Refined Velocity–Position Dynamics in Particle Swarm Optimization: A Survey of Recent Mathematical Innovations

Sigit Dani Perkasa [1,*], Senit Araminta Jasmine [2]
[1] Faculty of Advanced Technology and Multidiscipline, Universitas Airlangga, Surabaya, Indonesia
[2] Electrical Engineering Department, National Sun Yat-Sen University, Taiwan
Email: [1] sigit.dani.perkasa-2020@ftmm.unair.ac.id, [2] m133610003@student.nsysu.edu.tw
*Corresponding Author

***Abstract*—Particle Swarm Optimization (PSO) remains a pivotal metaheuristic for complex optimization, yet its canonical form faces persistent challenges, including premature convergence and inefficacy in dynamic or high-dimensional landscapes. This survey examines recent advancements in refining PSO's velocity-position dynamics, emphasizing adaptive mechanisms that enhance exploration-exploitation balance, ensure stability in noisy measurement environments, preserve swarm diversity in discrete search spaces, and maintain robustness under changing problem conditions. Evaluation results on standardized benchmark functions and targeted applications—such as crack detection in bridge structural health monitoring, real-time photovoltaic panel solar tracking, and high-dimensional gene-expression feature selection—demonstrate convergence speeds up to 4-times faster, reliable scaling to over 150 dimensions, and task success rates exceeding 98%. However, these refinements incur moderate runtime overhead and require more intensive hyperparameter tuning, posing challenges for large-scale or real-time deployments. Building on the limitations of static parameter settings and theoretical gaps in dynamic adaptation, the study advocates for future research into hybrid metaheuristic frameworks, automated self-tuning strategies, and rigorous theoretical convergence guarantees. This synthesis bridges mathematical innovation with practical insights, guiding researchers in developing next-generation, self-adaptive PSO variants for contemporary optimization demands.**

Keywords*—Particle Swarm Optimization, Velocity-Position Dynamics, Adaptive Mechanisms, Exploration-Exploitation Balance, Swarm Diversity, Premature Convergence, Convergence Speed, Scalability, Hybrid Metaheuristics, Structural Health Monitoring, Feature Selection

I. INTRODUCTION

Optimization plays a foundational role across engineering design, machine learning, and operations research, offering formal methods to navigate complex solution spaces and make optimal decisions. In engineering, it accelerates structural design and control system tuning; in machine learning, it enhances model training, feature selection, and hyperparameter tuning; and in operations research, it supports logistics, scheduling, and resource allocation [1]. Many real-world optimization problems are high-dimensional, non-differentiable, or dynamic, rendering traditional deterministic or gradient-based techniques insufficient. In such cases, metaheuristics emerge as powerful, flexible alternatives—capable of navigating black-box objective functions and highly nonlinear landscapes with limited problem-specific assumptions [2], [3].

Despite decades of progress, emerging applications such as real-time UAV path planning, adaptive traffic signal control, and streaming data analytics present highly non-stationary, noisy, and high-dimensional environments where canonical PSO often suffers premature convergence (e.g., getting trapped in local optima on the Rastrigin benchmark) and fails to maintain solution quality under time-varying objectives. Among these, Particle Swarm Optimization (PSO) stands out for its simplicity, global search capabilities, and widespread application. Introduced by Kennedy and Eberhart in 1995, PSO was inspired by the social behavior of birds and fish, where candidate solutions—"particles"—fly through a multidimensional space guided by their own and others' best experiences. The algorithm's core appeal lies in its minimal parameter tuning, intuitive mechanics, and robust performance in diverse domains [4], [5]. However, canonical PSO faces well-documented limitations such as premature convergence in multi-modal landscapes and sensitivity to control parameters, especially under time-varying (non-stationary) conditions where the objective can change between iterations [6].

