10] Z. Jakšić, S. Devi, O. Jakšić, and K. Guha, "A comprehensive review of bio-inspired optimization algorithms including applications in microelectronics and nanophotonics," *Biomimetics*, vol. 8, no. 3, p. 278, 2023, doi: 10.3390/biomimetics8030278.

[11] M. Sarhani, S. Voß, and R. Jovanovic, "Initialization of metaheuristics: Comprehensive review, critical analysis, and research directions," *International Transactions in Operational Research*, vol. 30, no. 6, pp. 3361–3397, 2022, doi: 10.1111/itor.13237.

[12] W. H. Bangyal, J. Ahmed, and H. T. Rauf, "A modified bat algorithm with torus walk for solving global optimisation problems," *International Journal of Bio-Inspired Computation*, vol. 15, no. 1, pp. 1–13, 2020, doi: 10.1504/IJBIC.2020.105861.

[13] W. H. Bangyal, A. Hameed, J. Ahmad, K. Nisar, M. R. Haque, A. A. A. Ibrahim, J. J. P. C. Rodrigues, M. A. Khan, D. B. Rawat, and R. Etengu, "New modified controlled bat algorithm for numerical optimization problem," *Computers, Materials & Continua*, vol. 70, no. 2, pp. 2241–2259, 2022, doi: 10.32604/cmc.2022.017789.

[14] W. H. Bangyal, J. Ahmad, and H. T. Rauf, "Optimization of neural network using improved bat algorithm for data classification," *Journal of Medical Imaging and Health Informatics*, vol. 9, no. 4, pp. 670–681, 2019, doi: 10.1166/jmihi.2019.2654.

[15] A. Bouaouda, F. A. Hashim, Y. Sayouti, and A. G. Hussien, "Pied kingfisher optimizer: A new bio-inspired algorithm for solving numerical optimization and industrial engineering problems," *Neural Computing and Applications*, 2024, doi: 10.1007/s00521-024-09879-5.

[16] W. Zhao, Z. Zhang, and L. Wang, "Manta ray foraging optimization: An effective bio-inspired optimizer for engineering applications," *Engineering Applications of Artificial Intelligence*, vol. 87, p. 103300, 2020, doi: 10.1016/j.engappai.2019.103300.

[17] A. N. K. Nasir, M. A. A. Roslan, M. F. M. Jusof, M. R. Ahmad, and A. A. A. Razak, "Mating-based manta ray foraging optimization for fuzzy-Hammerstein model of an electric water heater," *Journal of Advanced Research in Applied Mechanics*, vol. 129, no. 1, pp. 32–43, 2024, doi: 10.37934/aram.129.1.3243.

[18] K. Tang and L. Zhang, "An enhanced whale optimization algorithm with outpost and multi-population mechanisms for high-dimensional optimization and medical diagnosis," *PLOS ONE*, vol. 20, no. 6, p. e0325272, 2025, doi: 10.1371/journal.pone.0325272.

[19] T. Li, H. Meng, D. Wang, B. Fu, Y. Shao, and Z. Liu, "An enhanced slime mould algorithm based on best–worst management for numerical optimization problems," *Biomimetics*, vol. 10, no. 8, p. 504, 2025, doi: 10.3390/biomimetics10080504.

[20] J. Yan, G. Hu, H. Jia, A. G. Hussien, and L. Abualigah, "GPSOM: Group-based particle swarm optimization with multiple strategies for engineering applications," *Journal of Big Data*, vol. 12, article 114, 2025, doi: 10.1186/s40537-025-01140-7.

[21] I. Dagal, A. W. Ibrahim, A. Harrison, *et al.*, "Hierarchical multi step Gray Wolf optimization algorithm for energy systems optimization," *Scientific Reports*, vol. 15, article 8973, 2025, doi: 10.1038/s41598-025-92983-w.

[22] N. H. Awad, M. Z. Ali, P. N. Suganthan, J. J. Liang, and B. Y. Qu, "Problem definitions and evaluation criteria for the CEC 2017 special session and competition on single objective real-parameter numerical optimization," Technical Report, Nanyang Technological University, Singapore, 2016.

[23] J. Derrac, S. García, D. Molina, and F. Herrera, "A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms," *Swarm and Evolutionary Computation*, vol. 1, no. 1, pp. 3–18, 2011, doi: 10.1016/j.swevo.2011.02.002.

[24] E. H. Houssein, M. H. A. Gafar, N. Fawzy, and A. Y. Sayed, "Recent metaheuristic algorithms for solving some civil engineering optimization problems," *Scientific Reports*, vol. 15, no. 1, 2025, doi: 10.1038/s41598-025-90000-8.

[25] F. Ji and M. Jiang, "Lion swarm optimization by reinforcement pattern search," in *Advances in Swarm Intelligence*, Lecture Notes in Computer Science, vol. 12689. Cham: Springer, 2021, pp. 119–129, doi: 10.1007/978-3-030-78743-1_11.