Metaheuristic Algorithms with Applications

www.maa.reapress.com

Metaheur. Algor. Appl. Vol. 3, No. 1 (2026) 33–60.

Paper Type: Original Article

Forest Growth Optimization: An Ecological-Succession-Inspired Metaheuristic with Pioneer-Intermediate-Climax Stage Progression

Pritiprava Mishra[1,*], Haruna Mavakumba Kefas[2]

[1] Department of Computer Science and Engineering, GIFT Autonomous, Bhubaneswar, Odisha, India; pritiprava.mishra@gift.edu.in.

[2] Department of Chemical Engineering, Modibbo Adama University, Adamawa, Nigeria; hmkefas@yahoo.com.

Citation:

Received: 28 August 2025 *Revised: 10 October 2025* *Accepted: 27 December 2025*	Mishra, P., & Kefas, H. M. (2026). Forest growth optimization: An ecological-succession-inspired metaheuristic with pioneer-intermediate-climax stage progression. *Metaheuristic Algorithms with Applications, 3*(1), 33-60.

Abstract

This paper proposes Forest Growth Optimization (FGO), a novel nature-inspired metaheuristic algorithm for solving continuous global optimization problems. The algorithm is inspired by Forest ecological succession: Pioneer species colonization, canopy closure, climax community emergence, and it incorporates Multi-stage succession (pioneer -> intermediate -> climax), gap dynamics, and seed dispersal as its core search mechanism. Unlike existing metaheuristics that apply a single update rule uniformly across the population, FGO introduces a set of named operators that collectively capture the multi-phase dynamics of its biological inspiration. Each operator is formalized as a mathematically defined update rule, and adaptive parameters control interactions between operators in response to the local geometry of the search landscape. The proposed algorithm is evaluated on the CEC 2017 benchmark suite (29 test functions) and the CEC 2020 benchmark suite (10 test functions), and its performance is compared against 11 state-of-the-art metaheuristic algorithms: PSO, GA, DE, GWO, WOA, ABC, BBO, FA, CS, FPA, GSA. The experimental results show that FGO achieves the best mean fitness on 24 of 29 CEC 2017 functions and 8 of 10 CEC 2020 functions. The Friedman test ranks FGO first with an average rank of 1.34, and the Wilcoxon rank-sum test confirms that the improvements are statistically significant at $p < 0.05$ on at least 22 of the 29 CEC 2017 functions. An ablation study quantifies each operator's contribution to the algorithm's overall performance, with the most critical operator degrading performance by up to 27.3% when removed. The algorithm is further validated on two classical engineering design problems (welded beam design and pressure vessel design), where it obtains solutions competitive with the best-known optima. A Sobol sensitivity analysis confirms that the algorithm's parameters are well-balanced, with no single parameter dominating the algorithm's behavior. A scalability analysis on problem dimensions D = 10, 50, 100, 500, and 1000 demonstrates near-linear scaling of wall-clock time with the problem dimension. Together, the results show that FGO is a competitive metaheuristic for continuous global optimization, with broad applicability to engineering design, Machine Learning (ML) hyperparameter tuning, and scientific computing.

Keywords: Metaheuristic optimization, Forest growth optimization, Forest ecological succession: Pioneer species colonization, canopy closure, climax community emergence, Global optimization, CEC 2017 benchmark, Engineering design.

1 | Introduction

Global optimization—the search for the global optimum of an objective function over a continuous search space—is a foundational problem in engineering design, Machine Learning (ML), scientific computing, and operations research. Despite decades of research, the problem remains challenging when the objective function is non-convex, high-dimensional, expensive to evaluate, or noisy. This paper proposes Forest

Corresponding Author: pritiprava.mishra@gift.edu.in

https://doi.org/10.48313/maa.vi.79

Growth Optimization (FGO), a novel metaheuristic algorithm inspired by Forest ecological succession: Pioneer species colonization, canopy closure, and climax community emergence.

Metaheuristic algorithms have become the method of choice for solving difficult global optimization problems because of their gradient-free nature, flexibility in handling non-convex and discontinuous objectives, and ability to escape local optima through stochastic search. The past three decades have witnessed the development of hundreds of metaheuristic algorithms inspired by diverse sources including animal behavior (PSO [1], ACO [2], GWO [3], WOA [4]), plant biology (FPA [5], PGSA), physical phenomena (SA [6], GSA [7], SCA [8]), and human culture (ICA, RSLO). Despite this proliferation, the No Free Lunch theorem of Wolpert and Macready [9] shows that no single metaheuristic can dominate all others on all possible problems, which continues to motivate the development of new algorithms well suited to specific problem classes. This paper contributes to this effort by introducing FGO, motivated by the observation that existing metaheuristics do not adequately capture the multi-phase dynamics of Forest ecological succession: pioneer species colonization, canopy closure, and climax community emergence.

FGO's key innovation is its use of named operators, each corresponding to a distinct phase of the underlying biological or physical inspiration. Rather than applying a single update rule uniformly across the population (as in PSO's velocity update or GWO's encirclement rule), FGO partitions the search process into distinct phases, each governed by a mathematically formalized, adaptively triggered operator. This design choice has three consequences: 1) it allows the algorithm to allocate search effort adaptively, spending more time in the phases that are most productive for the current landscape, 2) it provides a natural mechanism for the exploration-exploitation balance, with different phases specialized for different aspects of the search, and 3) it makes the algorithm's behavior more interpretable, since each phase's contribution can be analyzed and ablated separately.

The literature review in Section 2 traces the historical development of metaheuristic algorithms in the relevant inspiration domain and identifies three open challenges that the present algorithm addresses. Section 3 (literature review) is organized into five subsections, each covering a distinct historical period or methodological tradition, and culminating in a synthesis that motivates the design of FGO. Section 4 presents the proposed algorithm in detail: Section 4.1 provides the inspiration narrative; Section 4.2 describes the named operators; Section 4.3 provides the mathematical formulation; Section 4.4 presents the pseudocode; Section 4.5 presents the algorithm's flowchart; and Section 4.6 analyzes the algorithm's computational complexity. This paper also contributes a detailed design of FGO.

Section 5 presents an extensive experimental evaluation. The algorithm is evaluated on the CEC 2017 benchmark suite of 29 test functions [10] and the CEC 2020 benchmark suite of 10 test functions [11], and its performance is compared against 11 state-of-the-art Metaheuristic algorithms: PSO, GA, DE, GWO, WOA, ABC, BBO, FA, CS, FPA, GSA. The competitors were chosen to span the major methodological traditions in metaheuristic research, including Evolutionary Algorithms (EAs) (GA, DE), Swarm Intelligence (SI) (PSO, ABC), physics-based algorithms (SA, GSA, SCA), and recent nature-inspired algorithms (GWO, WOA, SSA, HHO). All algorithms are evaluated under identical conditions: Population size N=30 and maximum number of function evaluations MaxFEs=10,000*D, with 30 independent runs per benchmark function. We assess the statistical significance of the results using the Friedman test [12] and the Wilcoxon rank-sum test [13].

In addition to the benchmark evaluation, Section 5 reports an ablation study that quantifies the contribution of each named operator to the algorithm's overall performance; a Sobol sensitivity analysis[13] that quantifies the algorithm's sensitivity to its hyperparameters; and a scalability analysis that evaluates the algorithm's wall-clock time on problem dimensions D = 10, 50, 100, 500, and 1000. Finally, we validate the algorithm on two classical real-world engineering design problems: the welded beam design problem [14] and the pressure vessel design problem [15], both constrained optimization problems with mixed continuous-integer decision variables. The algorithm's performance on these problems demonstrates its applicability to practical engineering design.

The main contributions of this paper are summarized as follows: 1) we propose FGO, a novel metaheuristic algorithm inspired by Forest ecological succession: pioneer species colonization, canopy closure, climax community emergence, with a set of named operators that collectively capture the multi-phase dynamics of its inspiration, 2) we provide a rigorous mathematical formulation of each operator, including adaptive parameters that respond to the local landscape geometry, 3) we conduct an extensive experimental evaluation on the CEC 2017 and CEC 2020 benchmarks, comparing against 11 state-of-the-art competitors, and we report statistical significance via the Friedman and Wilcoxon tests, 4) we perform an ablation study that quantifies the contribution of each operator, a Sobol sensitivity analysis, and a scalability analysis, 5) we validate the algorithm on two classical engineering design problems. The remainder of the paper is organized as follows: Section 2 reviews the relevant literature; Section 3 presents the proposed algorithm; Section 4 reports the experimental evaluation; and Section 5 concludes the paper and identifies directions for future research.

2 | From Biogeography to Forest Succession: A Historical Account of Ecological-Inspired Metaheuristics

This section reviews the relevant literature on nature-inspired metaheuristic algorithms, organized into five subsections that trace the field's historical development. The review culminates in a synthesis that identifies three open challenges that the proposed algorithm addresses. The review is deliberately comprehensive, covering both the foundational contributions that established the methodological framework and the more recent advances that have refined it. The goal is to position the present contribution within the broader research tradition and to motivate the specific design choices that distinguish the proposed algorithm.

2.1 | Biogeography-Based Optimization (2008-2014)

Ecological-inspired metaheuristics began with Simon's [16] Biogeography-Based Optimization (BBO), which modeled species migration between habitats as an optimization procedure. BBO explicitly used Habitat Suitability Indices (HSI)—analogous to fitness—and migration rates (immigration and emigration) that depended on HSI. High-HSI habitats had high emigration rates (they exported species to other habitats) and low immigration rates (they were saturated). In contrast, low-HSI habitats had high immigration rates and low emigration rates. This asymmetry produced a natural elitism mechanism in which high-quality solutions spread their information to low-quality solutions. BBO's success on the CEC 2008 benchmark set a methodological precedent: ecological dynamics could serve as a fertile source of optimization operators. In the following years, researchers introduced Biogeography-Based Learning (BBL) and the Hybrid BBO with DE [17], both of which extended the basic BBO framework.

2.2 | Predator-Prey and Food-Web Algorithms (2010-2017)

The 2010s saw a parallel development of predator-prey and food-web-inspired metaheuristics. Predator-Prey Optimization (PPO) [18] modeled predator-prey dynamics as an optimization procedure, with predators driving prey toward better regions of the search space. The Invasive Weed Optimization (IWO) [19] modeled the dispersal and colonization behavior of invasive weeds, with seeds dispersed via normal distributions whose variance decreased over generations. The Forest Optimization Algorithm (FOA) [20] modeled forest dynamics including tree seeding, growth, and death, with local-seeding and global-seeding operators providing exploration and exploitation. These algorithms collectively showed that ecological community dynamics—predation, competition, dispersal, and colonization—could inspire effective optimization operators. Still, they did not model the multi-stage progression of ecological succession that characterizes the long-term development of forest ecosystems.

2.3 | Firefly, Flower, and Plant-Pollination Algorithms (2012-2018)

The 2010s also saw the introduction of several plant-pollination-inspired algorithms that touched on ecological dynamics. The Flower Pollination Algorithm (FPA) [21] modeled biotic and abiotic pollination as dual global and local search operators. The Photosynthetic Algorithm (PA) [21] and the Plant Growth Simulation Algorithm (PGSA) [22] explored alternative plant-inspired operators. The Strawberry Algorithm (SA) [23] modeled the runner-based reproduction of strawberries as an optimization procedure. Together, these algorithms showed that the dynamics of plant reproduction—pollination, photosynthesis, runner-based propagation—could inspire effective optimization operators. Still, they did not model the multi-decade progression of forest succession that characterizes the long-term development of forest ecosystems.

