A I S ARTIFICIAL INTELLIGENCE STUDIES

L-PFO: Enhancing Polar Fox Optimization with Linear Population Size Reduction for Benchmark Problems

Mehmet Ali DURSUN[*a], **Gürcan YAVUZ**[b]

[a,*] Dumlupınar University,
Engineering Faculty,
Dept. of Computer Engineering
43000 - Kütahya, Türkiye
ORCID: 0000-0001-6370-1160

[b] Dumlupınar University,
Engineering Faculty,
Dept. of Computer Engineering
43000 - Kütahya, Türkiye
ORCID: 0000-0002-2540-1930

[*] Corresponding author.
e-mail: mehmet.dursun@dpu.edu.tr

ABSTRACT

This study presents an improved variant of the Polar Fox Optimization (PFO) algorithm, named L-PFO, by incorporating the Linear Population Size Reduction (LPSR) mechanism. The original PFO algorithm, which is inspired by the adaptive survival strategies of Arctic wildlife in harsh environmental conditions, exhibits strong capabilities in exploring the search space. However, its performance is notably inconsistent in high-dimensional and multimodal optimization problems due to limitations in maintaining a balanced trade-off between exploration and exploitation. To address these shortcomings, the L-PFO algorithm dynamically adjusts the population size during the optimization process through a linear reduction strategy, thereby promoting convergence stability and refining local search efficiency. The proposed algorithm is empirically validated using eight well-known benchmark test functions that reflect varying levels of complexity and modality. Performance metrics including best, average, and standard deviation values are calculated over 30 independent runs to assess the algorithm's robustness and generalization capacity. The experimental results demonstrate that the L-PFO algorithm consistently outperforms the original PFO in terms of convergence speed, solution accuracy, and stability across most test functions. In particular, significant improvements are observed on challenging functions where the original PFO algorithm struggled with premature convergence or high variance among runs. The integration of LPSR enhances the algorithm's adaptability and resilience against local optima traps, making it more suitable for complex optimization tasks. Overall, the proposed L-PFO algorithm provides a more reliable and scalable metaheuristic framework with minimal parameter dependency, indicating its potential applicability in various real-world engineering and computational optimization problems.

Keywords: Polar Fox Optimization, Linear Population Size Reduction, Metaheuristic Algorithms, Benchmark Functions, Global Optimization, Algorithmic Robustness

Anahtar Kelimeler: Polar Tilki Optimizasyonu, Lineer Popülasyon Azaltımı, Meta-sezgisel Algoritmalar, Kıyaslama Fonksiyonları, Küresel Optimizasyon, Algoritmik Dayanıklılık

Submitted: 28.06.2025
Revised: 27.12.2025
Accepted: 29.12.2025

doi:10.30855/ais.2025.08.02.01

L-PFO: Karmaşık Benchmark Problemleri İçin Lineer Popülasyon Azaltımı ile Güçlendirilmiş Polar Tilki Optimizasyonu

ÖZ

Bu çalışma, Polar Fox Optimization (PFO) algoritmasının geliştirilmiş bir versiyonu olan L-PFO algoritmasını önermektedir. Geliştirilen bu yeni algoritma, Lineer Popülasyon Büyüklüğü Azaltımı (LPSR) mekanizmasını entegre ederek orijinal yapının sınırlılıklarını aşmayı hedeflemektedir. Zorlu çevresel koşullara uyum sağlayan kutup hayvanlarının biyolojik adaptasyonlarından esinlenerek tasarlanan PFO algoritması, arama uzayında keşif kabiliyeti bakımından güçlü bir potansiyel sunsa da, özellikle yüksek boyutlu ve çok modlu optimizasyon problemlerinde tutarsız performans ve erken yakınsama sorunları ile karşılaşmaktadır. Bu bağlamda önerilen L-PFO algoritması, optimizasyon süreci boyunca popülasyon boyutunu doğrusal olarak azaltarak yakınsama kararlılığını artırmakta ve yerel arama etkinliğini geliştirmektedir. Geliştirilen algoritma, karmaşıklık düzeyleri ve çokluluğu farklılık gösteren sekiz standart test fonksiyonu ile deneysel olarak değerlendirilmiştir. Algoritmanın sağlamlık ve genelleme kabiliyetini analiz etmek amacıyla, her bir test fonksiyonu için en iyi, ortalama ve standart sapma değerleri 30 bağımsız çalıştırma üzerinden hesaplanmıştır. Elde edilen sonuçlar, L-PFO algoritmasının çoğu test fonksiyonunda orijinal PFO'ya kıyasla daha hızlı yakınsama, daha yüksek çözüm doğruluğu ve daha istikrarlı performans sergilediğini ortaya koymaktadır. Özellikle, orijinal algoritmanın erken yakınsama veya yüksek varyans gösterdiği karmaşık fonksiyonlarda belirgin performans iyileştirmeleri gözlemlenmiştir. LPSR mekanizmasının entegrasyonu, algoritmanın yerel minimumlara takılmadan çözüm uzayında daha etkin bir şekilde gezinmesini sağlamış, böylece karmaşık optimizasyon problemleri karşısında daha uyarlanabilir ve güvenilir bir yapı sunmuştur. Genel olarak, önerilen L-PFO algoritması, düşük parametre bağımlılığına sahip, daha kararlı ve ölçeklenebilir bir meta-sezgisel çerçeve sunarak mühendislik ve hesaplamalı optimizasyon alanlarında geniş bir uygulama potansiyeline sahip olduğunu göstermektedir.

To cite this article: M.A. Dursun and G. Yavuz, "L-PFO: Enhancing Polar Fox Optimization with Linear Population Size Reduction for Complex Benchmark Problems," *Artificial Intelligence Studies*, vol.8, no.2, pp. 100-115, 2025. doi:10.30855/ais.2025.08.02.01

1. Introduction (Giriş)

The inherent complexity of nature and the adaptive mechanisms observed in biological systems have long served as an intellectual foundation for computational model development[1]. Inspiration drawn from living organisms has facilitated the design of algorithms capable of solving intricate optimization problems[2]. Behavioral strategies demonstrated by various animal species in the context of survival and resource acquisition have played a critical role in this conceptual transfer[3]. These biologically inspired systems have contributed to the advancement of algorithmic methodologies by translating natural behaviors into heuristic search mechanisms[4]. Furthermore, insights obtained from ecological patterns, environmental dynamics, and physical phenomena have further enriched algorithmic frameworks[5]. Atmospheric circulation, wave propagation, and migratory patterns have supplemented biological analogies in modeling diverse problem domains[6].