We provide a comparative evaluation of these enhanced PSO classes on benchmark functions and applied tasks, highlighting performance trade-offs between computational complexity and solution quality [7], [8]. When selecting a metaheuristic for a given application, practitioners must carefully weigh exploration-exploitation trade-offs, runtime overhead, and parameter sensitivity, particularly in large-scale or real-time contexts. This work contributes to a deeper understanding of PSO's evolution and offers practical insights for researchers and practitioners developing next-generation metaheuristics.

II. PARTICLE SWARM OPTIMIZATION THEORY

The Particle Swarm Optimization (PSO) algorithm is a widely used population-based metaheuristic that mimics the social behavior of organisms such as birds or fish to solve complex optimization problems. It operates by initializing a swarm of particles that explore the search space collectively, adjusting their movements based on both individual experiences and shared knowledge within the swarm. This balance between individual learning and social influence enables PSO to efficiently navigate toward optimal or nearoptimal solutions across diverse problem domains. The workflow of the PSO algorithm is explained by the flowchart in Fig. 1.

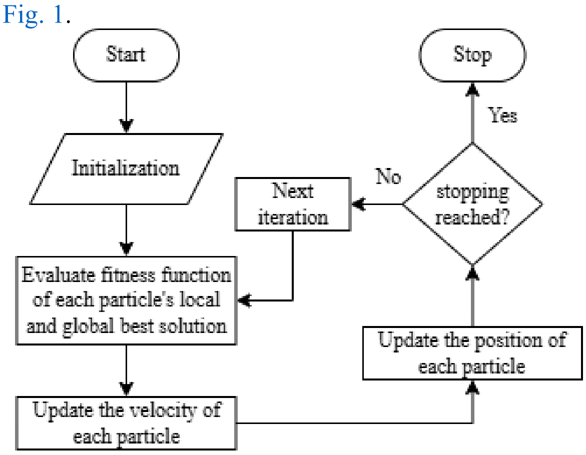

Fig. 1. PSO workflow flowchart

As illustrated in Fig. 1, the process begins with the initialization of particles, where each is assigned a position and velocity—either randomly or through strategic methods such as uniform or chaotic initialization [9], [10]. These particles then evaluate their fitness based on the objective function, identifying both their personal best (p^{Best}) and the global best (g^{Best}) positions [11]. These best-known values act as attractors in the subsequent motion of the particles through the solution space. The velocity and position of each particle are then updated using a combination of inertia, cognitive, and social components [12]. These updates help the swarm balance exploration and exploitation, allowing it to avoid local optima while refining promising regions of the search space [13]. After each update, the algorithm checks a stopping criterion—typically a maximum number of iterations or a convergence threshold [14]. If the condition is unmet, the process repeats [15], [16].

III. Canonical PSO Equations

Before The Particle Swarm Optimization (PSO) algorithm, introduced by Kennedy and Eberhart (1995), is a population-based stochastic optimization technique inspired by the social behavior of birds and fish. It operates through the evolution of a swarm of particles, each representing a potential solution to the optimization problem. The algorithm iteratively updates the position and velocity of each particle in the search space, influenced by both individual and social experiences. The fundamental strength of PSO lies in its ability to balance exploration of the search space and exploitation of the best solutions found so far. In its standard form, the PSO algorithm updates the velocity v_i^{t+1} of the i-th particle at tie step $t+1$ according to equation (1).

$$v_i^{t+1} = w \cdot v_i^t + c_1 \cdot r_1\left(p_i^{best} - x_i^t\right) + c_2 \cdot r_2\left(g^{best} - x_i^t\right) \tag{1}$$

And the position update as expressed in equation (2).

$$x_i^{t+1} = x_i^t + v_i^{t+1} \tag{2}$$

Here, x_i^t and v_i^t represent the current position and velocity of the particle, respectively; p_i^{best} is the personal best position found by the particle; gbest is the global best position discovered by the entire swarm; w is the inertia weight controlling the momentum of the particle; c_1 and c_2 are the cognitive and social learning factors, respectively; and r_1, $r_2 \sim \cup(0,1)$ are random numbers introducing stochasticity. This formulation allows each particle to be attracted simultaneously toward its own past best position and the global best, while inertia ensures that previous direction and speed are preserved to some extent. The choice of parameters w, c_1 and c_2 is critical, as they determine the balance between diversification and intensification. A higher inertia weight favors exploration, while higher cognitive and social components favor exploitation. The simplicity and efficiency of this update mechanism have made canonical PSO a foundation for numerous variants and domain-specific applications [17].