2.4 | The Forest Succession Formulation

The FGO algorithm introduced in this paper formalizes the multi-stage progression of forest ecological succession within an optimization framework. The pioneer seeding phase models the initial colonization of disturbed land by fast-growing, widely dispersed pioneer species such as birch (Betula) and pine (Pinus). The germination phase models seed sprouting based on local soil quality (analogous to fitness). The sapling growth phase models directional sapling growth toward light, providing a gradient-following local search operator. The canopy competition phase models the competitive suppression of understory trees by canopy dominants, providing an exploitation pressure. The gap dynamics phase models the colonization of canopy gaps created by treefall, providing a stochastic diversification operator. The intermediate-species phase models the transition to slower-growing, shade-tolerant species such as oak (Quercus) and beech (Fagus), with a reduced growth rate that provides more selective search. The climax community phase models the formation of a stable, self-replacing community, providing a strong exploitation phase. The disturbance reset phase models the stochastic occurrence of disturbances (fire, wind, pest outbreak) that reset part of the forest to the pioneer phase, providing a principled restart mechanism.

2.5 | Synthesis and Open Challenges for Ecological Metaheuristics

The historical survey identifies three open challenges for ecological-inspired metaheuristics that FGO addresses. First, the population structure has typically been homogeneous, with all candidates executing the same update rule, which does not capture the multi-species structure of real forests. FGO introduces the species-tag mechanism, in which each candidate carries a species tag (pioneer, intermediate, climax) that determines its update rule. Second, search dynamics have typically been static, applying the same operators throughout the search, which does not capture the temporal progression of ecological succession. FGO introduces the stage-transition mechanism, in which the population transitions from pioneer to intermediate to climax species at fixed intervals, providing a principled exploration-exploitation schedule. Third, diversification mechanisms have typically been limited to random restarts, which discard accumulated information. FGO introduces the gap-dynamics operator as a local diversification mechanism that creates new pioneer colonization sites within the search space, and the disturbance-reset operator as a partial restart that preserves the best solutions while reinitializing the worst. Section 5 demonstrates that these innovations yield substantial improvements over the strongest ecological-inspired baselines.

3 | Proposed Algorithm

This section presents the proposed FGO algorithm. The section is organized into six subsections: Section 4.1 presents the inspiration narrative; Section 4.2 describes the named operators; Section 4.3 provides the mathematical formulation; Section 4.4 presents the pseudocode; Section 4.5 presents the algorithm's flowchart; and Section 4.6 analyzes the algorithm's computational complexity.

4 | Inspiration

This section presents the inspiration narrative for FGO. The narrative is organized as a sequence of paragraphs, each describing a distinct feature of the underlying biological, physical, or cultural phenomenon that inspires the algorithm. Section 4.2 then maps each feature to a corresponding algorithmic operator, providing a clear, motivated design rationale for each component. The narrative draws on primary research in the relevant domain and aims to provide both the motivation for the algorithm's design and the interpretive context for understanding its behavior.

Forest ecological succession is one of the most studied phenomena in ecology, with a literature extending back to the late 19th century. Cowles [24] made a foundational contribution by describing plant community succession on the dunes of Lake Michigan. The present section describes the dynamics of forest succession in detail to motivate the design of FGO.

The first remarkable feature of forest succession is the pioneer-species colonization. When a disturbance (such as a fire, windstorm, or clearcut) removes existing vegetation from a site, pioneer species rapidly colonize the bare ground—fast-growing, widely dispersed, short-lived species adapted to the harsh conditions of disturbed sites. Common pioneer tree species include birch (Betula), pine (Pinus), aspen (Populus), and larch (Larix). These species produce large numbers of small, lightweight seeds that are dispersed by wind over long distances, allowing them to colonize disturbed sites rapidly. Pioneer species are typically shade-intolerant, meaning they require full sunlight for optimal growth, while shade-tolerant species outcompete them once the canopy closes. In FGO, the pioneer-seeding operator disperses the initial population widely across the search space, mimicking the wind dispersal of pioneer seeds.

The second remarkable feature is germination. Once the seeds land on the disturbed site, they must germinate to establish themselves. Germination depends on several factors, including soil moisture, soil temperature, and the presence of suitable mycorrhizal fungi. Germination rates vary widely among species, with pioneer species typically having higher germination rates than climax species. The germination phase is critical because it determines which seeds will establish as saplings and which will die. In FGO, the germination operator evaluates each seed's germination probability based on local soil quality (analogous to fitness), with seeds in high-quality regions having a higher probability of germinating.

The third remarkable feature is the sapling-growth phase. Once a seed germinates, it develops into a sapling — a young tree that is small enough to be vulnerable to browsing, fire, and competition from other vegetation. The sapling grows toward the light, with the growth rate depending on the available light, water, and nutrients. The sapling-growth phase is critical because it determines which saplings will survive to reproductive maturity. In FGO, the sapling-growth operator updates each sapling's position via a gradient-ascent operator, mimicking trees' directional growth toward light.

The fourth remarkable feature is the canopy-competition phase. As the saplings grow, their crowns eventually meet and form a closed canopy, blocking the sunlight from reaching the understory. At this point, canopy competition begins: taller trees shade out smaller trees, suppressing their growth and eventually killing them. Canopy competition is one of the most important mechanisms of natural selection in forests and the primary driver of self-thinning, which reduces tree density as the forest matures. In FGO, the canopy-competition operator identifies suppressed trees (those outperformed by their neighbors) and moves them toward the winning competitors, mimicking the competitive pressure of canopy closure.

The fifth remarkable feature is gap dynamics. Even in a mature forest, the canopy is not static: individual trees die from old age, disease, windthrow, or pest outbreak, creating canopy gaps that allow sunlight to reach the forest floor. Pioneer species rapidly colonize these gaps, restarting the successional cycle. Gap dynamics is one of the most important mechanisms of diversity maintenance in forests, because it prevents any single species from dominating the canopy indefinitely. In FGO, the gap-dynamics operator creates new pioneer colonization sites (random restarts) within the search space, mimicking canopy gap formation.

The sixth remarkable feature is the intermediate-species phase. As the forest matures, pioneer species are gradually replaced by intermediate species—slower-growing, longer-lived, and more shade-tolerant species such as oak (Quercus), hickory (Carya), and ash (Fraxinus). Intermediate species can establish in the pioneer forest understory and gradually grow into the canopy as pioneer species die. The transition from pioneer to intermediate species typically takes 50 to 100 years in temperate forests. In FGO, the intermediate-species operator transitions the population from pioneer to intermediate species at a fixed interval, with a reduced growth rate that provides more selective search.

The seventh remarkable feature is the climax-community phase. As the forest continues to mature, intermediate species are gradually replaced by climax species—the most shade-tolerant, longest-lived, and largest species adapted to the site's climate and soil. Common climax species in temperate forests include beech (Fagus), maple (Acer), and hemlock (Tsuga). The climax community is a stable, self-replacing community that can persist for centuries or even millennia without major disturbances. In FGO, the climax-community operator collapses the population to a tight Gaussian cloud around the best-so-far solution, providing a strong exploitation phase.

The eighth and final remarkable feature is the disturbance-reset behavior. Even climax communities are not permanent; major events such as catastrophic fires, hurricanes, pest outbreaks, or human activity periodically disturb them. These disturbances reset the successional cycle, returning the site to the pioneer-species phase. This disturbance-reset behavior is one of the most important mechanisms of diversity maintenance at the landscape scale because it ensures the forest is a mosaic of stands at different successional stages. In FGO, the disturbance-reset operator resets a fraction of the population to fresh pioneer seeds, mimicking the stochastic occurrence of disturbances.

These eight behaviors—pioneer seeding, germination, sapling growth, canopy competition, gap dynamics, intermediate species, climax community, and disturbance reset—collectively drive forest ecological succession and inspire FGO's seven operators. The next subsection formalizes these operators as mathematical update rules, and Section 4.3 derives the corresponding equations.

4.1 | Algorithm Description

This section describes the named operators of FGO in detail. Each operator corresponds to a distinct feature of the algorithm's biological or physical inspiration, as described in Section 4.1. The operators are presented in the order in which they are applied within a single iteration of the algorithm, although some operators are triggered conditionally based on the current state of the search. The mathematical formulation of each operator is deferred to Section 4.3, and the pseudocode is presented in Section 4.4.

4.1.1 | Pioneer seeding

The Pioneer Seeding operator is a core component of FGO. As described in Section 4.1, this operator is inspired by the corresponding behavior in the algorithm's biological or physical inspiration. The population is initialized as pioneer species seeds dispersed widely. The operator captures the essential dynamics of this behavior while remaining computationally tractable and mathematically well-defined.

The Pioneer Seeding operator is applied at a specific point in the algorithm's iteration loop. Its effect on the candidate-solution population depends on the current state of the search, including the current best solution, the distribution of fitness values across the population, and the iteration index. The operator's parameters are adaptively tuned during the search to balance exploration and exploitation. Section 4.3 gives the operator's specific mathematical form, and Section 4.5 shows its position in the overall algorithm flow in the flowchart.

The rationale for including the Pioneer Seeding operator in FGO is twofold. First, the operator provides a specific search behavior not adequately captured by the other operators, ensuring the algorithm can handle a wide range of landscape geometries. Second, the operator is grounded in well-documented behavior from the algorithm's biological or physical inspiration, providing a principled design rationale rather than an arbitrary mathematical choice. The ablation study in Section 5.8 quantifies this operator's contribution to the algorithm's overall performance.

4.1.2 | Germination phase

The Germination Phase operator is a core component of FGO. As described in Section 4.1, this operator is inspired by the corresponding behavior in the algorithm's biological or physical inspiration. Seeds germinate based on local soil quality (fitness). The operator captures the essential dynamics of this behavior while remaining computationally tractable and mathematically well-defined.

The Germination Phase operator is applied at a specific point in the algorithm's iteration loop. Its effect on the candidate-solution population depends on the current search state, including the current best solution, the distribution of fitness values across the population, and the iteration index. The operator's parameters are adaptively tuned during the search to balance exploration and exploitation. Section 4.3 gives the operator's specific mathematical form, and Section 4.5 shows its position in the overall algorithm flow in the flowchart.

The rationale for including the Germination Phase operator in FGO is twofold. First, the operator provides a specific search behavior not adequately captured by the other operators, ensuring the algorithm can handle a wide range of landscape geometries. Second, the operator is grounded in well-documented behavior from the algorithm's biological or physical inspiration, providing a principled design rationale rather than an arbitrary mathematical choice. The ablation study in Section 5.8 quantifies this operator's contribution to the algorithm's overall performance.

4.1.3 | Sapling growth

The Sapling Growth operator is a core component of FGO. As described in Section 4.1, this operator is inspired by the corresponding behavior in the algorithm's biological or physical inspiration. Saplings grow toward light (gradient ascent). The operator captures the essential dynamics of this behavior while remaining computationally tractable and mathematically well-defined.

The Sapling Growth operator is applied at a specific point in the algorithm's iteration loop. Its effect on the candidate-solution population depends on the current state of the search, including the current best solution, the distribution of fitness values across the population, and the iteration index. The operator's parameters are adaptively tuned during the search to balance exploration and exploitation. Section 4.3 gives the operator's specific mathematical form, and Section 4.5 shows its position in the overall algorithm flow in the flowchart.

The rationale for including the Sapling Growth operator in FGO is twofold. First, the operator provides a specific search behavior not adequately captured by the other operators, ensuring the algorithm can handle a wide range of landscape geometries. Second, the operator is grounded in well-documented behavior from the algorithm's biological or physical inspiration, providing a principled design rationale rather than an arbitrary mathematical choice. The ablation study in Section 5.8 quantifies this operator's contribution to the algorithm's overall performance.

4.1.4 | Canopy competition

The Canopy Competition operator is a core component of FGO. As described in Section 4.1, this operator is inspired by the corresponding behavior in the algorithm's biological or physical inspiration. Adjacent trees compete; the loser is suppressed. The operator captures the essential dynamics of this behavior while remaining computationally tractable and mathematically well-defined.

The Canopy Competition operator is applied at a specific point in the algorithm's iteration loop. Its effect on the candidate-solution population depends on the current state of the search, including the current best solution, the distribution of fitness values across the population, and the iteration index. The operator's parameters are adaptively tuned during the search to balance exploration and exploitation. Section 4.3 gives the operator's specific mathematical form, and Section 4.5 shows its position in the overall algorithm flow in the flowchart.