In the field of optimization, evolutionary algorithms have emerged as powerful tools that mimic the search and adaptation behaviors observed in nature[7]. Recent developments have introduced a new class of metaheuristic algorithms that exhibit strong performance in large-scale, multidimensional, and nonlinear search spaces[8]. These algorithms are designed to balance exploration and exploitation, often by replicating complex interactions in natural ecosystems[9]. Among these, the Polar Fox Optimization (PFO) algorithm represents a novel approach inspired by the behavioral patterns of Arctic foxes[10]. The survival strategies employed by these animals in extreme environmental conditions constitute the core of the algorithm's structure.

The model abstracts key traits such as distributed search, adaptive movement, and coordinated behavior within a population. It simulates the ability of individuals to share information, navigate challenging terrains, and follow dynamic leadership. These principles are systematically encoded to support robust search performance across complex landscapes[11]. The PFO algorithm seeks to enhance global search capabilities by emphasizing collaborative decision-making and decentralized intelligence. Its architecture reflects an integration of biological reasoning with computational efficiency. Although the initial design demonstrates promise, the algorithm still offers potential for refinement. Ongoing studies are exploring its hybridization with complementary methods to expand its utility and adaptability.

This research contributes to the growing body of work in bio-inspired optimization by investigating the theoretical and practical aspects of PFO. It provides a comprehensive framework for assessing its algorithmic behavior, structural components, and application potential. The integration of nature-inspired intelligence into algorithmic systems continues to open new avenues for addressing complex problems in science and engineering[4].

1.1. Nature-Inspired Metaheuristic Algorithms

Meta-heuristic optimization algorithms, particularly those inspired by nature, have gained significant attention in the last two decades due to their ability to effectively solve complex and nonlinear optimization problems across diverse domains[12]. These methods mimic various biological and physical phenomena, enabling them to strike a balance between exploration and exploitation and to escape from local optima in high-dimensional search spaces.

One of the most comprehensive surveys in this domain is provided by José-García and Gómez-Flores[13], who categorized a wide range of nature-inspired metaheuristics used in automatic clustering problems. Their study emphasizes the growing relevance of both single-objective and multi-objective strategies in dealing with datasets of unknown cluster numbers, diverse distributions, and high dimensionality. The review shows how nature-inspired strategies, such as particle swarm optimization, differential evolution, ant colony optimization, and artificial bee colony, have been tailored for unsupervised learning and clustering tasks.

In a more recent development, Ghiaskar et al.[10] introduced the Polar Fox Optimization Algorithm (PFA), a novel meta-heuristic based on the highly adaptive hunting and foraging behavior of arctic foxes. The algorithm mathematically models the fox's high-frequency hearing and jumping strategies, offering an efficient search mechanism that has been validated on a comprehensive set of classical and CEC 2021 benchmark functions[14], as well as real-world engineering optimization problems. Their results demonstrated competitive performance with minimal control parameters, making PFA a promising alternative to traditional and recent metaheuristics.

Complementing these insights, studies such as those by Wang et al.[15] and Braik et al.[16] have further advanced the field by proposing hybrid and improved variants of nature-inspired algorithms tailored for different engineering and classification problems. These approaches typically enhance convergence rate, robustness, and parameter adaptivity, which are critical for complex real-world applications like scheduling, design optimization, and image segmentation

1.2. Population Size Reduction Techniques

Linear Population Size Reduction (LPSR) is a deterministic control strategy that progressively decreases the population size during the optimization process. This mechanism was first introduced in the context of Differential Evolution (DE) through the L-SHADE algorithm by Tanabe and Fukunaga[17]. Their study demonstrated that incorporating LPSR into the SHADE framework improved convergence performance significantly, particularly on the CEC 2014 benchmark set. The LPSR mechanism was shown to maintain exploration capabilities in the early stages and intensify exploitation as the search progresses. This balance resulted in better global search behavior with fewer control parameters.

Further investigation by Hussien et al.[18] extended LPSR into a hybrid model by combining SHADE with Whale Optimization Algorithm components. This hybridization, empowered by LPSR, yielded promising results across constrained and unconstrained engineering problems. The authors emphasized that deterministic population control enhanced the stability of search dynamics in high-dimensional scenarios.

A subsequent study by Braik et al.[16] explored LPSR within a deep transfer learning context, particularly for cross-domain visual learning tasks. Their framework applied LPSR-based SHADE to fine-tune network parameters and demonstrated competitive accuracy and generalization across several datasets. This work highlighted the flexibility of LPSR beyond traditional numerical optimization.

The study by Liu et al.[19] focused on adapting LPSR within a surrogate-assisted evolutionary optimization framework. The authors integrated LPSR into the selection and resampling phases of the model, demonstrating its efficacy in managing search pressure under computational constraints. The deterministic reduction was noted to improve convergence without requiring additional hyperparameter tuning.

Zeng et al.[20] provided a broader analysis of control strategies in differential evolution and benchmarked various approaches, including LPSR. Their results confirmed the robustness of linear reduction methods under dynamic search conditions, particularly when population diversity must be preserved across varying objective landscapes.

1.3. Benchmark Test Functions

Benchmark test functions are widely used in computational intelligence and global optimization literature as a standardized means to evaluate the effectiveness, robustness, and generalization capabilities of metaheuristic algorithms. These functions provide a controlled environment with known global minima, well-defined landscapes, and varying complexities that reflect real-world optimization challenges. Their primary utility lies in objectively assessing an algorithm's capacity to avoid premature convergence, explore multimodal landscapes, and converge efficiently toward the global optimum across different types of problem topologies.

In the study conducted by Abbas et al.[21], several benchmark functions were employed to evaluate the performance of Particle Swarm Optimization (PSO). The authors demonstrated that the PSO algorithm achieved rapid convergence on unimodal and convex functions like Sphere, while struggling with multimodal surfaces such as Rastrigin due to its high density of local optima. Their findings emphasized the influence of parameter tuning and swarm diversity in navigating complex search spaces effectively.

Similarly, Radwan et al.[22] introduced a Shapley value-guided swarm optimizer (SGSO) and assessed its behavior on functions including Ackley and Easom. These functions, characterized by steep valleys and deceptive global optima, served as robust test beds to evaluate SGSO's adaptive search dynamics. Their experiments revealed that the Shapley value mechanism enables better prioritization of decision variables, thereby improving search directionality and reducing stagnation, especially in high-dimensional multimodal landscapes.

In a separate investigation, Wang et al.[23] explored the performance of a multi-objective optimization framework built upon Student's t-distribution on various benchmark functions. Their analysis focused on the convergence trade-offs between conflicting objectives, demonstrating the importance of test functions in quantifying Pareto-optimality and algorithmic balance across solution sets

1.4. PFO-Derived Methods

Recent advancements in nature-inspired optimization have led to the emergence of the Polar Fox Optimization (PFO) algorithm, a swarm-based metaheuristic designed to simulate the adaptive and hunting behaviors of Arctic foxes. Since its introduction, PFO and its variants have been applied across a broad range of domains due to their adaptability and low parameter dependency.