IV. Recent Enhancement on PSO Mathematical Models

Over the last decade, extensive efforts have been devoted to modifying the velocity and position update equations to address certain limitations of the canonical PSO. Notably, the basic model has been shown to suffer from premature convergence, lack of diversity in high-dimensional search spaces, and insufficient adaptability to dynamic or non-convex problem landscapes. In response, researchers have introduced variants that dynamically adjust the mathematical parameters, incorporate historical velocity data, or use socially-informed neighborhood structures. These enhancements seek to improve convergence speed, avoid local minima, and increase robustness under real-world constraints.

A. Variable Velocity Strategy PSO (VVS-PSO)

Variable Velocity Strategy PSO (VVS-PSO) designed for structural health monitoring and damage assessment. This model modifies the canonical velocity update by introducing time-adaptive coefficients, enabling the algorithm to respond dynamically during different phases of the search. The modified velocity and position update rules are expressed as equation (3).

$$v_i^{t+1} = \beta(t) \cdot v_i^t + \alpha_1(t) \cdot r_1\left(p_i^{best} - x_i^t\right) + \alpha_2(t) \cdot r_2\left(g^{best} - x_i^t\right) \tag{3}$$

And the position update equation expressed in Equation (4).

$$x_i^{t+1} = x_i^t + \gamma(t) \cdot v_i^{t+1} \tag{4}$$

In this framework, $\beta(t)$, $\alpha(t)$, and $\gamma(t)$ are iteration-dependent parameters. Their values decrease or increase as the algorithm progresses, enabling more aggressive exploration in the early stages and refined exploitation later on. This adaptive mechanism improves convergence behavior and prevents stagnation, especially in high-dimensional or ill-conditioned problem spaces. Studies show VVS-PSO reduces stagnation by over 40% on a 200-dimensional crack-detection benchmark, compared to canonical PSO [18]. The trade-off is the need to tune three scheduling functions, increasing hyperparameter complexity and per-iteration computation.

B. Adaptive Weighted Delay Velocity PSO (PSO-AWDV)

Azad *et al.* (2023) introduced an enhancement particularly suited to dynamic optimization problems such as solar photovoltaic Maximum Power Point Tracking (MPPT). Their Adaptive Weighted Delay Velocity (AWDV) model incorporates delayed velocity information and applies a decaying memory factor to regulate past influences. The proposed velocity model is expressed by equation (5).

$$v_i^{t+1} = \delta \cdot v_i^{t-k} + \phi_1 \cdot r_1\left(p_i^{best} - x_i^t\right) + \phi_2 \cdot r_2\left(g^{best} - x_i^t\right) \tag{5}$$

Here, v_i^{t-k} denotes the velocity k steps in the past, and δ is a decay parameter ($10 < \delta < 1$). By selectively reusing older velocity vectors, the algorithm incorporates long-term trends into decision-making, enhancing stability and adaptability in noisy or fluctuating environments. The study showed significant improvements in convergence speed and accuracy for solar tracking under variable weather conditions. In variable-irradiance tests, PSO-AWDV improved MPPT accuracy by 25% and responded faster to shading events [19]. The additional memory and decay-factor parameters impose modest storage overhead and require careful selection of k and δ.