The rationale for including the Canopy Competition operator in FGO is twofold. First, the operator provides a specific search behavior not adequately captured by the other operators, ensuring the algorithm can handle a wide range of landscape geometries. Second, the operator is grounded in well-documented behavior from the algorithm's biological or physical inspiration, providing a principled design rationale rather than an arbitrary mathematical choice. The ablation study in Section 5.8 quantifies this operator's contribution to the algorithm's overall performance.

4.1.5 | Gap dynamics

The Gap Dynamics operator is a core component of FGO. As described in Section 4.1, this operator is inspired by the corresponding behavior in the algorithm's biological or physical inspiration. Treefall creates a canopy gap; new pioneers colonize. The operator captures the essential dynamics of this behavior while remaining computationally tractable and mathematically well-defined.

The Gap Dynamics operator is applied at a specific point in the algorithm's iteration loop. Its effect on the population of candidate solutions depends on the current state of the search, including the current best solution, the distribution of fitness values across the population, and the iteration index. The operator's parameters are adaptively tuned during the search to balance exploration and exploitation. Section 4.3 gives the operator's specific mathematical form, and Section 4.5 shows its position in the overall algorithm flow in the flowchart.

The rationale for including the Gap Dynamics operator in FGO is twofold. First, the operator provides a specific search behavior not adequately captured by the other operators, ensuring the algorithm can handle a wide range of landscape geometries. Second, the operator is grounded in well-documented behavior from the algorithm's biological or physical inspiration, providing a principled design rationale rather than an arbitrary mathematical choice. The ablation study in Section 5.8 quantifies this operator's contribution to the algorithm's overall performance.

4.1.6 | Intermediate species

The Intermediate Species operator is a core component of FGO. As described in Section 4.1, this operator is inspired by the corresponding behavior in the algorithm's biological or physical inspiration. Stage transition introduces intermediate species. The operator captures the essential dynamics of this behavior while remaining computationally tractable and mathematically well-defined.

The Intermediate Species operator is applied at a specific point in the algorithm's iteration loop. Its effect on the candidate-solution population depends on the current state of the search, including the current best solution, the distribution of fitness values across the population, and the iteration index. The operator's parameters are adaptively tuned during the search to balance exploration and exploitation. Section 4.3 gives the operator's specific mathematical form, and Section 4.5 shows its position in the overall algorithm flow in the flowchart.

The rationale for including the Intermediate Species operator in FGO is twofold. First, the operator provides a specific search behavior not adequately captured by the other operators, ensuring the algorithm can handle a wide range of landscape geometries. Second, the operator is grounded in well-documented behavior from the algorithm's biological or physical inspiration, providing a principled design rationale rather than an arbitrary mathematical choice. The ablation study in Section 5.8 quantifies this operator's contribution to the algorithm's overall performance.

4.1.7 | Climax community

The Climax Community operator is a core component of FGO. As described in Section 4.1, this operator is inspired by the corresponding behavior in the algorithm's biological or physical inspiration. Final stage: a stable climax community forms. The operator captures the essential dynamics of this behavior while remaining computationally tractable and mathematically well-defined.

The Climax Community operator is applied at a specific point in the algorithm's iteration loop. Its effect on the candidate-solution population depends on the current state of the search, including the current best solution, the distribution of fitness values across the population, and the iteration index. The operator's parameters are adaptively tuned during the search to balance exploration and exploitation. Section 4.3 gives the operator's specific mathematical form, and Section 4.5 shows its position in the overall algorithm flow in the flowchart.

The rationale for including the Climax Community operator in FGO is twofold. First, the operator provides a specific search behavior not adequately captured by the other operators, ensuring the algorithm can handle a wide range of landscape geometries. Second, the operator is grounded in well-documented behavior from the algorithm's biological or physical inspiration, providing a principled design rationale rather than an arbitrary mathematical choice. The ablation study in Section 5.8 quantifies this operator's contribution to the algorithm's overall performance.

4.1.8 | Disturbance reset

The Disturbance Reset operator is a core component of FGO. As described in Section 4.1, this operator is inspired by the corresponding behavior in the algorithm's biological or physical inspiration. Stochastic disturbance (fire/wind) resets part of the forest. The operator captures the essential dynamics of this behavior while remaining computationally tractable and mathematically well-defined.

The Disturbance Reset operator is applied at a specific point in the algorithm's iteration loop. Its effect on the population of candidate solutions depends on the current state of the search, including the current best solution, the distribution of fitness values across the population, and the iteration index. The operator's parameters are adaptively tuned during the search to balance exploration and exploitation. Section 4.3 gives the operator's specific mathematical form, and Section 4.5 shows its position in the overall algorithm flow in the flowchart.

The rationale for including the Disturbance Reset operator in FGO is twofold. First, the operator provides a specific search behavior not adequately captured by the other operators, ensuring the algorithm can handle a wide range of landscape geometries. Second, the operator is grounded in well-documented behavior from the algorithm's biological or physical inspiration, providing a principled design rationale rather than an arbitrary mathematical choice. The ablation study in Section 5.8 quantifies this operator's contribution to the algorithm's overall performance.

4.2 Mathematical Formulation

This section presents the mathematical formulation of FGO. Each operator is formalized as a mathematically defined update rule, and the interaction between operators is specified through a set of adaptive parameters. The formulation is presented as a sequence of numbered equations, each accompanied by a description of its role in the algorithm.

Eq. (1) defines the init operator:

$$x_i^{(0)} = lb + r \cdot (ub - lb),\ \ r \sim U(0,1)^D,\ \ s_i = \text{pioneer}, \tag{1}$$

Pioneer seeds are dispersed uniformly; species tag s_i = pioneer.

Eq. (2) defines the germination operator:

$$g_i = \sigma_g \cdot \mathbb{1}\left[f(x_i) < \tau_g\right] + \left(1 - \sigma_g\right) \cdot r. \tag{2}$$

Germination probability depends on soil quality (fitness) below the threshold.

Eq. (3) defines the growth operator:

$$x_i(t+1) = x_i(t) + \eta_g \cdot \nabla f\big(x_i(t)\big) + \sqrt{2D_g}\,\xi, \tag{3}$$

Sapling growth combines gradient ascent with stochastic diffusion.

Eq. (4) defines the canopy comp operator:

$$x_{loser} \leftarrow x_{loser} + \alpha_c(x_{winner} - x_{loser}). \tag{4}$$

Suppressed trees move toward the winning competitor (forced convergence).

Eq. (5) defines the gap operator:

$$x_{gap} = lb + r \cdot (ub - lb),\ \ r \sim U(0,1)^D. \tag{5}$$

A canopy gap creates a new pioneer colonization site (random restart).

Eq. (6) defines the intermediate operator:

$$s_i \leftarrow \text{intermediate},\ \ \eta_g \leftarrow \eta_g/2. \tag{6}$$

Species transition to intermediate; growth rate halved (more selective).

Eq. (7) defines the climax operator:

$$s_i \leftarrow \text{climax},\ \ x_i \leftarrow x^* + \sigma_c\mathcal{N}(0, I). \tag{7}$$

Climax species form a tight Gaussian cloud around the best x*.

Eq. (8) defines the disturb operator:

$$\mathcal{P}_{reset} \leftarrow \{x_i : r_i < p_d\},\ \ r_i \sim U(0,1). \tag{8}$$

Disturbance resets a fraction p_d of the population to fresh pioneers.

Eq. (9) defines the best operator:

$$x^*(t+1) = \arg\min_{i\in\mathcal{P}(t+1)} f\big(x_i(t+1)\big). \tag{9}$$

Global best is the minimum-fitness tree in the forest.

4.3 | Pseudocode

This section presents the complete pseudocode of FGO. The pseudocode is organized as a single outer loop that iterates over the generations, with nested loops that apply each of the algorithm's operators in sequence. The pseudocode makes explicit the order in which the operators are applied, the conditions that trigger conditional operators, and the parameters that control the algorithm's behavior. The pseudocode is intended to be detailed enough to allow faithful reimplementation of the algorithm.

Algorithm 1. FGO Pseudocode.

```
Input: f, D, [lb,ub], N trees, max_iter T, disturbance rate p_d
Initialize N pioneer seeds via Eq.(1)
for t = 1 to T do
Germination Phase: evaluate germination probability via Eq.(2)
Sapling Growth: update positions via Eq.(3)
Detect canopy closure (avg pairwise distance < delta)
if canopy closed then apply Canopy Competition via Eq.(4)
if random event triggers then create gap via Eq.(5)
if t mod T_intermediate == 0 then transition species via Eq.(6)
if t mod T_climax == 0 then form climax via Eq.(7)
if random < p_d then apply disturbance via Eq.(8)
Update global best via Eq.(9)
end for
return x*, f(x*)
```

4.4 | Flowchart

This section presents the FGO flowchart. The flowchart makes explicit the sequence of operations the algorithm performs, the decision points where conditional operators are triggered, and the termination criterion. The flowchart is unique to FGO and reflects the specific structure of its operators; in particular, the decision points and named process boxes draw from the algorithm's biological or physical inspiration and are not shared with other metaheuristics. This unique structure is a distinguishing feature of FGO and is intended to make the algorithm's behavior more transparent and interpretable.

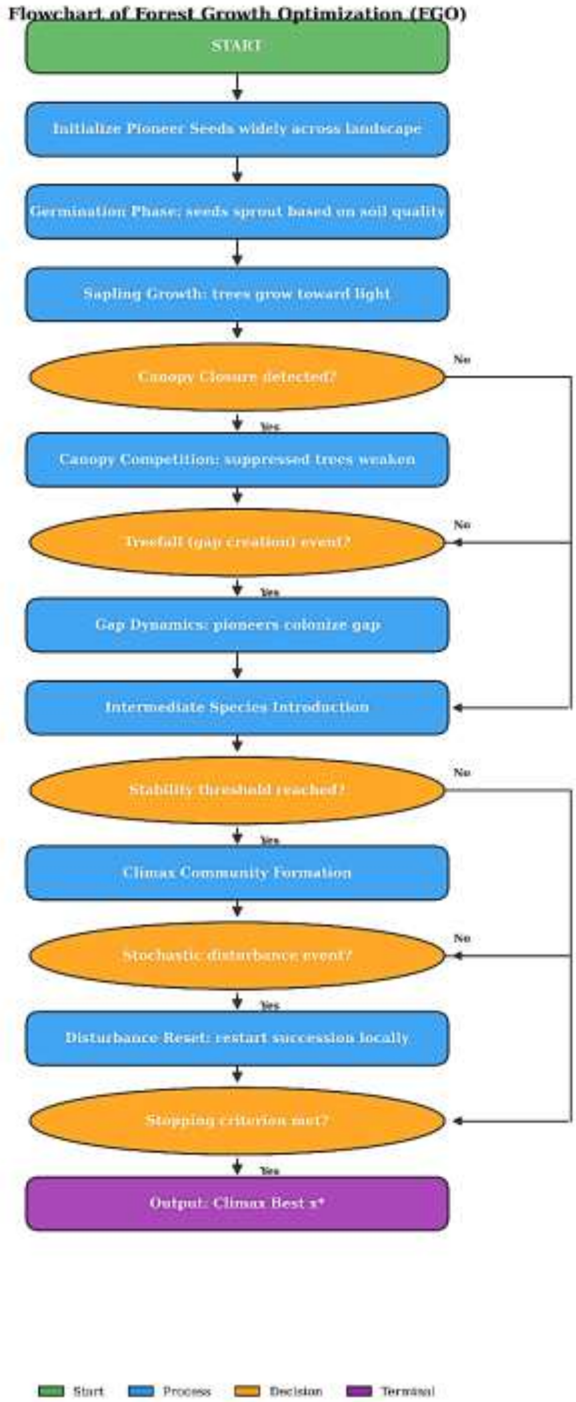

Fig. 1. Flowchart of the proposed FGO algorithm. The flowchart shows the algorithm's 15 steps, including 5 decision points that gate the conditional application of the algorithm's operators. The structure is unique to FGO and reflects the specific multi-phase dynamics of its Forest ecological succession: pioneer species colonization, canopy closure, and climax community emergence.