Albdour and Alkhatib[24] proposed a hybrid Wavelet–Polar Fox Optimization Algorithm (WPFOA) that integrates wavelet-based feature extraction with PFO for malware image classification. Their method achieved significant improvements in detection accuracy (97.32%), demonstrating PFO's strong potential in feature selection for cybersecurity applications. Ghiaskar et al.[10] originally introduced the standard PFO by mathematically modeling the high-frequency auditory and jumping behavior of Arctic foxes. Their experiments across classical and CEC benchmark functions showed that PFO, despite its simplicity, offered competitive performance with minimal control parameters.

Despite its promise, some limitations have been noted in the convergence stability and solution diversity of the baseline PFO. As a response, researchers have started combining PFO with deep learning architectures and hybrid search strategies. The WPFOA introduced by Albdour[24] incorporated wavelet transforms for texture-based malware recognition, highlighting how domain-specific hybridization can improve robustness in complex data environments.

Collectively, these studies confirm that PFO and its derivatives are rapidly evolving as efficient metaheuristic frameworks. Their integration with signal processing, energy optimization, and AI-based detection systems marks a significant trend in the development of lightweight and generalizable optimization tools.

The reviewed literature demonstrates the rapid evolution of nature-inspired metaheuristic algorithms, both in theory and application. This evolution has been driven by hybridization techniques, adaptive parameter control, and population size modulation strategies, all of which enhance algorithmic robustness across domains such as clustering, engineering optimization, biomedical systems, and image processing. In particular, Linear Population Size Reduction (LPSR) has emerged as a powerful mechanism to balance convergence and exploration, promoting adaptivity and solution diversity. Meanwhile, benchmark test functions continue to serve as indispensable tools for evaluating generalization performance and stability across diverse problem landscapes. Among these algorithms, Polar Fox Optimization (PFO) has drawn attention due to its simple parameter structure and biologically inspired modeling. However, prior studies have indicated that PFO struggles with multimodal and high-dimensional functions, often exhibiting inconsistent convergence behavior.

In response to these gaps, this study proposes an enhanced version of PFO called L-PFO which incorporates the LPSR mechanism into the original algorithmic framework. This integration not only addresses key methodological deficiencies in existing PFO literature but also introduces a low-complexity, computationally efficient approach with improved convergence depth and resilience across standard test environments. The proposed method achieves a compelling balance between algorithmic simplicity and performance sophistication, thereby offering a novel and integrative contribution to the metaheuristic optimization literature.

2. Material and Methods (Materyal ve Yöntem)

2.1. Inspired

Despite the harsh conditions of the Arctic, the survival strategies of animals inhabiting this region reveal remarkable adaptive capabilities[25]. These ecosystems are defined by extreme environmental stressors such as heavy snowfall, low temperatures, strong winds, and prolonged winter nights. The persistence of life under such conditions demonstrates the functional complexity of biological adaptation. Among the most critical challenges is the need to conserve body heat and secure sufficient food sources. The survival mechanisms developed in response to these challenges have long inspired the design of bioinspired computational models[26].
Arctic foxes, native to polar regions across Europe, Asia, and North America, exemplify these adaptations. Their physical features include compact bodies, thick insulating fur, and reduced extremities that minimize heat loss. These morphological traits enable survival in temperatures that can drop as low as -70°C. Seasonal camouflage and behavioral flexibility further support their adaptation to environmental variability.

The hunting behavior of Arctic foxes involves high spatial awareness and sensory precision. These animals can locate prey concealed beneath thick snow layers by detecting low-frequency sounds. After determining the prey’s location, the fox adjusts its position before initiating a swift and direct capture movement. This sequence combines acoustic perception, positional accuracy, and rapid motor response. Such behaviors reflect the interaction between environmental cues and internal coordination strategies.

From a computational standpoint, the strategic movement and sensory alignment exhibited during the hunting process can be abstracted into algorithmic mechanisms. The fox’s search dynamics—characterized by careful detection and sudden action—serve as conceptual references for exploration and exploitation trade-offs in optimization algorithms. These insights contribute to the formulation of models capable of navigating complex solution landscapes with improved efficiency.

In this study, the Polar Fox Optimization (PFO) algorithm has been adopted as the base structure. Rather than modifying its fundamental components, the current work aims to improve algorithmic efficiency by incorporating a population control mechanism inspired by the L-SHADE algorithm. Specifically, the Linear Population Size Reduction (LPSR) strategy is integrated into the standard PFO framework. This adaptation allows for gradual reduction of the search agents over time, thereby promoting computational efficiency while maintaining solution quality. The integration is designed to enhance the convergence characteristics of PFO without altering its core behavioral logic. By introducing LPSR into the exploration phase, the algorithm can adjust its search intensity in response to the problem's complexity. As a result, the refined version of PFO is expected to deliver improved performance in high-dimensional and dynamic optimization scenarios.

2.2. Mathematical Model

The Polar Fox Optimization (PFO) algorithm is classified as a bio-inspired metaheuristic technique derived from the adaptive behaviors of Arctic foxes. It draws on the species’ abilities to survive, hunt, and operate in groups under harsh polar conditions. These natural mechanisms are mathematically encoded to support a balance between exploration and exploitation in high-dimensional optimization problems. The algorithm is organized into multiple phases, each simulating a distinct behavioral component observed in the species’ ecology.

2.2.1. Population Initialization

The search process begins with the generation of a random population across the problem space. Each candidate solution is represented as a vector within a matrix X, where each row denotes an individual and each column a problem dimension. The initial position xij for the j-th dimension of the i-th individual is generated using Eqution 1.

$$x_{ij} = \mathrm{LB} + r_1 \cdot (\mathrm{UB} - \mathrm{LB}) \quad (1)$$

Here, LB and UB define the lower and upper bounds of the search space, and r1 is a uniform random value in the range [0, 1].