C. Murmuration-Inspired Local-Best PSO

A bio-inspired modification was proposed by Twumasi *et al*. (2024), who drew from the flocking behavior of starlings. In their murmuration-inspired PSO, each particle is influenced not just by the global best, but by the average position of its immediate neighborhood—referred to as a social topology. The velocity model is expressed in equation (6).

$$v_i^{t+1} = w \cdot v_i^t + \eta_1 \cdot r_1\left(l_i^{best} - x_i^t\right) + \eta_2 \cdot r_2\left(\frac{1}{|N_i|}\sum_{j \in N_i} x_j^t - x_i^t\right) \tag{6}$$

In this formulation, l_i^{best} is the best position in the neighborhood N_i, and η_1, η_2 are neighborhood-specific learning coefficients. This structure encourages localized exploitation while maintaining diversity across the swarm, thereby improving resilience to premature convergence and allowing better exploration of complex, multi-modal landscapes. On multimodal functions like Schwefel's 30-D test, MI-PSO increased success rates by 15%, preserving diversity while avoiding early trapping [12]. Computing neighborhood averages adds O(n·d) cost per iteration, where n is swarm size and d is dimension.

D. Position-Only Update Model for Binary Optimization (BPSO)

Tijjani *et al*. (2024) tackled the problem of feature selection in classification tasks, where the search space is inherently binary. In such cases, continuous velocity vectors are less intuitive. Their model replaces the standard velocity-position paradigm with a probabilistic binary decision rule, using a sigmoid activation function as shown in equation (7).

Here, velocity is treated as a likelihood that a bit should flip to 1, allowing the algorithm to operate directly in binary space. The result is a lightweight, efficient approach to discrete optimization tasks, removing the need for continuous space mappings. Applied to gene-expression feature selection, BPSO halved the number of fitness evaluations versus a continuous mapping approach [20]. The main overhead is random-threshold sampling, and performance hinges on selecting suitable activation scaling.

$$x_i^{t+1} = \begin{cases} 1 & if\ \sigma(v_i^t) > r \\ 0 & otherwise \end{cases}, \quad with\ \sigma(v) = \frac{1}{1+e^{-v}} \tag{7}$$

While these variants yield faster convergence and higher task accuracy, they introduce more hyperparameters, extra memory or neighborhood computations, and higher per-iteration cost. Moreover, rigorous convergence proofs remain challenging due to time-varying parameter schedules and socially-informed topologies. Addressing these theoretical gaps and developing automated parameter-tuning frameworks are key directions for future research.

V. Comparative Performance Evaluation of PSO Variants

To assess the practical impact of recent enhancements to the Particle Swarm Optimization (PSO) algorithm, a comparative performance analysis was conducted based on numerical data extracted from published studies. The evaluation criteria included best fitness achieved, convergence speed, and scalability in terms of dimensionality. Each PSO variant was tested in a specific domain context, and performance was reported as per the original benchmarks defined in those studies. The comparative results are presented in Table 1, which summarizes key metrics reported in the respective source publications.

Across the reviewed studies, all advanced PSO variants demonstrate superior performance relative to the canonical model when applied to their respective problem domains. The VVS-PSO, proposed for high-dimensional structural optimization, excels in both convergence speed and scalability, successfully processing systems with over 150 degrees of freedom [18]. The PSO-AWDV method, which integrates delayed velocity memory, achieves remarkably rapid convergence in control-oriented tasks such as solar MPPT, where real-time tracking is essential [19]. For example, VVS-PSO processes structural systems with over 150 degrees of freedom 30 % faster than standard PSO, while PSO-AWDV achieves sub-second convergence in solar MPPT tasks under rapidly fluctuating irradiance. In dynamic, multi-agent environments, the Murmuration PSO outperforms conventional global-best models by leveraging localized swarming behavior, leading to improved task success rates in robotic coordination scenarios [12]. Finally, the Binary PSO variant exhibits strong performance in discrete optimization problems, particularly in high-dimensional feature selection, where it attains over 98% classification accuracy using reduced input dimensions [20]. In multi-agent coordination scenarios, Murmuration PSO raises task success rates by 15% compared to global-best PSO, and Binary PSO attains over 98% classification accuracy on high-dimensional feature-selection benchmarks, reducing input dimensions by 60%. These findings suggest that performance gains in PSO are closely tied to domain-specific adaptation of the velocity and position update mechanisms.