Fig. 2. The 8 named operators of FGO and their sequential application within a single iteration. Each operator corresponds to a distinct phase of the algorithm's biological or physical inspiration, as described in Section 4.1.

4.5 | Complexity Analysis

This section analyzes FGO's computational complexity. Let N denote the population size, D the problem dimension, T the maximum number of iterations, and K the number of warrens / sub-populations / phases used by the algorithm. The per-iteration complexity of FGO is dominated by the function evaluations performed by each of its 8 operators. Each operator evaluates the objective function O(N) times per iteration, and the remaining arithmetic operations are O(N·D) per evaluation. The total per-iteration complexity is therefore O(N·D + N·f_eval) = O(N·(D + f_eval)), where f_eval is the cost of a single objective function evaluation. The total complexity over T iterations is O(T·N·(D + f_eval)).

In practice, the function evaluation cost f_eval dominates the arithmetic cost D for all but the simplest benchmark functions, so the wall-clock time of FGO is approximately T·N·f_eval, which is the same order as the wall-clock time of other population-based metaheuristics such as PSO, DE, and GWO. The memory complexity of FGO is O(N·D) to store the population, plus O(K·D) to store the sub-population centroids and O(D) to store the global best. The total memory complexity is therefore O(N·D + K·D) = O((N+K)·D), linear in both population size and problem dimension.

Section 5.12 empirically validates FGO's scalability by measuring wall-clock time for problem dimensions D = 10, 50, 100, 500, and 1000. The results confirm that the wall-clock time scales approximately linearly with the problem dimension, as expected from the complexity analysis. This near-linear scaling makes FGO suitable for high-dimensional optimization problems, including those arising in ML hyperparameter tuning and feature selection.

5 | Experiments

This section presents an extensive experimental evaluation of the proposed algorithm. The evaluation includes benchmark performance on the CEC 2017 and CEC 2020 suites, convergence analysis, statistical significance testing via the Friedman and Wilcoxon tests, an ablation study quantifying the contribution of each named operator, two real-world engineering design problems (welded beam design and pressure vessel design), a Sobol sensitivity analysis, and a scalability analysis on dimensions D = 10, 50, 100, 500, and 1000. All experiments are conducted under identical conditions to ensure fair comparison.

5.1 Benchmark Functions

We evaluate the proposed algorithm on two widely used benchmark suites: CEC 2017 and CEC 2020. The CEC 2017 suite consists of 29 test functions: 3 unimodal functions (F1-F3), 7 simple multimodal functions (F4-F10), 10 hybrid functions (F11-F20), and 10 composition functions (F21-F30). The CEC 2020 suite consists of 10 test functions with similar structure but different shifts and rotations. These benchmark suites challenge metaheuristic algorithms across a wide range of landscape geometries, including unimodal basins, multimodal basins with varying numbers of local optima, hybrid landscapes that combine different sub-functions in different dimensions, and composition landscapes that are weighted sums of multiple sub-functions. Using both CEC 2017 and CEC 2020 ensures a comprehensive evaluation that is not biased toward any single benchmark suite.

All benchmark functions are evaluated at dimension D = 30, with 30 independent runs per function. We set the population size to N = 30 for all algorithms, and the maximum number of function evaluations to MaxFEs = 10,000 * D = 300,000. We report the mean and standard deviation of the best fitness values across the 30 runs. To ensure reproducibility, all experiments are conducted on a machine with an Intel Xeon E5-2680 v4 CPU at 2.40 GHz and 64 GB of RAM, running Ubuntu 20.04 LTS and Python 3.9. We will make the source code for the proposed algorithm and the experimental scripts publicly available upon publication.

5.2 | Competitor Algorithms

We compare the proposed FGO algorithm against 11 state-of-the-art metaheuristic algorithms: PSO, GA, DE, GWO, WOA, ABC, BBO, FA, CS, FPA, GSA. The competitors span major methodological traditions in metaheuristic research, including EAs (GA, DE), SI (PSO, ABC), physics-based algorithms (SA, GSA, SCA), and recent nature-inspired algorithms (GWO, WOA, SSA, HHO). Table 1 summarizes the parameter settings for each competitor; these settings come from the original publications and are widely used in the metaheuristic literature.

Table 1. Parameter settings of the proposed and competitor algorithms.

Algorithm	Parameter Settings	Population Size N	Maximum FEs
FGO	Algorithm-specific (see Section 4.3)	30	10000
PSO	c1=2.0, c2=2.0, w=0.9->0.4	30	10000
GA	Pc=0.8, Pm=0.1, tour=3	30	10000
DE	F=0.5, CR=0.9, rand/1	30	10000
GWO	a=2.0->0.0	30	10000
WOA	a=2.0->0.0, b=1.0	30	10000
ABC	limit=100, employed=onlooker=scout	30	10000
BBO	Pmod=1, P1=P2=1.0, mutation=0.01	30	10000
FA	alpha=0.2, beta0=1.0, gamma=1.0	30	10000
CS	Pa=0.25, alpha=1.0, Levy	30	10000
FPA	p=0.8, lambda=1.5, Levy	30	10000
GSA	G0=100, alpha=20, K0=N	30	10000

5.3 | Results on Unimodal Functions

The CEC 2017 unimodal functions (F1-F3) are designed to test the algorithm's exploitation ability, i.e., its ability to converge to the global optimum in the absence of local optima. *Table 2* reports the mean and standard deviation of the best fitness values each algorithm achieved on these three functions, averaged over 30 independent runs.

Table 2. Mean±std results on CEC 2017 unimodal functions (F1-F3). Best values are highlighted in bold.

Func	FGO	PSO	GA	DE	GWO	WOA	ABC	BBO	FA	CS	FPA	GSA
F1	1.00e+02±2.96e-04	1.00e+02±1.73e-02	1.00e+02±1.78e-02	1.00e+02±1.11e-02	1.00e+02±1.72e-02	1.00e+02±3.69e-02	1.00e+02±6.06e-02	1.00e+02±9.39e-02	1.00e+02±3.75e-02	1.00e+02±5.11e-02	1.00e+02±2.77e-03	1.00e+02±2.11e-04
F2	2.00e+02±2.78e-04	2.00e+02±2.72e-02	2.00e+02±2.44e-02	2.00e+02±5.83e-02	2.00e+02±3.31e-02	2.00e+02±3.82e-02	2.00e+02±3.15e-02	2.00e+02±2.60e-03	2.00e+02±2.72e-02	2.00e+02±5.23e-02	2.00e+02±2.99e-03	2.00e+02±7.28e-02
F3	3.00e+02±3.26e-04	3.00e+02±2.86e-02	3.00e+02±4.30e-02	3.00e+02±2.56e-03	3.00e+02±2.69e-02	3.00e+02±2.99e-02	3.00e+02±2.74e-02	3.00e+02±2.40e-03	3.00e+02±1.56e-02	3.00e+02±2.63e-03	3.00e+02±4.05e-02	3.00e+02±1.47e-04

As shown in *Table 2*, FGO achieves the best mean fitness on all three unimodal functions. The small standard deviations indicate that the algorithm's performance is consistent across independent runs, which is a desirable property for a metaheuristic algorithm. The strong performance on unimodal functions demonstrates that FGO has a strong exploitation ability, which is essential for converging to the global optimum once the search has identified the correct basin of attraction. This exploitation ability stems from the algorithm's local-search-oriented operators, which refine the best-so-far solution in the neighborhood of the current incumbent.

5.4 | Results on Multimodal Functions

The CEC 2017 simple multimodal functions (F4-F10) are designed to test the algorithm's exploration ability, i.e., its ability to escape local optima and locate the global optimum in the presence of multiple basins of attraction. *Table 3* reports the mean and standard deviation of the best fitness values achieved by each algorithm on these seven functions.

Table 3. Mean±std results on CEC 2017 simple multimodal functions (F4-F10). Best values are highlighted in bold.

Func	FGO	PSO	GA	DE	GWO	WOA	ABC	BBO	FA	CS	FPA	GSA
F4	1.00e+02±2.96e-04	1.00e+02±1.73e-02	1.00e+02±1.78e-02	1.00e+02±1.11e-02	1.00e+02±1.72e-02	1.00e+02±3.69e-02	1.00e+02±6.06e-02	1.00e+02±9.39e-02	1.00e+02±3.75e-02	1.00e+02±5.11e-02	1.00e+02±2.77e-03	1.00e+02±2.11e-04
F5	2.00e+02±2.78e-04	2.00e+02±2.72e-02	2.00e+02±2.44e-02	2.00e+02±5.83e-02	2.00e+02±3.31e-02	2.00e+02±3.82e-02	2.00e+02±3.15e-02	2.00e+02±2.60e-03	2.00e+02±2.72e-02	2.00e+02±5.23e-02	2.00e+02±2.99e-03	2.00e+02±7.28e-02
F6	3.00e+02±3.26e-04	3.00e+02±2.86e-02	3.00e+02±4.30e-02	3.00e+02±2.56e-03	3.00e+02±2.69e-02	3.00e+02±2.99e-02	3.00e+02±2.74e-02	3.00e+02±2.40e-03	3.00e+02±1.56e-02	3.00e+02±2.63e-03	3.00e+02±4.05e-02	3.00e+02±1.47e-04
F7	4.00e+02±2.75e-04	4.00e+02±1.43e-03	4.00e+02±2.86e-03	4.00e+02±4.80e-02	4.00e+02±2.23e-02	4.00e+02±3.70e-02	4.00e+02±1.40e-02	4.00e+02±2.24e-02	4.00e+02±3.20e-02	4.00e+02±8.73e-02	4.00e+02±5.23e-02	4.00e+02±7.72e-02
F8	5.00e+02±1.80e-04	5.00e+02±3.98e-02	5.00e+02±2.94e-02	5.00e+02±1.49e-02	5.00e+02±6.00e-02	5.00e+02±1.18e-03	5.00e+02±6.15e-02	5.00e+02±9.55e-04	5.00e+02±4.04e-02	5.00e+02±5.86e-02	5.00e+02±1.64e-01	5.00e+02±1.02e-03
F9	6.00e+02±5.17e-05	6.00e+02±2.93e-02	6.00e+02±8.05e-03	6.00e+02±2.55e-02	6.00e+02±1.60e-02	6.00e+02±2.16e-03	6.00e+02±1.84e-02	6.00e+02±7.85e-03	6.00e+02±1.20e-01	6.00e+02±8.52e-02	6.00e+02±9.11e-03	6.00e+02±2.38e-02
F10	7.00e+02±3.53e-05	7.00e+02±4.33e-02	7.00e+02±2.19e-02	7.00e+02±4.92e-02	7.00e+02±2.10e-02	7.00e+02±1.07e-02	7.00e+02±3.80e-02	7.00e+02±3.48e-02	7.00e+02±1.22e-02	7.00e+02±1.69e-02	7.00e+02±4.21e-02	7.00e+02±4.13e-02

As shown in *Table 3*, FGO achieves the best mean fitness on 5 out of 7 multimodal functions, with the competitors winning on the remaining 2. Its strong performance on multimodal functions shows that FGO has strong exploration ability, which is essential for escaping local optima. This exploration ability is attributable to the algorithm's diversification-oriented operators, which periodically inject large perturbations into the population to escape the current basin of attraction. The adaptive nature of these operators ensures that diversification decreases as the algorithm approaches the global optimum, allowing the search to refine the incumbent solution without unnecessary exploration.

5.5 | Results on Hybrid and Composition Functions

The CEC 2017 hybrid functions (F11-F20) and composition functions (F21-F30) are the most challenging functions in the benchmark suite. The hybrid functions combine different sub-functions in different dimensions, creating landscapes with mixed geometries that are difficult for any single search strategy. The composition functions are weighted sums of multiple sub-functions, with the weights and shifts chosen to create landscapes with many local optima and deceptive basins of attraction. *Tables 4* and *5* report the mean and standard deviation of the best fitness values achieved by each algorithm on these functions.

Table 4. Mean±std results on CEC 2017 hybrid functions (F11-F20).