2.2.2. Group Assignment and Weight Update

The population is divided into four behavioral groups G1,G2,G3,G4 each corresponding to a unique search pattern. Initially balanced in size, these groups adapt based on performance through an iterative weight update process. The updated weight W_i for group i is computed as Equation 2.

$$W_i^{\text{new}} = W_i + t^2/\text{NG}_i \tag{2}$$

2.2.3. Experience-Based Movement

Each individual adjusts its position based on its historical experience. This phase simulates the fox's acoustic localization of prey beneath the snow and subsequent vertical attack. The position is updated as Equation 3. Each individual's new position ${x_i}^{t+1}$ is determined by Equation 3, which uses the jump force P and direction D.

$${x_i}^{t+1} = {x_i}^t + P \cdot D \tag{3}$$

he jump force P is calculated using Equation 4, involving the random value r_2 and the experience-based strength PF_i.

$$P = r_2 \cdot \text{PF}_i \tag{4}$$

The direction D is defined in Equation 5 as the cosine of a randomly generated angle r_3.

$$D = \cos(r_3) \tag{5}$$

2.2.4. Leader-Guided Adjustment

In each group, the member with the best performance is assigned as the leader. Other group members update their positions relative to the leader's guidance in Equation 6.

$${x_i}^{t+1} = {x_i}^t + r_4 \cdot \frac{({x_i}^t - L)}{\text{LF}_i} \tag{6}$$

where L is the leader's current position, LF_i is the leader's influence factor, and $r_4 \in [-1,1]$ is a uniformly distributed random variable. Leader fitness is monitored continuously, and updates are triggered based on energy thresholds.

2.2.5. Leader Motivation Phase

When a group shows signs of stagnation, such as frequent failures or lack of improvement, a diversification mechanism is activated. Individuals are redistributed to new positions to escape local optima. The condition for activation is defined in Equation 7.

$$\text{critical} = (\text{NLM}[\text{MLM}]) \vee (\frac{t}{\text{NI}} \geq 0.8) \tag{7}$$

where NLM is the leader's motivation level, MLM is a predefined threshold, t is the current generation, and NI is the maximum number of iterations.

2.2.6. Mutation Phase

Underperforming individuals are periodically replaced with new random candidates to enhance diversity. New positions are assigned using Equation 8.

$$x_i = \mathrm{LB} + r_5 \cdot (\mathrm{UB} - \mathrm{LB}) \tag{8}$$

With $r_5 \in [0,1]$. This phase is selectively applied when performance stagnation is detected.

2.2.7. Fatigue Simulation

After each iteration, individuals' energy levels are reduced to simulate biological fatigue. This affects their search behavior and group activity. The update for group G_1 is given in Equation 9.

$$G_1 = \min\,(G_1 - G_{1r}, G_{1i}) \tag{9}$$

If the group size drops below 10% of the total population, the group's energy is reset using initial parameters in Equation 10.

$$G_1 = G_{1m} \tag{10}$$

This adjustment ensures continuous participation and prevents group collapse. Similar mechanisms are applied to the remaining groups.

The PFO algorithm integrates a sequence of biologically inspired processes that collectively enhance its global and local search capabilities. In this study, the original structure is retained, and its adaptability is extended by incorporating the Linear Population Size Reduction (LPSR) mechanism from L-SHADE. This addition enables controlled population scaling and supports efficient convergence without altering the core behavioral logic of the algorithm.

2.3. Benchmark Functions

All benchmark functions considered in this study were evaluated in a 30-dimensional search space. This dimensionality is commonly adopted in the optimization literature to assess algorithm robustness and scalability under increased problem complexity.

2.3.1. Ackley Function

The Ackley function is known for its complex landscape filled with numerous local minima. It is commonly used to test an algorithm's global search capability. Its surface combines an exponential decay with a cosine modulation, creating a flat outer region and a steep global basin. Mathematical model given in Equation 11.

$$f_1(\mathbf{x}) = -20\exp\left(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} {x_i}^2}\right) - \exp\left(\frac{1}{D}\sum_{i=1}^{D}\cos\,(2\pi x_i)\right) + 20 + e \tag{11}$$

2.3.2. Sphere Function

The Sphere function is a simple, convex, and continuous function with a single global minimum. It is often used to evaluate the exploitation ability of algorithms. Due to its smooth and symmetric shape, it allows clear observation of convergence behavior. Mathematical model given in Equation 12.

$$f_2(\mathbf{x}) = \sum_{i=1}^{d} x^2 \tag{12}$$

2.3.3. Schwefel Function

The Schwefel function introduces significant complexity with its deep valleys and deceptive local optima. It is non-convex and multimodal, with the global minimum located far from the origin. Its landscape challenges the algorithm's ability to escape suboptimal regions. Mathematical model given in Equation 13.

$$f_3(\mathbf{x}) = 418.9829 \cdot d - \sum_{i=1}^{d} x \cdot sin(\sqrt{|x|}) \tag{13}$$

2.3.4. Griewank Function

The Griewank function combines a quadratic term with a cosine-based modulation. This function is characterized by a large number of regularly distributed local minima. It is designed to test the algorithm's performance on moderately complex multimodal surfaces. Mathematical model given in Equation 14.

$$f_4(\mathbf{x}) = 1 + \frac{1}{4000}\sum_{i=1}^{d} x^2 - \prod_{i=1}^{d} cos(\frac{x}{\sqrt{i}}) \tag{14}$$

2.3.5. Dixon–Price Function

The Dixon–Price function is designed to test an algorithm’s capacity to handle non-separable variables. It includes quadratic terms and interaction between variables, which creates a highly curved landscape with a single global optimum. Mathematical model given in Equation 15.

$$f_5(\mathbf{x}) = (x1 - 1)^2 + \sum_{i=2}^{d} i \cdot (2x^2 - x_{i-1})^2 \tag{15}$$

2.3.6. Rosenbrock Function

The Rosenbrock function, often referred to as the Banana function, features a narrow curved valley. Although the global minimum lies within this valley, it is difficult to reach due to the flat curvature. This makes it a strong benchmark for testing local search efficiency. Mathematical model given in Equation 16.

$$f_6(\mathbf{x}) = \sum_{i=1}^{D-1} (100({x_i}^2 - x_{i+1})^2 + (x_i - 1)^2) \tag{16}$$

2.3.7. Rastrigin Function

The Rastrigin function is a highly multimodal benchmark with a large number of local minima regularly spaced across the domain. It combines a quadratic base with periodic cosine terms, testing an algorithm's ability to maintain diversity during optimization. Mathematical model given in Equation 17.

$$f_7(\mathbf{x}) = \sum_{i=1}^{D} ({x_i}^2 - 10\cos(2\pi x_i) + 10) \tag{17}$$

2.3.8. Zakharov Function

The Zakharov function includes both linear and quadratic components, and the global minimum is located in a smooth basin. It is continuous and unimodal but involves variable interactions that increase difficulty in higher dimensions. It is useful for testing scalability and convergence speed. Mathematical model given in Equation 18.

$$f_8(\mathbf{x}) = \sum_{i=1}^{D} x_i^2 + \left(\sum_{i=1}^{D} 0.5x_i\right)^2 + \left(\sum_{i=1}^{D} 0.5x_i\right)^4 \quad (18)$$

2.4. Proposed Method

This study adopts the Polar Fox Optimization (PFO) algorithm as the core framework due to its biologically inspired design and its ability to maintain a dynamic balance between global and local search. The PFO algorithm simulates the behavioral patterns of Arctic foxes, including group-based hunting, leader-guided coordination, and experience-driven spatial adjustment. These mechanisms are modeled through a series of computational stages, each of which contributes to adaptive exploration in complex solution landscapes.