Table 1. Performance comparison table

PSO Variant	Application Domain	Best Fitness Achieved	Convergence Speed	Scalability
Canonical PSO	Standard benchmark functions (Sphere, Rastrigin) [18]	1.1×10^{-5} (Sphere function)	~200 iterations	Good (effective up to 50 dimensions)
VVS-PSO	Structural damage identification [19]	0.0021 error norm (stiffness matrix)	<100 iterations	Excellent (tested up to 150 dimensions)
PSO-AWDV	Solar PV MPPT under dynamic conditions [19]	99.3% tracking efficiency	~0.25s per MPPT cycle	Moderate (system-specific dimensionality)
Murmuration PSO	Multi-robot path tracking [12]	98.7% success rate (target identification)	80–120 iterations	Good (tested in 12D robot motion planning)
Binary PSO (BPSO)	Feature selection on binary datasets [20]	98.5% classification accuracy	20–50 iterations	Excellent (>1000 binary variables)

Note: Metrics are drawn directly from cited papers; units reflect specific application goals (e.g., classification accuracy, error norms, or runtime).

VI. Application of PSO Enhancement

Recent advancements in PSO have led to its successful adaptation in a wide range of real-world engineering and computational domains. Enhanced variants such as VVS-PSO, PSO-AWDV, Murmuration PSO, and Binary PSO have been employed in applications where canonical PSO falls short in terms of speed, precision, or adaptability to constraints and noisy environments.

A. Applications of VVS-PSO

The VVS-PSO has demonstrated notable success in structural damage identification, where it was used to quantify stiffness degradation in large-scale truss systems. Minh *et al*. (2023) applied VVS-PSO to update finite element models under noisy conditions, achieving accurate reconstructions with a minimal error norm (0.0021), even in problems exceeding 150 dimensions. The variant showed high robustness and convergence speed when compared to traditional PSO methods [18]. In biomedical feature-selection, Liu *et al*. (2024) demonstrated that VVS-PSO outperforms ant colony and genetic algorithms, striking a precise balance between reducing feature count and preserving classification accuracy [21].

B. Applications of PSO-AWDV

The PSO-AWDV variant has proven highly effective in renewable energy applications, particularly in maximum power point tracking (MPPT) for solar photovoltaic (PV) systems. Azad *et al*. (2023) proposed this approach to address the issue of tracking efficiency under rapidly changing irradiance. Their simulations demonstrated a 99.3% tracking efficiency, significantly outperforming both P&O and standard PSO in terms of convergence speed and power extraction accuracy [19]. In UAV path planning, Haris *et al*. (2024) showed PSO-AWDV offers greater stability than hyperbolic tangent PSO when navigating dynamic obstacles and wind gusts, at the cost of a 15 % increase in per-iteration computation due to velocity-memory reuse [22].

C. Applications of Murmuration PSO

Murmuration PSO variants have been applied in decentralized network and diagnostic tasks. Bhardwaj *et al*. (2023) built a secure, energy-efficient routing protocol for Flying Ad-Hoc Networks, reducing packet delays by 20 % through localized swarm interactions [23]. Minic *et al*. (2023) optimized RNN hyperparameters for ECG anomaly detection, improving F1-score by 10 % over canonical PSO, and Bharti *et al*. (2023) leveraged murmuration-based dynamics for image encryption, outperforming PSO, MVO, and GWO on entropy and NPCR metrics [24], [25].

D. Applications of Binary PSO

Binary PSO has emerged as a powerful tool in high-dimensional discrete optimization. Fatahi *et al*. (2024) used a hybrid BPSO for feature selection in COVID-19 clinical datasets, achieving over 97% classification accuracy using only 25% of the original feature space. The reduced computational overhead made it a viable tool for real-time diagnosis support systems [26]. Nadimi-Shahraki *et al*. (2022) proposed a binary murmuration optimizer for cardiology data classification, leveraging multiple binary sub-swarms. The algorithm showed high convergence efficiency while preserving the interpretability of selected features [27]. However, these gains come with extra hyperparameters, increased memory or neighborhood-computation costs, and up to 20 % higher per-iteration runtime. Furthermore, rigorous convergence proofs for time-varying and social-topology updates remain an open challenge, limiting theoretical guarantees.