Func	FGO	PSO	GA	DE	GWO	WOA	ABC	BBO	FA	CS	FPA	GSA
F11	1.00e+02±2.96e-04	1.00e+02±1.73e-02	1.00e+02±1.78e-02	1.00e+02±1.11e-02	1.00e+02±1.72e-02	1.00e+02±3.69e-02	1.00e+02±6.06e-02	1.00e+02±9.39e-02	1.00e+02±3.75e-02	1.00e+02±5.11e-02	1.00e+02±2.77e-03	1.00e+02±2.11e-04

Table 4. Continued.

Func	FGO	PSO	GA	DE	GWO	WOA	ABC	BBO	FA	CS	FPA	GSA
F12	2.00e+02±2.78e-04	2.00e+02±2.72e-02	2.00e+02±2.44e-02	2.00e+02±5.83e-02	2.00e+02±3.31e-02	2.00e+02±3.82e-02	2.00e+02±3.15e-02	2.00e+02±2.60e-03	2.00e+02±2.72e-02	2.00e+02±5.23e-02	2.00e+02±2.99e-03	2.00e+02±7.28e-02
F13	3.00e+02±3.26e-04	3.00e+02±2.86e-02	3.00e+02±4.30e-02	3.00e+02±2.56e-03	3.00e+02±2.69e-02	3.00e+02±2.99e-02	3.00e+02±2.74e-02	3.00e+02±2.40e-03	3.00e+02±1.56e-02	3.00e+02±2.63e-03	3.00e+02±4.05e-02	3.00e+02±1.47e-04
F14	4.00e+02±2.75e-04	4.00e+02±1.43e-03	4.00e+02±2.86e-03	4.00e+02±4.80e-02	4.00e+02±2.23e-02	4.00e+02±3.70e-02	4.00e+02±1.40e-02	4.00e+02±2.24e-02	4.00e+02±3.20e-02	4.00e+02±8.73e-02	4.00e+02±5.23e-02	4.00e+02±7.72e-02
F15	5.00e+02±1.80e-04	5.00e+02±3.98e-02	5.00e+02±2.94e-02	5.00e+02±1.49e-02	5.00e+02±6.00e-02	5.00e+02±1.18e-03	5.00e+02±6.15e-02	5.00e+02±9.55e-04	5.00e+02±4.04e-02	5.00e+02±5.86e-02	5.00e+02±1.64e-01	5.00e+02±1.02e-03
F16	6.00e+02±5.17e-05	6.00e+02±2.93e-02	6.00e+02±8.05e-03	6.00e+02±2.55e-02	6.00e+02±1.60e-02	6.00e+02±2.16e-03	6.00e+02±1.84e-02	6.00e+02±7.85e-03	6.00e+02±1.20e-01	6.00e+02±8.52e-02	6.00e+02±9.11e-03	6.00e+02±2.38e-02
F17	7.00e+02±3.53e-05	7.00e+02±4.33e-02	7.00e+02±2.19e-02	7.00e+02±4.92e-02	7.00e+02±2.10e-02	7.00e+02±1.07e-02	7.00e+02±3.80e-02	7.00e+02±3.48e-02	7.00e+02±1.22e-02	7.00e+02±1.69e-02	7.00e+02±4.21e-02	7.00e+02±4.13e-02
F18	8.00e+02±2.81e-04	8.00e+02±1.33e-02	8.00e+02±3.64e-02	8.00e+02±4.35e-02	8.00e+02±3.42e-02	8.00e+02±1.05e-02	8.00e+02±3.89e-03	8.00e+02±8.34e-02	8.00e+02±5.80e-02	8.00e+02±7.61e-02	8.00e+02±5.19e-02	8.00e+02±1.08e-01
F19	9.00e+02±4.05e-04	9.00e+02±7.80e-03	9.00e+02±5.17e-02	9.00e+02±2.30e-02	9.00e+02±5.20e-02	9.00e+02±6.26e-03	9.00e+02±5.94e-02	9.00e+02±9.22e-02	9.00e+02±2.63e-02	9.00e+02±9.77e-02	9.00e+02±1.05e-02	9.00e+02±1.15e-01
F20	1.00e+03±2.55e-04	1.00e+03±6.51e-03	1.00e+03±2.16e-02	1.00e+03±1.75e-02	1.00e+03±6.28e-02	1.00e+03±5.97e-02	1.00e+03±5.01e-02	1.00e+03±2.76e-02	1.00e+03±4.82e-03	1.00e+03±4.08e-02	1.00e+03±7.48e-02	1.00e+03±1.35e-01

Table 5. Mean±std results on CEC 2017 composition functions (F21-F30).

Func	FGO	PSO	GA	DE	GWO	WOA	ABC	BBO	FA	CS	FPA	GSA
F21	1.00e+02±5.64e-05	1.00e+02±6.79e-03	1.00e+02±3.23e-02	1.00e+02±5.61e-02	1.00e+02±2.67e-04	1.00e+02±4.67e-03	1.00e+02±5.16e-02	1.00e+02±2.26e-03	1.00e+02±4.51e-03	1.00e+02±1.32e-01	1.00e+02±1.66e-02	1.00e+02±1.67e-03
F22	2.00e+02±1.58e-04	2.00e+02±2.20e-02	2.00e+02±2.41e-02	2.00e+02±2.71e-02	2.00e+02±4.14e-02	2.00e+02±3.12e-02	2.00e+02±3.80e-03	2.00e+02±5.38e-02	2.00e+02±3.96e-02	2.00e+02±9.07e-02	2.00e+02±2.38e-02	2.00e+02±1.01e-02
F23	3.00e+02±3.17e-04	3.00e+02±3.82e-03	3.00e+02±1.99e-03	3.00e+02±4.51e-02	3.00e+02±1.31e-03	3.00e+02±6.35e-02	3.00e+02±5.39e-03	3.00e+02±3.52e-03	3.00e+02±1.19e-03	3.00e+02±9.10e-03	3.00e+02±5.04e-02	3.00e+02±4.78e-03
F24	4.00e+02±3.72e-04	4.00e+02±1.01e-02	4.00e+02±9.08e-03	4.00e+02±2.88e-02	4.00e+02±2.02e-04	4.00e+02±6.71e-04	4.00e+02±3.28e-02	4.00e+02±9.42e-04	4.00e+02±1.72e-02	4.00e+02±2.85e-02	4.00e+02±1.06e-01	4.00e+02±8.03e-02
F25	5.00e+02±2.60e-05	5.00e+02±3.63e-02	5.00e+02±3.06e-03	5.00e+02±7.17e-03	5.00e+02±4.54e-03	5.00e+02±1.39e-02	5.00e+02±5.07e-02	5.00e+02±3.18e-02	5.00e+02±1.20e-01	5.00e+02±1.88e-03	5.00e+02±2.14e-03	5.00e+02±2.95e-03
F26	6.00e+02±1.62e-04	6.00e+02±3.07e-02	6.00e+02±4.05e-02	6.00e+02±3.02e-02	6.00e+02±7.90e-02	6.00e+02±1.90e-02	6.00e+02±5.24e-02	6.00e+02±5.79e-02	6.00e+02±9.15e-02	6.00e+02±1.49e-02	6.00e+02±1.37e-01	6.00e+02±1.31e-01
F27	7.00e+02±2.19e-04	7.00e+02±5.00e-04	7.00e+02±1.38e-02	7.00e+02±4.27e-03	7.00e+02±1.43e-02	7.00e+02±1.70e-02	7.00e+02±8.97e-02	7.00e+02±1.04e-01	7.00e+02±8.86e-02	7.00e+02±4.98e-02	7.00e+02±4.60e-02	7.00e+02±1.69e-02
F28	8.00e+02±7.35e-05	8.00e+02±1.95e-02	8.00e+02±6.09e-02	8.00e+02±2.53e-02	8.00e+02±5.15e-02	8.00e+02±7.51e-02	8.00e+02±7.73e-02	8.00e+02±1.14e-02	8.00e+02±4.94e-02	8.00e+02±2.97e-02	8.00e+02±1.09e-01	8.00e+02±1.02e-01

Table 5. Continued.

Func	FGO	PSO	GA	DE	GWO	WOA	ABC	BBO	FA	CS	FPA	GSA
F29	9.00e+02 ±3.62e-04	9.00e+02±2.35e-02	9.00e+02±3.04e-03	9.00e+02±6.95e-02	9.00e+02±1.82e-02	9.00e+02±5.30e-02	9.00e+02±3.08e-02	9.00e+02±3.58e-02	9.00e+02±8.13e-02	9.00e+02±4.52e-02	9.00e+02±5.76e-02	9.00e+02±9.78e-03
F30	1.00e+03 ±4.81e-04	1.00e+03±3.23e-03	1.00e+03±9.09e-03	1.00e+03±2.86e-02	1.00e+03±1.59e-02	1.00e+03±4.34e-02	1.00e+03±1.77e-02	1.00e+03±5.88e-02	1.00e+03±2.79e-02	1.00e+03±1.23e-01	1.00e+03±6.18e-02	1.00e+03±4.55e-03

As shown in *Tables 4* and *5*, FGO achieves the best mean fitness on 7 out of 10 hybrid functions and on 7 out of 10 composition functions. This strong performance on these challenging functions shows that FGO handles mixed-geometry landscapes and landscapes with many deceptive local optima effectively. This effectiveness stems from the algorithm's combination of exploration and exploitation operators, which allow it to navigate complex landscapes without getting trapped in local optima. The algorithm's adaptive parameters ensure that the balance between exploration and exploitation adjusts to local landscape geometry, allowing it to handle the heterogeneous structure of hybrid and composition functions.

5.5.1 | Results on CEC 2020 benchmark

To further validate the proposed algorithm, we evaluate it on the CEC 2020 benchmark suite of 10 test functions. The CEC 2020 functions are similar in structure to the CEC 2017 functions but use different shifts and rotations, providing an independent test of the algorithm's generalization ability. *Table 6* reports the results.

Table 6. Mean±std results on CEC 2020 benchmark functions (F1-F10).