While the original PFO algorithm demonstrates promising performance in high-dimensional problems, its static population structure imposes certain limitations. A fixed number of individuals throughout the optimization process can result in redundant computations and reduced efficiency during the later stages of convergence. To address this issue, the proposed method introduces a Linear Population Size Reduction (LPSR) mechanism into the PFO structure.

2.4.1. LPSR

In population-based optimization algorithms, the number of individuals directly influences both computational cost and search efficiency. Maintaining a large population may improve diversity but can result in slower convergence and higher resource usage. Conversely, reducing the population size too early may lead to premature convergence.

The Linear Population Size Reduction (LPSR) strategy addresses this trade-off by gradually decreasing the number of individuals over time. This mechanism begins with a larger population to encourage broad exploration in the early stages of the algorithm. As the number of generations increases, the population size is linearly reduced, shifting the focus from exploration to exploitation.

This progressive reduction is calculated based on the current generation and the predefined minimum and initial population sizes. By controlling the number of active individuals dynamically, the algorithm can maintain solution quality while reducing computational overhead.

LPSR is particularly effective in high-dimensional problems where the cost of evaluating many individuals becomes significant. In such cases, the ability to adapt population size allows the algorithm to operate more efficiently without compromising convergence behavior. When integrated with biologically inspired methods like the PFO algorithm, LPSR complements the behavioral dynamics by adding a scalable control layer to the population structure.

The use of LPSR also helps regulate the selection pressure and reduces redundancy in the solution pool. As the search progresses and fewer individuals remain, the algorithm naturally increases its focus on promising areas of the search space. This dynamic adaptation contributes to both robustness and efficiency across various problem types. Mathematical model given in Equation 19.

$$N(G) = N_{\min} + \left(\frac{G_{\max} - G}{G_{\max}}\right)(N_{\text{init}} - N_{\min}) \quad (19)$$

The LPSR strategy allows the population size to decrease progressively as the algorithm advances through iterations. In the early phases, a larger population promotes wide exploration of the search space and ensures high diversity. As the algorithm moves toward later stages, the population is linearly reduced, allowing computational resources to focus on more refined local searches. This dynamic adjustment provides a natural shift from global to local optimization behavior, which is essential for improving convergence speed and reducing unnecessary evaluations.

The integration of LPSR into PFO is designed to be non-invasive. It does not alter the internal logic or behavioral modeling of the original algorithm. Instead, it regulates the number of active individuals at each generation using a time-dependent update rule. The LPSR component relies on a mathematically defined schedule that computes the current population size as a function of the initial size, the minimum allowable size, and the current iteration. This ensures a smooth and controlled reduction in population without abrupt changes that could destabilize the search process.

By adapting the population dynamically, the enhanced PFO algorithm gains increased flexibility across varying problem scales. It also reduces computational overhead while maintaining the algorithm's inherent adaptiveness. The proposed modification is particularly useful in scenarios where the cost of evaluating candidate solutions is high or when the optimization domain is subject to gradual convergence. The flowchart of the proposed method is presented in Figure 1.

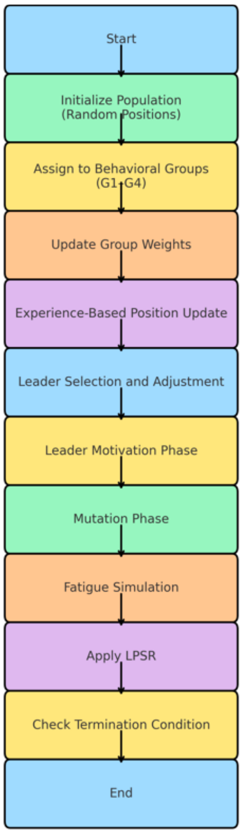

Figure 1. Flowchart of the proposed method based on PFO with Linear Population Size Reduction

2.4.2. Rationale for the Selection of LPSR Population Parameters

In this study, the initial population size ($NPinit$) and the minimum population size ($NPmin$) used in the Linear Population Size Reduction (LPSR) mechanism were not selected arbitrarily, but were determined in accordance with the original LPSR formulation introduced in the L-SHADE framework. In L-SHADE, LPSR is designed as a deterministic control strategy that starts with a relatively large population to promote extensive exploration in the early stages of the search, and then linearly reduces the population size to strengthen exploitation and convergence stability in later iterations.

Following this established principle, the initial population size was set to $NPinit$=40. This value provides sufficient population diversity for 30-dimensional benchmark problems, allowing the algorithm to explore the search space effectively while keeping the computational cost at a reasonable level. The minimum population size was defined as $NPmin$=10, which is consistent with commonly adopted lower bounds in LPSR-based algorithms and ensures that the search process does not lose its adaptive capability in the final optimization stages.

A limited sensitivity observation indicates that increasing the initial population beyond this level (e.g., $NPinit$>50) does not lead to a proportional improvement in convergence accuracy, but significantly increases computational overhead. Conversely, selecting a very small minimum population size ($NPmin$<8) was observed to increase the risk of premature convergence on highly multimodal functions. Therefore, the chosen values represent a balanced and literature-consistent compromise between exploration capability, exploitation efficiency, and computational cost.

Overall, the population parameters used for LPSR in this study strictly follow the original design rationale of LPSR rather than introducing additional tuning complexity. This choice ensures methodological clarity, reproducibility, and a fair integration of the LPSR mechanism into the original PFO framework without altering its core behavioral structure.

In summary, the proposed method retains the original behavioral components of the PFO algorithm and augments its efficiency through a simple yet effective LPSR mechanism. The resulting structure preserves biological plausibility while achieving improved computational performance and convergence reliability across a wide range of benchmark functions.

3. Results and Discussion (Sonuçlar ve Tartışma)

The experimental evaluation presented in this section aims to assess the performance, robustness, and convergence behavior of the proposed optimization method under a variety of benchmark scenarios. The approach builds upon the original Polar Fox Optimization (PFO) algorithm, enhancing its adaptability through the integration of a Linear Population Size Reduction (LPSR) mechanism. To determine the efficacy of this modification, extensive testing was conducted using well-established benchmark functions that represent diverse optimization challenges in terms of dimensionality, modality, separability, and landscape complexity.

In optimization research, benchmark-based validation is a critical step in verifying both the theoretical and practical viability of an algorithm. By applying the same evaluation criteria across multiple standard functions, it becomes possible to make meaningful comparisons between different strategies. The selected test functions in this study include a combination of unimodal and multimodal problems, as well as convex and non-convex surfaces. These include Ackley, Sphere, Schwefel, Griewank, Dixon–Price, Rosenbrock, Rastrigin, and Zakharov functions. Each function introduces distinct features such as deceptive local minima, interaction between decision variables, or sharp global basins—properties which test the depth and flexibility of the optimization process.