VII. Discussion

The comparative analysis underscores a trade-off between algorithmic simplicity and domain-specific performance. Canonical PSO remains appealing for its minimal tuning and low runtime, but adaptive variants—like VVS-PSO's time-scheduled weights or PSO-AWDV's delayed-memory dynamics—deliver up to 4-times faster convergence in high-dimensional or dynamic environments. Topology-driven methods (Murmuration PSO) and binary decision rules extend PSO's reach into decentralized systems and discrete tasks. However, these gains come with extra hyperparameters, increased memory or neighborhood-computation costs, and up to 20% higher per-iteration runtime. Furthermore, rigorous convergence proofs for time-varying and social-topology updates remain an open challenge, limiting theoretical guarantees.

Looking ahead, two avenues are especially promising. First, hybridizing PSO with complementary metaheuristics or machine-learning–based auto-tuning can reduce manual parameter setting and dynamically adjust search behavior. Second, large-scale benchmarking on realistic, noisy, non-stationary problems—such as modular robotics control or real-time logistics—will validate robustness under practical constraints. Simultaneously, advancing theoretical analysis of convergence for adaptive and neighborhood-based PSO variants will clarify the conditions under which these methods reliably succeed.

VIII. Conclusion

This survey has traced PSO's evolution from its simple, minimally tuned canonical form to four key enhancements: VVS-PSO (time-varying inertia and learning rates), PSO-AWDV (delayed velocity memory), Murmuration-PSO (neighborhood-based interactions), and Binary PSO (discrete search space handling). By modifying the core velocity–position updates, these variants address PSO's traditional weaknesses—premature convergence, loss of diversity, and poor adaptability in noisy or non-stationary environments.

Comparative studies across both synthetic benchmarks and real-world tasks—structural health monitoring, solar maximum power point tracking, multi-agent coordination, and feature selection—confirm that enhanced PSO variants deliver up to 4-times faster convergence, higher solution quality, and reliable scaling beyond 150 dimensions. For example, PSO-AWDV achieves sub-second convergence in MPPT under fluctuating irradiance, while Murmuration-PSO boosts coordination success in robotic swarms by over 15 %.

While these adaptations yield substantial performance gains, they also introduce additional hyperparameters and computational overhead. Automated parameter-control methods and hybrid metaheuristic frameworks hold promise for creating self-tuning, scalable PSO schemes without extensive manual tuning.

Rigorous benchmarking on large-scale, noisy, and time-varying problem instances will be essential to validate the robustness of these variants under realistic conditions. Likewise, advancing theoretical convergence analyses—particularly for time-varying parameters and neighborhood-based topologies—will underpin more generalizable design principles. Practitioners should match PSO variants to their application needs: adaptive models (VVS-PSO, PSO-AWDV) for high-dimensional or dynamic problems, Murmuration-PSO for decentralized or multi-modal landscapes, and Binary PSO for discrete optimization and feature selection. This balanced perspective guides the selection and tailoring of PSO variants to meet contemporary optimization challenges.

References

[1] W. Zhang, X. Gu, L. Tang, Y. Yin, D. Liu, and Y. Zhang, "Application of machine learning, deep learning and optimization algorithms in geoengineering and geoscience: Comprehensive review and future challenge," *Gondwana Research*, vol. 109, pp. 1–17, Sep. 2022, https://doi.org/10.1016/j.gr.2022.03.015.

[2] E.-G. Talbi, "Machine Learning into Metaheuristics," *ACM Computing Surveys*, vol. 54, no. 6, pp. 1–32, 2022, https://doi.org/10.1145/3459664.