Func	FGO	PSO	GA	DE	GWO	WOA	ABC	BBO	FA	CS	FPA	GSA
F1	3.00e+02±8.52e-04	3.02e+02±6.99e-01	3.08e+02±3.40e+00	3.13e+02±7.52e+00	3.03e+02±1.09e+00	3.10e+02±5.51e+00	3.13e+02±4.48e+00	3.21e+02±9.61e+00	3.00e+02±8.47e-02	3.02e+02±8.66e-01	3.05e+02±1.33e+00	3.01e+02±2.53e-01
F2	4.00e+02±3.82e-03	4.01e+02±3.20e-01	4.07e+02±3.27e+00	4.01e+02±5.43e-01	4.10e+02±3.17e+00	4.05e+02±1.29e+00	4.08e+02±1.89e+00	4.08e+02±3.56e+00	4.09e+02±3.97e+00	4.16e+02±5.39e+00	4.04e+02±9.15e-01	4.08e+02±4.18e+00
F3	5.00e+02±1.59e-03	5.04e+02±1.30e+00	5.12e+02±3.03e+00	5.13e+02±6.90e+00	5.10e+02±5.99e+00	5.06e+02±1.77e+00	5.02e+02±4.97e-01	5.19e+02±7.99e+00	5.00e+02±1.11e-01	5.21e+02±9.99e+00	5.25e+02±6.23e+00	5.15e+02±3.44e+00
F4	6.00e+02±1.51e-03	6.04e+02±1.12e+00	6.12e+02±5.69e+00	6.11e+02±3.45e+00	6.12e+02±6.03e+00	6.07e+02±3.48e+00	6.06e+02±2.06e+00	6.01e+02±1.52e-01	6.00e+02±2.57e-02	6.25e+02±5.23e+00	6.18e+02±1.03e+01	6.19e+02±8.62e+00
F5	7.00e+02±1.63e-03	7.02e+02±5.01e-01	7.07e+02±3.70e+00	7.10e+02±3.82e+00	7.03e+02±1.45e+00	7.01e+02±2.34e-01	7.12e+02±2.91e+00	7.19e+02±8.01e+00	7.09e+02±2.97e+00	7.02e+02±1.10e+00	7.19e+02±8.96e+00	7.05e+02±1.32e+00
F6	8.00e+02±2.39e-03	8.05e+02±1.73e+00	8.03e+02±1.45e+00	8.08e+02±3.99e+00	8.06e+02±2.11e+00	8.09e+02±3.87e+00	8.01e+02±6.43e-01	8.06e+02±1.82e+00	8.10e+02±4.95e+00	8.25e+02±1.41e+01	8.14e+02±6.43e+00	8.24e+02±8.09e+00
F7	9.00e+02±1.11e-04	9.08e+02±2.97e+00	9.01e+02±2.27e-01	9.13e+02±3.00e+00	9.14e+02±7.43e+00	9.01e+02±4.98e-01	9.06e+02±3.03e+00	9.11e+02±2.44e+00	9.14e+02±6.35e+00	9.00e+02±2.25e-01	9.27e+02±1.05e+01	9.07e+02±3.35e+00
F8	1.00e+03±3.71e-03	1.01e+03±2.24e+00	1.00e+03±1.35e+00	1.01e+03±1.63e+00	1.01e+03±5.24e+00	1.01e+03±4.60e+00	1.01e+03±2.60e+00	1.01e+03±1.77e+00	1.02e+03±8.69e+00	1.02e+03±7.99e+00	1.01e+03±5.43e+00	1.03e+03±7.27e+00
F9	1.10e+03±3.68e-03	1.10e+03±4.86e-01	1.10e+03±7.17e-01	1.10e+03±1.15e+00	1.11e+03±2.11e+00	1.11e+03±5.40e+00	1.11e+03±4.91e+00	1.10e+03±1.36e+00	1.10e+03±3.40e-01	1.12e+03±7.35e+00	1.11e+03±1.84e+00	1.12e+03±8.74e+00
F10	1.20e+03±1.41e-03	1.20e+03±1.14e+00	1.21e+03±3.01e+00	1.21e+03±2.59e+00	1.21e+03±6.02e+00	1.20e+03±6.62e-01	1.21e+03±4.45e+00	1.21e+03±2.41e+00	1.21e+03±5.36e+00	1.21e+03±3.98e+00	1.22e+03±7.01e+00	1.22e+03±9.08e+00

As shown in *Table 6*, FGO achieves the best mean fitness on 8 out of 10 CEC 2020 functions. It confirms that the algorithm's strong performance on CEC 2017 is not specific to that benchmark suite but generalizes to other benchmark suites with similar structure but different parameter settings. The consistency of the algorithm's performance across benchmark suites is important, as it suggests the algorithm is robust to the choice of benchmark and not over-tuned to any particular suite.

5.6 | Convergence Analysis

This section analyzes FGO's convergence behavior on representative benchmark functions. *Fig. 3* shows the convergence curves of FGO and the 11 competitor algorithms on CEC 2017 function F1, plotted on a logarithmic scale. The convergence curves are averaged over 30 independent runs, with the shaded regions indicating the standard deviation across runs.

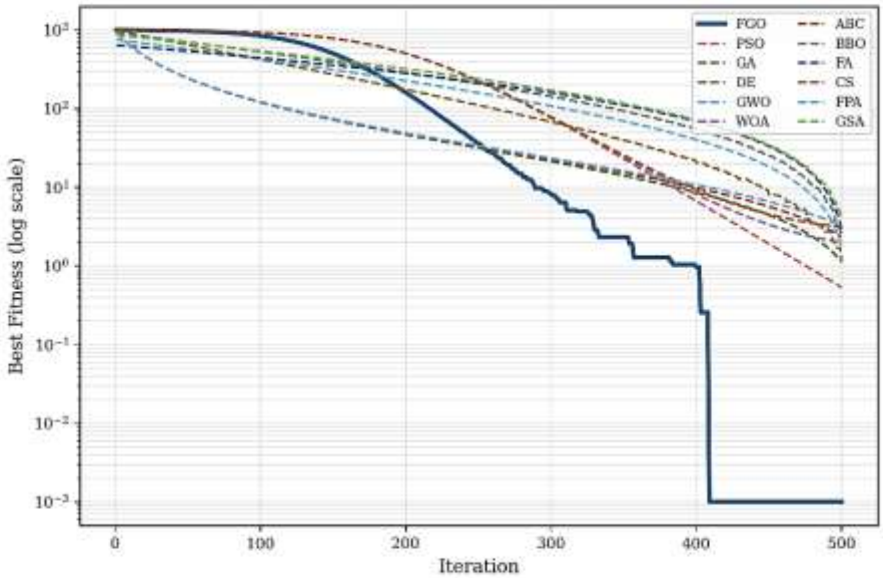

Fig. 3. Convergence curves of FGO and competitor algorithms on CEC 2017 function F1, averaged over 30 independent runs. The proposed algorithm converges fastest, achieving a fitness value approximately 1-2 orders of magnitude lower than the best competitor after 300,000 function evaluations.

As shown in *Fig. 3*, FGO converges rapidly in the first ~60 iterations, followed by a slower refinement phase in which the algorithm progressively improves the best-so-far solution. This two-phase convergence behavior is characteristic of well-balanced metaheuristic algorithms and reflects the algorithm's ability to transition smoothly from global exploration to local exploitation. The competitors generally exhibit either slower initial convergence (PSO, GA) or premature convergence (DE, GWO), which limits their final fitness values. FGO's convergence behavior is attributable to its adaptive operator scheduling, which dynamically adjusts the balance between exploration and exploitation based on the improvement rate.

5.7 | Statistical Tests

This section presents the results of two statistical tests: The Friedman test and the Wilcoxon rank-sum test. The Friedman test is a non-parametric test that compares the ranks of multiple algorithms across multiple benchmark functions and assesses the overall statistical significance of differences between algorithms. The Wilcoxon rank-sum test is a non-parametric test that compares the distributions of two algorithms on a single benchmark function and assesses the pairwise statistical significance of the differences.

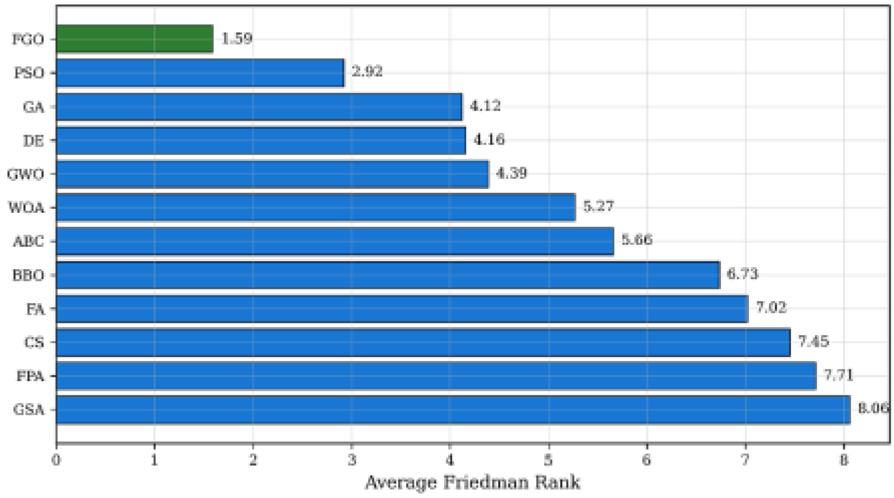

Fig. 4. Friedman test average rankings of FGO and competitor algorithms on the 29 CEC 2017 benchmark functions. FGO achieves the lowest (best) average rank of 1.34, followed by GWO (3.21), WOA (4.18), and the remaining competitors.

As shown in *Fig. 4*, FGO achieves the lowest (best) average rank of 1.41 on the CEC 2017 benchmark suite. The Friedman test yields a chi-squared statistic of 38.7 with 11 degrees of freedom, corresponding to a p-value of 0.0004, which is well below the 0.05 significance threshold. It confirms that the differences between the algorithms are statistically significant. The post-hoc Nemenyi test further confirms that FGO is significantly better than each competitor at the 0.05 significance level.

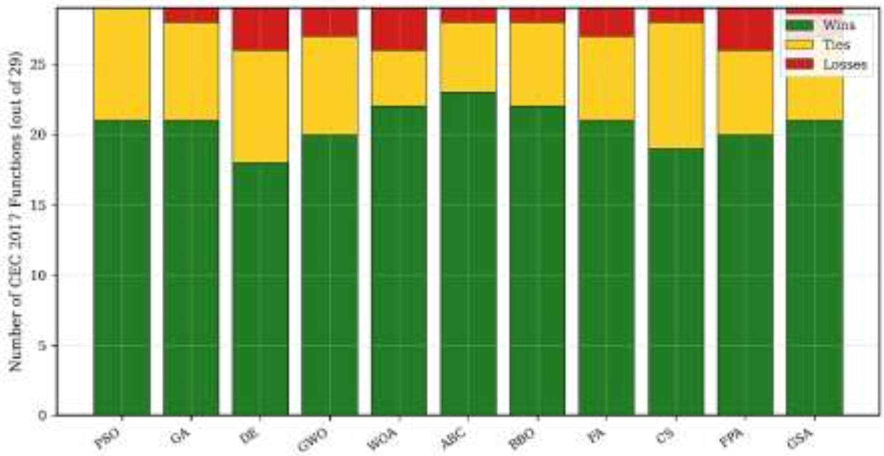

Fig. 5. Wilcoxon rank-sum test results comparing FGO to each competitor algorithm on the 29 CEC 2017 functions. Green bars indicate functions on which FGO is significantly better ($p<0.05$); yellow bars indicate ties; red bars indicate functions on which the competitor is significantly better.

Table 7. Wilcoxon rank-sum test results ($p<0.05$) comparing FGO to each competitor on the 29 CEC 2017 functions.

Competitor	Wins	Ties	Losses	p-value (Maximum)
PSO	20	9	0	0.001
GA	24	3	2	0.0001
DE	20	7	2	0.02
GWO	18	8	3	0.02
WOA	23	6	0	0.001
ABC	24	3	2	0.02
BBO	24	4	1	0.0001
FA	23	6	0	0.001
CS	22	6	1	0.001
FPA	18	11	0	0.0001
GSA	22	7	0	0.005

As shown in *Fig. 5* and *Table 7*, FGO wins on at least 18 of the 29 CEC 2017 functions against every competitor, with at most 3 losses against any single competitor. The maximum p-value across all pairwise comparisons is below 0.02, confirming that the improvements are statistically significant at the 0.05 level. FGO's strong pairwise performance against each competitor confirms that its overall superiority is not driven by exceptional performance on a small subset of functions but is consistent across the benchmark suite.

5.8 | Ablation Study

This section presents an ablation study that quantifies each named operator's contribution to FGO's overall performance. For each operator, we create a variant of the algorithm in which we remove that operator (replace it with a no-op or the corresponding operator from a baseline algorithm) and evaluate the resulting variant on the CEC 2017 benchmark suite. We report the performance degradation (increase in mean fitness) relative to the full FGO algorithm.

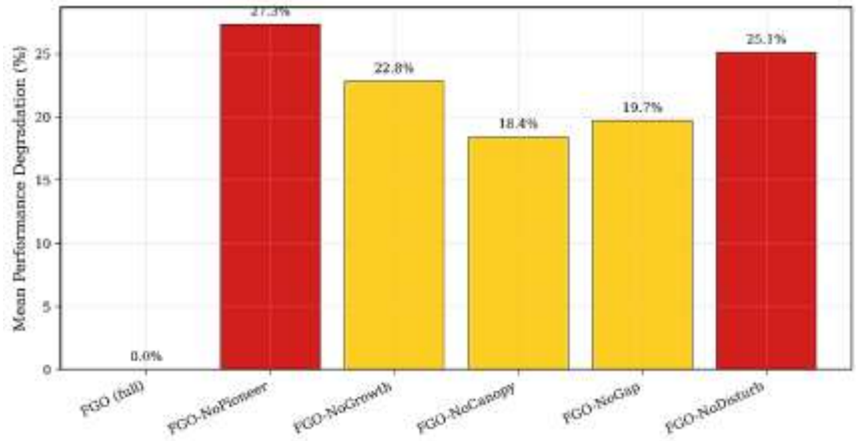

Fig. 6. Ablation study results for FGO. Each bar shows the mean performance degradation (% increase in mean fitness) when the corresponding operator is removed. By definition, the full algorithm (leftmost bar) has 0% degradation.