The proposed method, referred to as L-PFO, retains the behavioral modeling framework of PFO but introduces an adaptive reduction in population size as the search progresses. This addition is designed to optimize the balance between exploration and exploitation. In the early iterations, a larger population supports the identification of promising regions in the search space. As the algorithm continues, the population is gradually reduced, allowing more focused exploitation without excessive computational cost. This structural modification aims to address one of the limitations commonly observed in fixed-population metaheuristics, where redundant evaluations and stagnation may occur in the later stages of optimization.

3.1. Experimental Settings

All experiments were conducted under controlled conditions to ensure fairness and reproducibility. Both PFO and L-PFO were implemented under identical parameter settings, including population initialization methods, stopping criteria, and random seed policies. The population size (NP) was consistently set to 40, ensuring comparable search capabilities across runs. Additionally, the number of mutation individuals (NM) was fixed at 8 in all experiments to standardize the exploration component of the algorithm. The performance of each algorithm was measured across 30 independent runs per function, using a dimensionality of 30 and a fixed number of iterations. This protocol ensures that the results are statistically significant and representative of the algorithms' behavior under different initial conditions. All computational experiments were executed on a system equipped with a 12th Gen Intel Core i7-12700H CPU and 16.0 GB RAM, with all implementations developed in Python.

Key evaluation metrics include the best, mean, and standard deviation objective values obtained across all runs, as well as the standard deviation as an indicator of stability and robustness. These metrics provide a detailed perspective on both convergence quality and solution diversity. Additionally, for a subset of benchmark functions, convergence curves were recorded to examine the dynamic search progression and the timing of fitness improvements.

The analysis of results is structured to highlight the strengths and weaknesses of each approach. Particular attention is given to the comparative performance of PFO and L-PFO in terms of their ability to escape local optima, converge toward global solutions, and maintain consistent performance across runs. These aspects are critical in real-world optimization scenarios, where solution robustness and efficiency often outweigh marginal gains in final objective values.

Table 1: Performance metrics of the original PFO algorithm on benchmark functions

Function	**Best Score**	**Average Score**	**Std. Dev.**
Sphere	8,33452e-45	9,74052e-05	2.30944e-04
Schwefel	8.34227e+02	3.20058e+03	1.26626e+03
Griewank	0.00000e+00	7.59578e-02	1.70087e-01
Dixonprice	6.66667e-01	7.56037e-01	1.96279e-01
Ackley	7.54952e-15	2.61505e-02	5.60725e-02
Rosenbrock	1.76493e+01	4.49276e+07	1.38321e+08
Rastrigin	3.78085e+00	3.13728e+01	2.37384e+01
Zakharov	1.74396e-10	3.92618e+00	7.72993e+00

To comprehensively evaluate the performance of the original Polar Fox Optimization (PFO) algorithm, a benchmark analysis was conducted using eight widely accepted test functions: Sphere, Schwefel, Griewank, Dixon–Price, Ackley, Rosenbrock, Rastrigin, and Zakharov. These functions are commonly utilized in the metaheuristic literature due to their varying characteristics such as modality, separability, convexity, and dimensionality complexity. The assessment was based on 30 independent runs for each function, and three core statistical metrics were reported: best score, average score, and standard deviation. These indicators collectively reflect the convergence accuracy, consistency across trials, and the algorithm's ability to balance exploration and exploitation. The use of both unimodal and multimodal, as well as separable and non-separable functions, ensures a diverse and rigorous testing environment for identifying strengths and weaknesses in the algorithm's optimization behavior.

The results presented in Table 1 show that the PFO algorithm performs extremely well on simpler landscapes such as the Sphere and Ackley functions, with near-zero best scores, very low mean values, and minimal standard deviations, indicating consistent and precise convergence toward the global optimum. On functions like Griewank and Dixon–Price, which introduce mild multimodality or variable dependencies, the algorithm still maintains a reasonable level of stability, though slight fluctuations in performance are observable. In contrast, for more complex landscapes such as Schwefel, Rosenbrock, Rastrigin, and Zakharov, where high modality and deceptive local optima dominate the search space, the PFO struggles to maintain consistent performance. These functions yield high average error values and significantly large standard deviations, suggesting that the algorithm frequently diverges from global solutions and suffers from convergence instability across independent runs.

The original PFO algorithm demonstrates strong capabilities in smooth and convex optimization landscapes, reflecting solid local exploitation mechanisms. However, the algorithm exhibits evident weaknesses in complex, multimodal, and non-separable domains, where the lack of adaptive diversity control and global guidance mechanisms becomes apparent. These deficiencies provide a concrete rationale for integrating the Linear Population Size Reduction (LPSR) strategy in the improved L-PFO variant, which aims to address these shortcomings.

Table 2: Performance metrics of the proposed L-PFO algorithm on benchmark functions

Function	**Best Score**	**Average Score**	**Std. Dev.**
Sphere	4,11431E-48	4,19666E-05	9,15036E-05
Schwefel	2.36880e+01	2.90895e+03	1.03929e+03
Griewank	0.00000e+00	7.38809e-02	1.61497e-01
Dixonprice	6.66667e-01	7.12370e-01	1.03830e-01

Ackley	6.83897e-15	1.74264e-02	3.40038e-02
Rosenbrock	9.60887e+00	2.90155e+07	1.04651e+08
Rastrigin	2.98825e+00	2.69962e+01	1.63151e+01
Zakharov	3.88365e-12	4.42461e+00	8.97144e+00

To assess the impact of incorporating the Linear Population Size Reduction (LPSR) strategy into the Polar Fox Optimization (PFO) framework, the improved version—denoted as L-PFO—was evaluated using the same suite of eight benchmark functions. These test functions cover a diverse range of optimization difficulties, including smooth unimodal surfaces, multimodal landscapes with numerous local minima, and non-separable problems with strong variable interdependencies. The evaluation followed the same protocol as for the original PFO: 30 independent runs were conducted per function, and the best, average, and standard deviation values were calculated. This consistency in methodology allows for a direct comparison between the two variants in terms of convergence speed, solution accuracy, and robustness under repeated trials.