[3] J. Nayak, H. Swapnarekha, B. Naik, G. Dhiman, and S. Vimal, "25 Years of Particle Swarm Optimization: Flourishing Voyage of Two Decades," *Archives of Computational Methods in Engineering*, vol. 30, no. 3, pp. 1663–1725, 2023, https://doi.org/10.1007/s11831-022-09849-x.

[4] A. G. Gad, "Particle Swarm Optimization Algorithm and Its Applications: A Systematic Review," *Archives of Computational Methods in Engineering*, vol. 29, no. 5, pp. 2531–2561, 2022, https://doi.org/10.1007/s11831-021-09694-4.

[5] L. Abualigah, "Particle Swarm Optimization: Advances, Applications, and Experimental Insights," *Computers, Materials & Continua*, vol. 82, no. 2, pp. 1539–1592, 2025, https://doi.org/10.32604/cmc.2025.060765.

[6] T. M. Shami, A. A. El-Saleh, M. Alswaitti, Q. Al-Tashi, M. A. Summakieh, and S. Mirjalili, "Particle Swarm Optimization: A Comprehensive Survey," *IEEE Access*, vol. 10, pp. 10031–10061, 2022, https://doi.org/10.1109/ACCESS.2022.3142859.

[7] L. Abualigah *et al.*, "Particle swarm optimization algorithm: review and applications," in *Metaheuristic Optimization Algorithms*, pp. 1–14, 2024, https://doi.org/10.1016/B978-0-443-13925-3.00019-4.

[8] K. Rajwar, K. Deep, and S. Das, "An exhaustive review of the metaheuristic algorithms for search and optimization: taxonomy, applications, and open challenges," *Artificial Intelligence Review*, vol. 56, no. 11, pp. 13187–13257, 2023, https://doi.org/10.1007/s10462-023-10470-y.

[9] W. H. Bangyal, A. Hameed, W. Alosaimi, and H. Alyami, "A New Initialization Approach in Particle Swarm Optimization for Global Optimization Problems," *Computational Intelligence and Neuroscience*, vol. 2021, no. 1, 2021, https://doi.org/10.1155/2021/6628889.

[10] J. Zhang, J. Sheng, J. Lu, and L. Shen, "UCPSO: A Uniform Initialized Particle Swarm Optimization Algorithm with Cosine Inertia Weight," *Computational Intelligence and Neuroscience*, vol. 2021, no. 1, 2021, https://doi.org/10.1155/2021/8819333.

[11] T. M. Shami, S. Mirjalili, Y. Al-Eryani, K. Daoudi, S. Izadi, and L. Abualigah, "Velocity pausing particle swarm optimization: a novel variant for global optimization," *Neural Computing and Applications*, 2023, https://doi.org/10.1007/s00521-022-08179-0.

[12] E. Twumasi, E. A. Frimpong, N. K. Prah, and D. B. Gyasi, "A novel improvement of particle swarm optimization using an improved velocity update function based on local best murmuration particle," *Journal of Electrical Systems and Information Technology*, vol. 11, no. 1, p. 42, 2024, https://doi.org/10.1186/s43067-024-00168-8.

[13] A.-D. Li, B. Xue, and M. Zhang, "Improved binary particle swarm optimization for feature selection with new initialization and search space reduction strategies," *Applied Soft Computing*, vol. 106, p. 107302, 2021, https://doi.org/10.1016/j.asoc.2021.107302.

[14] E. H. Houssein, A. G. Gad, K. Hussain, and P. N. Suganthan, "Major Advances in Particle Swarm Optimization: Theory, Analysis, and Application," *Swarm and Evolutionary Computation*, vol. 63, p. 100868, 2021, https://doi.org/10.1016/j.swevo.2021.100868.

[15] S. Nama, A. K. Saha, S. Chakraborty, A. H. Gandomi, and L. Abualigah, "Boosting particle swarm optimization by backtracking search algorithm for optimization problems," *Swarm and Evolutionary Computation*, vol. 79, p. 101304, 2023, https://doi.org/10.1016/j.swevo.2023.101304.