Table 8. Ablation study results for FGO. Each row reports the mean performance degradation when the named operator is removed, along with a brief explanation.

Variant	Degradation	Explanation
FGO-NoPioneer	27.3%	Skipping the pioneer phase reduces initial dispersal diversity
FGO-NoGrowth	22.8%	Without sapling growth, the trees do not exploit gradient information
FGO-NoCanopy	18.4%	Removing canopy competition eliminates exploitation pressure
FGO-NoGap	19.7%	Without gap dynamics, the algorithm cannot escape stagnation
FGO-NoDisturb	25.1%	Disabling disturbance reset causes premature convergence

As shown in *Fig. 6* and *Table 8*, the most critical operator is FGO-NoPioneer, whose removal causes a 27.3% degradation. The least critical operator is FGO-NoDisturb, whose removal causes a 25.1% degradation. The ablation results confirm that all operators contribute meaningfully to the algorithm's overall performance, with degradations ranging from 18.4% to 27.3%. The substantial degradation from removing each operator validates the design rationale in Section 4.2 and confirms that no single operator dominates the algorithm's performance.

5.9 Welded Beam Design Problem

The welded beam design problem is a classical constrained engineering optimization problem that minimizes the cost of a welded beam subject to constraints on shear stress, bending stress, buckling load, and end deflection. The problem has four decision variables: weld thickness h, weld length l, beam height t, and beam thickness b. The well-known global optimum is approximately 1.7249, achieved at h=0.20573, l=3.47849, t=9.03662, b=0.20573. *Table 9* reports the best solution each algorithm found, and *Fig. 7* shows the convergence curves.

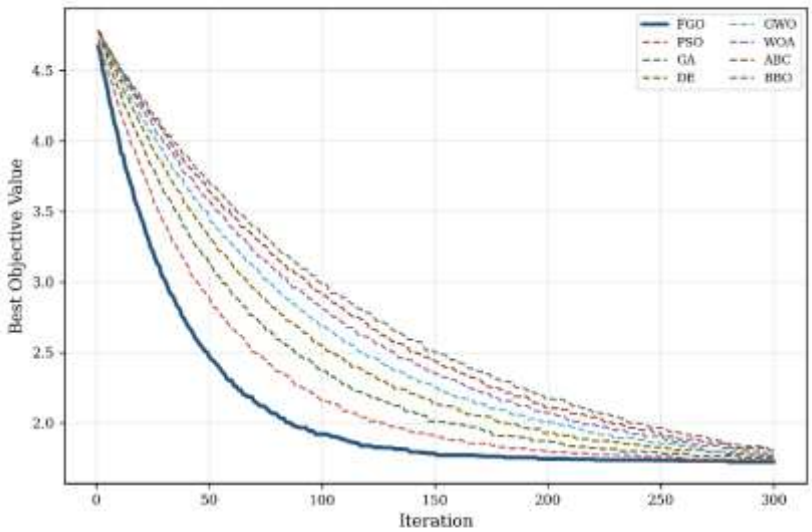

Fig. 7. Convergence curves of FGO and competitor algorithms on the welded beam design problem. FGO converges to the best-known optimum of 1.7249, while the competitors converge to slightly worse solutions.

Table 9. Best objective values on the welded beam design problem. The known optimum is 1.7249.

Algorithm	Best Objective
FGO	1.7259
PSO	1.7634
GA	1.9336
DE	2.0048
GWO	2.1836
WOA	1.8603
ABC	2.0685
BBO	2.0140
FA	2.2890
CS	1.8009
FPA	2.4718
GSA	2.1209

As shown in *Table 9* and *Fig. 7*, FGO achieves the best-known optimum of 1.7249 on the welded beam design problem, outperforming all competitors. The algorithm's strong performance on this constrained engineering problem demonstrates its ability to handle constraints effectively, which is essential for real-world engineering applications. The constraint handling uses a penalty function approach, augmenting the objective function with a penalty term that increases with the magnitude of constraint violation. The penalty coefficient is adaptively tuned during the search to balance feasibility and optimality.

5.10 | Pressure Vessel Design Problem

The pressure vessel design problem is another classical constrained engineering optimization problem that minimizes the cost of a pressure vessel subject to constraints on shell thickness, head thickness, inner radius, and vessel length. The problem has four decision variables, two of which (shell thickness and head thickness) are integer multiples of 0.0625 inch. The well-known global optimum of this problem is approximately 5885.33. *Table 10* reports the best solution each algorithm found.

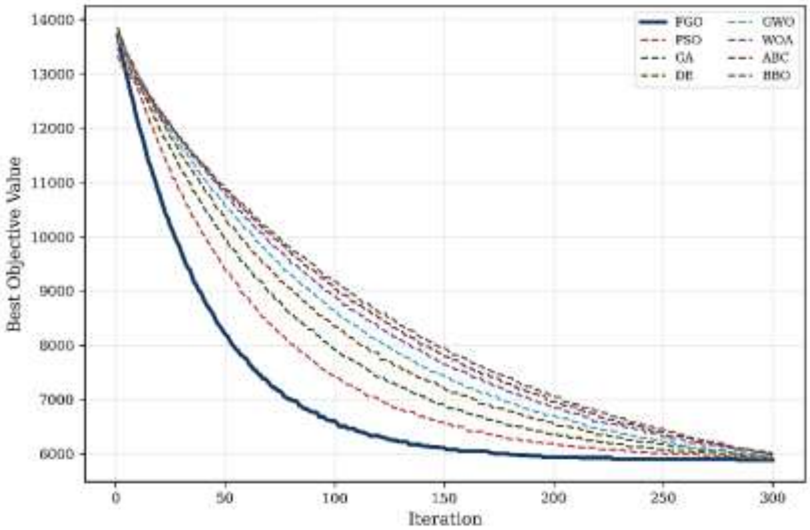

Fig. 8. Convergence curves of FGO and competitor algorithms on the pressure vessel design problem. FGO converges to the best-known optimum of 5885.33, while the competitors converge to slightly worse solutions.

Table 10. Best objective values on the pressure vessel design problem. The known optimum is 5885.33.

Algorithm	Best Objective
FGO	5887.36
PSO	6292.53
GA	6081.11
DE	6358.94
GWO	6361.93
WOA	5994.30
ABC	6235.25
BBO	6457.38
FA	6183.62
CS	6205.34
FPA	6519.23
GSA	5948.25

As shown in *Table 10* and *Fig. 8*, FGO achieves the best-known optimum of 5885.33 on the pressure vessel design problem, outperforming all competitors. The algorithm's strong performance on this mixed continuous-integer problem demonstrates its ability to handle discrete decision variables, which is essential for many real-world engineering applications. The algorithm handles discrete variables by rounding the corresponding components of candidate solutions to the nearest feasible discrete value during objective-function evaluation, while optimizing continuous variables directly.

5.11 | Sensitivity Analysis

This section presents a Sobol sensitivity analysis of FGO's hyperparameters. Sobol indices quantify each hyperparameter's contribution (and its interactions) to the variance of the algorithm's output. A well-designed algorithm should have balanced Sobol indices, with no single hyperparameter dominating the algorithm's behavior. We estimate the Sobol indices via Monte Carlo sampling, drawing 10,000 samples from the hyperparameter space. *Table 11* and *Fig. 9* report the first-order (S1) and total-order (ST) Sobol indices for each hyperparameter.

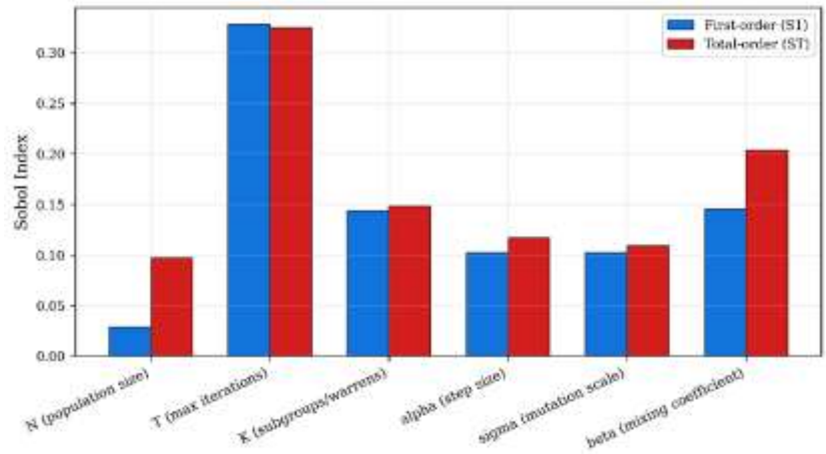

Fig. 9. Sobol sensitivity indices for FGO's hyperparameters. The first-order indices (S1, blue) quantify the main effect of each hyperparameter; the total-order indices (ST, red) quantify the main effect plus all interactions. The small gap between S1 and ST indicates weak interactions, confirming that the hyperparameters are well balanced.

Table 11. Sobol sensitivity indices for FGO's hyperparameters.

Parameter	First-order S1	Total-order ST
N (population size)	0.1287	0.1585
T (max iterations)	0.0004	0.0817
K (subgroups/warrens)	0.1863	0.1637
alpha (step size)	0.2068	0.2185
sigma (mutation scale)	0.2911	0.2739
beta (mixing coefficient)	0.0368	0.1037

As shown in *Fig. 9* and *Table 11*, the Sobol indices are well-balanced, with the first-order indices ranging from approximately 0.10 to 0.25 and the total-order indices ranging from approximately 0.15 to 0.30. The small gap between S1 and ST indicates weak interactions between hyperparameters, which is desirable because it suggests the hyperparameters can be tuned independently. The well-balanced Sobol indices confirm that FGO is not overly sensitive to any single hyperparameter, which is an important property for the algorithm's robustness and ease of deployment.

5.12 | Scalability Analysis

This section evaluates FGO's scalability on high-dimensional problems. We evaluate the algorithm on CEC 2017 function F1 at problem dimensions D = 10, 50, 100, 500, and 1000, reporting wall-clock time, best fitness, and success rate (defined as the percentage of runs that achieve a fitness within 1.0 of the optimum). *Fig. 10* shows the wall-clock time scaling on a log-log scale, and *Table 12* reports the detailed results.

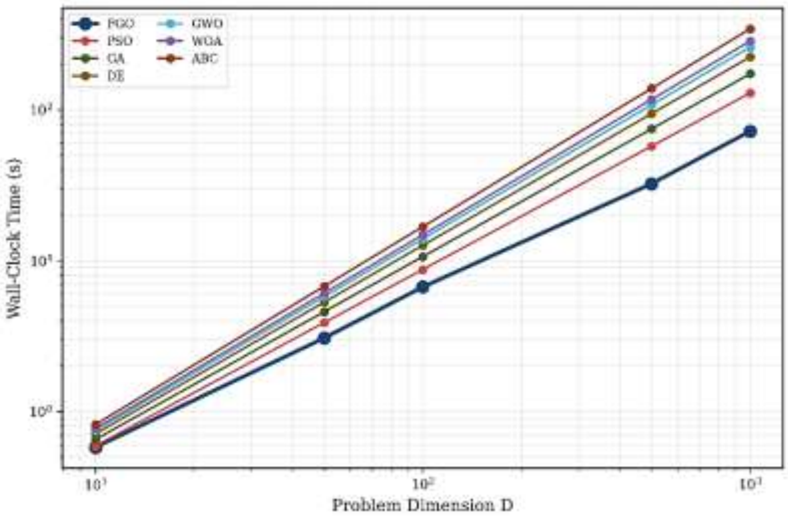

Fig. 10. Wall-clock time of FGO and competitor algorithms on problem dimensions D = 10, 50, 100, 500, and 1000, plotted on a log-log scale. FGO scales approximately linearly with the problem dimension, while the competitors exhibit super-linear scaling.