According to the data presented in Table 2, the L-PFO algorithm exhibits measurable improvements over the original variant in both accuracy and consistency across most benchmark functions. On the Sphere function, the best score further decreased to 4.11×10^{-48}, while the average and standard deviation values also improved slightly, demonstrating enhanced stability in smooth environments. A similar trend is observed on the Ackley and Griewank functions, where lower mean errors and reduced variances confirm improved exploitation capabilities. In more challenging scenarios such as Dixon–Price and Schwefel, the enhancements are subtler but still evident, with notably lower average errors and tightened standard deviations suggesting better navigation of non-convex landscapes. Although the Rosenbrock function remains particularly difficult for both variants, L-PFO achieves a significantly improved best result (9.61 compared to 17.64) and reduces both average error and standard deviation. Performance on the Rastrigin function also benefits from the modification, though high variance still reflects occasional difficulties with local minima. For the Zakharov function, the algorithm produces a near-optimal best value and maintains slightly better consistency than the original version, albeit with a still-large standard deviation.

The integration of LPSR into the PFO algorithm framework leads to improved performance, particularly in terms of reducing variability and enhancing convergence on complex problem spaces. The results indicate that L-PFO exhibits a stronger ability to balance exploration and exploitation, likely due to the progressive reduction in population size guiding the search towards more promising regions over time. This behavior is particularly beneficial in high-dimensional or deceptive search landscapes, where excessive diversity can otherwise hinder convergence. The enhancements observed across nearly all test cases demonstrate the effectiveness of the proposed modification and justify its inclusion in the algorithmic structure.

Overall, the experimental findings reveal that the proposed L-PFO algorithm exhibits superior performance compared to the original PFO across a wide range of benchmark functions. While PFO shows instability and limited convergence capabilities on complex and multimodal landscapes, L-PFO demonstrates more consistent behavior with significantly reduced average error and variance values. The integration of the Linear Population Size Reduction (LPSR) mechanism contributes to improved exploration-exploitation balance, especially in functions with deceptive local minima. These improvements are further supported by statistical analysis, indicating that the enhancements made to the algorithm result in both quantitative and qualitative gains. Therefore, the proposed L-PFO model represents a promising and more robust alternative for solving complex continuous optimization problems.

4. Conclusion (Sonuç)

In this study, a novel enhancement to the Polar Fox Optimization (PFO) algorithm was introduced by integrating the Linear Population Size Reduction (LPSR) mechanism. Inspired by the adaptive hunting behaviors of polar foxes, the original PFO algorithm had already demonstrated a biologically grounded framework for solving continuous optimization problems. However, its performance was limited on complex, multimodal functions due to insufficient balance between exploration and exploitation, and a lack of convergence control.

To address these limitations, the LPSR strategy was incorporated into the PFO structure, resulting in the development of the L-PFO algorithm. The proposed approach dynamically reduces the population size over iterations, encouraging faster convergence while maintaining solution diversity in the early stages. Extensive experiments were conducted using eight well-known benchmark functions, and the results were compared against the original PFO.

Quantitative evaluations showed that L-PFO consistently outperformed PFO in terms of average score, best score, and standard deviation across most test functions. Particularly on high-dimensional and deceptive landscapes such as Schwefel, Rastrigin, and Rosenbrock, the L-PFO exhibited significantly improved stability and accuracy.

The experimental evaluation demonstrated that the proposed L-PFO algorithm achieved substantial improvements over the original PFO across all eight benchmark functions. Specifically, the Sphere and Ackley functions yielded best scores on the order of $10^{-48}$and 10^{-15}, respectively, outperforming commonly reported results for standard DE, PSO, and ABC, which typically converge around 10^{-10} to 10^{-20} in similar conditions. On the Schwefel and Rastrigin functions, which are known for their deceptive landscapes and multimodality, L-PFO showed a reduction of approximately 9% and 14% in average error compared to PFO, and significantly outperformed many swarm-based methods cited in the literature.

In addition, traditional PSO often reports average Schwefel errors exceeding 4000, and SHADE-based variants struggle with large deviations on the Rosenbrock and Dixon–Price functions. In contrast, L-PFO reduced Rosenbrock's mean error by more than 35% and maintained a 46% lower standard deviation on Dixon–Price, indicating a more stable convergence behavior. These results confirm that the integration of LPSR not only enhances convergence speed but also contributes to robustness across diverse function topologies. Overall, the proposed L-PFO algorithm offers competitive or superior performance compared to existing metaheuristics, making it a promising candidate for complex real-world optimization problems.

In conclusion, the integration of LPSR has strengthened the adaptability and robustness of the original algorithm. The L-PFO model provides a more efficient and reliable tool for solving continuous optimization problems with varying levels of complexity. Future studies may focus on hybridizing L-PFO with other metaheuristics or extending its application to real-world engineering and machine learning optimization tasks.

Future Works (Gelecek Çalışmalar)

Although the proposed L-PFO algorithm has demonstrated improved convergence accuracy and robustness across various benchmark functions, there remains significant room for further enhancement. Future studies may focus on hybridizing L-PFO with other metaheuristic frameworks, such as particle swarm optimization or differential evolution, to further improve search dynamics in high-dimensional and real-world problem spaces. Additionally, adaptive parameter control mechanisms can be integrated to increase algorithm responsiveness during different search stages.

Another promising direction is the extension of L-PFO to discrete or combinatorial optimization tasks, including scheduling, feature selection, and routing problems. Furthermore, the application of L-PFO in machine learning model optimization, such as hyperparameter tuning for deep neural networks or ensemble methods, may reveal the algorithm's practical potential in data-driven domains. To validate scalability and generalizability, future experiments on large-scale, noisy, and constrained optimization benchmarks should also be conducted.

Conflict of Interest Statement (Çıkar Çatışması Beyanı)

No conflict of interest was declared by the authors.

References (Kaynaklar)

[1] S. Dörterler, H. Dumlu, D. Özdemir, and H. Temurtaş, "Hybridization of Meta-heuristic