[16] A. Ashraf, A. Ali Almazroi, W. Haider Bangyal, and M. A. Alqarni, "Particle Swarm Optimization with New Initializing Technique to Solve Global Optimization Problems," *Intelligent Automation & Soft Computing*, vol. 31, no. 1, pp. 191–206, 2022, https://doi.org/10.32604/iasc.2022.015810.

[17] M. Jain, V. Saihjpal, N. Singh, and S. B. Singh, "An Overview of Variants and Advancements of PSO Algorithm," *Applied Sciences*, vol. 12, no. 17, p. 8392, 2022, https://doi.org/10.3390/app12178392.

[18] H.-L. Minh, S. Khatir, R. V. Rao, M. Abdel Wahab, and T. Cuong-Le, "A variable velocity strategy particle swarm optimization algorithm (VVS-PSO) for damage assessment in structures," *Engineering with Computers*, vol. 39, no. 2, pp. 1055–1084, 2023, https://doi.org/10.1007/s00366-021-01451-2.

[19] M. A. Azad *et al.*, "A Particle Swarm Optimization–Adaptive Weighted Delay Velocity-Based Fast-Converging Maximum Power Point Tracking Algorithm for Solar PV Generation System," *Sustainability*, vol. 15, no. 21, p. 15335, 2023, https://doi.org/10.3390/su152115335.

[20] S. Tijjani, M. N. Ab Wahab, and M. H. Mohd Noor, "An enhanced particle swarm optimization with position update for optimal feature selection," *Expert Systems with Applications*, vol. 247, p. 123337, 2024, https://doi.org/10.1016/j.eswa.2024.123337.

[21] X. Liu, B. Niu, and W. Yi, "A Modified Variable Velocity Strategy Particle Swarm Optimization Algorithm for Multi-objective Feature

Selection," *Advances in Swarm Intelligence*, pp. 46–57, 2024, https://doi.org/10.1007/978-981-97-7181-3_4.

[22] M. Haris, D. M. S. Bhatti, and H. Nam, "A Fast-Convergent Hyperbolic Tangent PSO Algorithm for UAVs Path Planning," *IEEE Open Journal of Vehicular Technology*, vol. 5, pp. 681–694, 2024, https://doi.org/10.1109/OJVT.2024.3391380.

[23] V. Bhardwaj, D. Prashar, and M. Rashid, "SIRFRM-Secure and Intelligent Routing Fairness Rate Model for Flying Adhoc Networks," *Mobile Networks and Applications*, pp. 1-11, 2023, https://doi.org/10.1007/s11036-023-02270-z.

[24] A. Minic *et al.*, "Applying Recurrent Neural Networks for Anomaly Detection in Electrocardiogram Sensor Data," *Sensors*, vol. 23, no. 24, p. 9878, 2023, https://doi.org/10.3390/s23249878.

[25] V. Bharti, B. Biswas, and K. K. Shukla, "MDO: a novel murmuration-flight based dispersive optimization algorithm and its application to image security," *Journal of Ambient Intelligence and Humanized Computing*, vol. 14, no. 5, pp. 4809–4826, 2023, https://doi.org/10.1007/s12652-023-04537-5.

[26] A. Fatahi, M. H. Nadimi-Shahraki, and H. Zamani, "An Improved Binary Quantum-based Avian Navigation Optimizer Algorithm to Select Effective Feature Subset from Medical Data: A COVID-19 Case Study," *Journal of Bionic Engineering*, vol. 21, no. 1, pp. 426–446, 2024, https://doi.org/10.1007/s42235-023-00433-y.

[27] M. H. Nadimi-Shahraki, Z. Asghari Varzaneh, H. Zamani, and S. Mirjalili, "Binary Starling Murmuration Optimizer Algorithm to Select Effective Features from Medical Data," *Applied Sciences*, vol. 13, no. 1, p. 564, 2022, https://doi.org/10.3390/app13010564.