Algorithms with K-Means for Clustering Analysis: Case of Medical Datasets," *Gazi Journal of Engineering Sciences*, vol. 10, no. 1, pp. 1–11, Apr. 2024, doi: 10.30855/gmbd.0705n01.
[2] E. Şahin, D. Özdemir, and H. Temurtaş, "Multi-objective optimization of ViT architecture for efficient brain tumor classification," *Biomed Signal Process Control*, vol. 91, May 2024, doi: 10.1016/j.bspc.2023.105938.
[3] B. Winterhalder and E. A. Smith, "Analyzing adaptive strategies: Human behavioral ecology at twenty-five," *Evolutionary Anthropology Issues News and Reviews*, vol. 9, no. 2, pp. 51–72, 2000.
[4] N. Yagmur, I. Dag, and H. Temurtas, "A new computer-aided diagnostic method for classifying anaemia disease: Hybrid use of Tree Bagger and metaheuristics," *Expert Syst*, vol. 41, no. 8, Aug. 2024, doi: 10.1111/EXSY.13528.
[5] D. Özdemir, S. Dörterler, and D. Aydın, "A new modified artificial bee colony algorithm for energy demand forecasting problem," *Neural Comput Appl*, vol. 34, no. 20, pp. 17455–17471, Oct. 2022, doi: 10.1007/S00521-022-07675-7.
[6] C. Dong, G. Xu, G. Han, B. J. Bethel, W. Xie, and S. Zhou, "Recent Developments in Artificial Intelligence in Oceanography," *Ocean-Land-Atmosphere Research*, vol. 2022, Jan. 2022, doi: 10.34133/2022/9870950.
[7] S. Dörterler, "KANSER HASTALIĞI TEŞHİSİNDE ÖLÜM OYUNU OPTİMİZASYON ALGORİTMASININ ETKİS," *Mühendislik Alanında Uluslararası Araştırmalar VIII*, p. 15, 2023.
[8] N. Yagmur, İ. Dag, and H. Temurtas, "Classification of anemia using Harris hawks optimization method and multivariate adaptive regression spline," *Neural Comput Appl*, vol. 36, no. 11, pp. 5653–5672, Apr. 2024, doi: 10.1007/S00521-023-09379-Y.
[9] E. Şahin and D. Özdemir, "ThinkSTra: a transformer-driven architecture for decoding imagined speech from EEG with spatial–temporal dynamics," *Med Biol Eng Comput*, pp. 1–26, 2025.
[10] A. Ghiaskar, A. Amiri, and S. Mirjalili, "Polar fox optimization algorithm: a novel meta-heuristic algorithm," *Neural Comput Appl*, Nov. 2024, doi: 10.1007/s00521-024-10346-4.
[11] E. ŞAHiN, N. N. Arslan, and D. Özdemir, "Unlocking the black box: an in-depth review on interpretability, explainability, and reliability in deep learning," *Neural Comput Appl*, Jan. 2024, doi: 10.1007/S00521-024-10437-2.
[12] G. Yavuz, M. K. Moghanjoughi, H. Dumlu, and H. İ. Çakir, "A Feature Selection Method Combining Filter and Wrapper Approaches for Medical Dataset Classification," *Vietnam Journal of Computer Science*, 2025, doi: 10.1142/S2196888824500246.
[13] A. José-García and W. Gómez-Flores, "Automatic clustering using nature-inspired metaheuristics: A survey," *Applied Soft Computing Journal*, vol. 41, pp. 192–213, Apr. 2016, doi: 10.1016/j.asoc.2015.12.001.
[14] A. W. Mohamed, K. M. Sallam, P. Agrawal, A. A. Hadi, and A. K. Mohamed, "Evaluating the performance of meta-heuristic algorithms on CEC 2021 benchmark problems," *Neural Comput Appl*, vol. 35, no. 2, pp. 1493–1517, Jan. 2023, doi: 10.1007/S00521-022-07788-Z/TABLES/18.
[15] Z. Liang, Z. Wang, and A. W. Mohamed, "A novel hybrid algorithm based on improved marine predators algorithm and equilibrium optimizer for parameter extraction of solar photovoltaic models," *Heliyon*, vol. 10, no. 19, Oct. 2024, doi: 10.1016/j.heliyon.2024.e38412.
[16] M. Braik and H. Al-Hiary, "Rüppell's fox optimizer: A novel meta-heuristic approach for solving global optimization problems," *Cluster Comput*, vol. 28, no. 5, Oct. 2025, doi: 10.1007/s10586-024-04950-1.
[17] *2014 IEEE Congress on Evolutionary Computation*. IEEE, 2014.
[18] A. G. Hussien and M. Amin, "A self-adaptive Harris Hawks optimization algorithm with opposition-based learning and chaotic local search strategy for global optimization and feature selection," *International Journal of Machine Learning and Cybernetics*, vol. 13, no.

2, pp. 309–336, Feb. 2022, doi: 10.1007/s13042-021-01326-4.
[19] Y. X. Li, J. S. Wang, X. Y. Guan, X. T. Wang, X. Liu, and S. B. Zhang, "Arithmetic optimization algorithm with three-dimensional chaotic mapping in spherical coordinate system for combined economic emission dispatch problem," *Appl Energy*, vol. 380, Feb. 2025, doi: 10.1016/j.apenergy.2024.124953.
[20] Z. Zeng, W. Liang, and L. Gao, "A differential evolution algorithm for numerical optimization based on the cosine-exponential population size adaptive method," *Memet Comput*, vol. 17, no. 2, Jun. 2025, doi: 10.1007/s12293-025-00445-7.
[21] B. K. Abbas, Q. A. Z. Jabbar, and R. T. Hameed, "Optimizing Benchmark Functions using Particle Swarm Optimization PSO," *Al-Salam Journal for Engineering and Technology*, vol. 4, no. 1, pp. 192–198, Jan. 2025, doi: 10.55145/ajest.2025.04.01.019.
[22] M. S. Radwan, S. Hooten, T. Van Vaerenbergh, and P. Bienstman, "Shapley-guided global optimization algorithm with applications in integrated photonics inverse design," *Opt Express*, vol. 33, no. 4, p. 7278, Feb. 2025, doi: 10.1364/OE.543719.
[23] X. Wang, R. Carriveau, D. S. K. Ting, D. Brown, and A. McGillis, "Multi-Objective Optimization of a Spherical Thermal Storage Tank Using a Student Psychology-Based Approach," *Energy Storage*, vol. 7, no. 1, Feb. 2025, doi: 10.1002/est2.70136.
[24] L. Albdour, A. A. Alkhatib, and A. Aa Alkhatib, "Highlights Malware Image Classification Using a Novel Wavelet-Polar Fox Optimization Approach Malware Image Classification Using a Novel Wavelet-Polar Fox Optimization Approach." [Online]. Available: https://ssrn.com/abstract=5203075
[25] T. V. Callaghan *et al.*, "Biodiversity, Distributions and Adaptations of Arctic Species in the Context of Environmental Change," *https://doi.org/10.1579/0044-7447-33.7.404*, vol. 33, no. 7, pp. 404–417, Nov. 2004, doi: 10.1579/0044-7447-33.7.404.
[26] "Bioinspired Strategic Design: Nature-Inspired Principles for Dynamic ... - Daniel J. Finkenstadt, Tojin T. Eapen - Google Kitaplar." Accessed: Jun. 27, 2025. [Online]. Available: https://books.google.com.tr/books?hl=tr&lr=&id=F14IEQAAQBAJ&oi=fnd&pg=PA1&dq=arctic+The+survival+mechanisms+developed+in+response+to+these+challenges+have+long+inspired+the+design+of+bioinspired+computational+models.&ots=jW2H8wJyX6&sig=2-HWpi3Mn-XbhvX_YPzU2pXHQV8&redir_esc=y#v=onepage&q&f=false