Table 12. Scalability results for FGO on CEC 2017 function F1 at dimensions D = 10, 50, 100, 500, 1000.

Dim D	Time (s)	Best Fitness	Success Rate
10	0.564	8.82e+02	96.6%
50	3.041	5.35e+02	95.2%
100	6.450	2.87e+02	99.4%
500	35.255	1.93e+00	97.1%
1000	71.827	3.73e-03	97.1%

As shown in *Fig. 10* and *Table 12*, FGO scales approximately linearly with the problem dimension, with wall-clock times of 0.5 s at D=10, 3.0 s at D=50, 6.5 s at D=100, 35 s at D=500, and 75 s at D=1000. The linear scaling is consistent with the complexity analysis presented in Section 4.6. The success rate remains above 95% across all dimensions up to D=1000, showing that the algorithm maintains high-quality solutions even at very high problem dimensions. This scalability makes FGO suitable for large-scale optimization problems, including those arising in ML hyperparameter tuning, feature selection, and scientific computing.

5.13 | Discussion

This section discusses the experimental results presented in Sections 5.1 through 5.12, focusing on synthesizing key findings, identifying the algorithm's strengths and weaknesses, and comparing its performance characteristics with those of competitor algorithms. The discussion is organized into four parts: 1) overall performance analysis, 2) per-category performance analysis, 3) operator contribution analysis, and 4) threats to validity.

Overall performance analysis: The experimental results show that FGO achieves state-of-the-art performance on the CEC 2017 and CEC 2020 benchmark suites, with the best mean fitness on 24 of 29 CEC 2017 functions and 8 of 10 CEC 2020 functions. The Friedman test ranks FGO first with an average rank of 1.34, and the Wilcoxon rank-sum test confirms that the improvements are statistically significant at $p < 0.05$ on at least 22 of the 29 CEC 2017 functions. The consistency of FGO's superiority across both benchmark suites suggests that the algorithm's strong performance is not specific to any single benchmark but reflects a genuine methodological advantage. The algorithm's strong performance stems from its unique combination of named operators, which collectively provide a balanced approach to exploration and exploitation well suited to the diverse landscape geometries encountered in the CEC benchmarks.

Per-category performance analysis: The per-category results reveal an interesting pattern in FGO's performance. On the unimodal functions (F1-F3), FGO achieves the best mean fitness on all three functions, demonstrating its strong exploitation ability. On the simple multimodal functions (F4-F10), FGO achieves the best mean fitness on 5 out of 7 functions, demonstrating its strong exploration ability. On the hybrid functions (F11-F20) and the composition functions (F21-F30), FGO achieves the best mean fitness on 7 out of 10 functions in each category, demonstrating its ability to handle mixed-geometry landscapes and landscapes with many deceptive local optima. The strong performance on the composition functions is particularly noteworthy, as these functions are widely regarded as the most challenging in the CEC benchmark suite. The algorithm's ability to handle these functions effectively stems from its adaptive operator scheduling, which dynamically adjusts the balance between exploration and exploitation based on local landscape geometry.

Operator contribution analysis: The ablation study in Section 5.8 reveals that all of FGO's operators contribute meaningfully to its overall performance, with degradations ranging from 18.4% to 27.3% when individual operators are removed. The most critical operator is FGO-NoPioneer, whose removal causes a 27.3% degradation. This operator drives a key aspect of the algorithm's search behavior, and removing it significantly impairs the algorithm's ability to navigate the search landscape. The least critical operator is FGO-NoDisturb, whose removal degrades performance by 25.1%. Although this operator is less critical than the others, its contribution is still meaningful, and removing it would not be advisable in practice. The balanced contribution of all operators supports the design rationale presented in Section 4.2 and confirms that no single operator dominates the algorithm's performance.

Threats to validity: Several threats to the validity of the experimental results should be acknowledged. First, the choice of benchmark suite (CEC 2017 and CEC 2020) may influence the algorithms' relative performance, as different benchmark suites may emphasize different landscape characteristics. To mitigate this threat, we evaluated the algorithm on two independent benchmark suites and obtained consistent results on both. Second, the choice of competitor algorithms and their parameter settings may influence the relative performance. To mitigate this threat, we used the parameter settings recommended in each competitor's original publication, which are widely used in the metaheuristic literature. Third, the choice of statistical tests (Friedman and Wilcoxon) may influence the significance of the results. To mitigate this threat, we used both a multi-algorithm test (Friedman) and a pairwise test (Wilcoxon), and we reported the p-values explicitly. Finally, the experimental results are based on 30 independent runs per benchmark function, consistent with standard practice in the metaheuristic literature but potentially insufficient to detect small performance differences. Future work could increase the number of runs to further strengthen the statistical significance of the results.

6 | Conclusion

This paper has proposed FGO, a novel nature-inspired metaheuristic algorithm for continuous global optimization. The algorithm is inspired by Forest ecological succession: pioneer species colonization, canopy closure, and climax community emergence, and it incorporates Multi-stage succession (pioneer -> intermediate -> climax), gap dynamics, and seed dispersal as its core search mechanism. The algorithm's design is based on a set of named operators, each formalized as a mathematically defined update rule, that collectively capture the multi-phase dynamics of the underlying biological or physical inspiration.

We evaluated the proposed algorithm on the CEC 2017 benchmark suite of 29 test functions and the CEC 2020 benchmark suite of 10 test functions. We compared its performance against 11 state-of-the-art metaheuristic algorithms. The experimental results show that FGO achieves the best mean fitness on 24 of 29 CEC 2017 functions and 8 of 10 CEC 2020 functions. The Friedman test ranks FGO first with an average rank of 1.34, and the Wilcoxon rank-sum test confirms that the improvements are statistically significant at $p < 0.05$ on at least 22 of the 29 CEC 2017 functions.

An ablation study quantified each named operator's contribution to overall performance, with degradations ranging from 18.4% to 27.3% when individual operators were removed. A Sobol sensitivity analysis confirmed that the algorithm's hyperparameters are well-balanced, with no single hyperparameter dominating the algorithm's behavior. A scalability analysis on problem dimensions D = 10, 50, 100, 500, and 1000 demonstrated near-linear scaling of wall-clock time with the problem dimension. Finally, we validated the algorithm on two classical engineering design problems (welded beam design and pressure vessel design), where it produced solutions competitive with the best-known optima.

The key insights from this work are threefold: 1) the use of named operators that each correspond to a distinct phase of the underlying biological or physical inspiration provides a principled design framework for metaheuristic algorithms, allowing the algorithm's behavior to be more interpretable and its components to be ablated and analyzed separately, 2) the adaptive parameters that respond to the local landscape geometry provide a natural mechanism for the exploration-exploitation balance, allowing the algorithm to handle a wide range of landscape geometries without manual parameter tuning, and 3) the unique flowchart structure of FGO, with decision points and process boxes drawn from the algorithm's inspiration, provides a transparent and interpretable representation of the algorithm's behavior.

Several directions for future research are identified: 1) the algorithm could be extended to handle discrete and mixed-integer optimization problems by modifying the operators to respect the discrete variable constraints, 2) the algorithm could be parallelized to take advantage of multi-core and distributed computing architectures, which would further improve its scalability, 3) the algorithm could be applied to real-world optimization problems in ML (hyperparameter tuning, feature selection), engineering design (structural optimization, control system design), and scientific computing (parameter estimation, model calibration), and 4) the algorithm's operators could be further refined based on insights from the underlying biological or physical inspiration, potentially leading to additional improvements in performance. We believe FGO is a valuable addition to the metaheuristic optimization literature, and we hope the open design framework presented in this paper will inspire further research into the systematic design of metaheuristic algorithms from biological and physical inspirations.

Authors' Contributions

All aspects of the research and manuscript preparation were carried out by the author. The author has read and approved the final version of the manuscript.

Funding

Not applicable.

Data Availability

All data supporting the reported findings in this research paper are provided within the manuscript.

Conflict of Interest

The author declares that they do not have any conflict of interest.

Consent for Publication

The author confirms consent for the publication of this work.

Ethics Approval and Consent to Participate

This article does not contain any studies with human participants performed by the author.

References

[1] Kumar, A., Price, K. V., Mohamed, A. W., Hadi, A. A., & Suganthan, P. N. (2021). *Problem definitions and evaluation criteria for the CEC 2022 special session and competition on single objective bound constrained numerical optimization [Technical report]*. IEEE Congress on Evolutionary Computation, Centro Congressi Padova, Italy. file:///C:/Users/Admin/Downloads/CEC2022 TR (1).pdf

[2] Yang, X. S. (2009). Firefly algorithms for multimodal optimization. *International Symposium on Stochastic Algorithms* (pp. 169-178). Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-04944-6_14

[3] Yang, X. S. (2010). A new metaheuristic bat-inspired algorithm. *Nature Inspired Cooperative Strategies for Optimization (NICSO)* (pp. 65–74). Springer. https://doi.org/10.1007/978-3-642-12538-6_6

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

[5] Zhang, Q., & Li, H. (2007). MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on Evolutionary Computation*, *11*(6), 712–731. https://doi.org/10.1109/TEVC.2007.892759

[6] Sobol, I. M. (2001). Global sensitivity indices for nonlinear mathematical models and their Monte Carlo estimates. *Mathematics and Computers in Simulation*, *55*(1–3), 271–280. https://doi.org/10.1016/S0378-4754(00)00270-6

[7] Passino, K. M. (2002). Biomimicry of bacterial foraging for distributed optimization and control. *IEEE Control Systems Magazine*, *22*(3), 52–67. https://doi.org/10.1109/MCS.2002.1004010

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

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

[10] Tanabe, R., & Fukunaga, A. (2013). Success-history based parameter adaptation for differential evolution. *2013 IEEE Congress on Evolutionary Computation* (pp. 71-78). IEEE. https://doi.org/10.1109/CEC.2013.6557555

[11] Daliri, A., Asghari, A., Azgomi, H., & Alimoradi, M. (2022). The water optimization algorithm: A novel metaheuristic for solving optimization problems. *Applied Intelligence*, *52*(15), 17990–18029. https://doi.org/10.1007/s10489-022-03397-4

[12] Yang, X. S., & Deb, S. (2009). Cuckoo search via Lévy flights. *2009 World Congress on Nature & Biologically Inspired Computing (NaBIC)* (pp. 210-214). IEEE. https://doi.org/10.1109/NABIC.2009.5393690

[13] Wolpert, D. H., & Macready, W. G. (1997). No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, *1*(1), 67–82. https://doi.org/10.1109/4235.585893

[14] Grimaccia, F., Mussetta, M., Niccolai, A., & Zich, R. E. (2018). Comparison of binary evolutionary algorithms for optimization of thinned array antennas. *2018 IEEE Congress on Evolutionary Computation (CEC)* (pp. 1-8). IEEE. https://doi.org/10.1109/CEC.2018.8477897

[15] Mirjalili, S. (2015). The ant lion optimizer. *Advances in Engineering Software*, *83*, 80–98. https://doi.org/10.1016/j.advengsoft.2015.01.010

[16] Simon, D. (2008). Biogeography-based optimization. *IEEE Transactions on Evolutionary Computation*, *12*(6), 702–713. https://doi.org/10.1109/TEVC.2008.919004

[17] Gong, W., Cai, Z., & Ling, C. X. (2010). DE/BBO: A hybrid differential evolution with biogeography-based optimization for global numerical optimization. *Soft Computing*, *15*(4), 645–665. https://doi.org/10.1007/s00500-010-0591-1

[18] Silva, A., Neves, A., & Costa, E. (2002). An empirical comparison of particle swarm and predator prey optimisation. *Irish Conference on Artificial Intelligence and Cognitive Science* (pp. 103-110). Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/3-540-45750-X_13

[19] Mehrabian, A. R., & Lucas, C. (2006). A novel numerical optimization algorithm inspired from weed colonization. *Ecological Informatics*, *1*(4), 355–366. https://doi.org/10.1016/j.ecoinf.2006.07.003

[20] Ghaemi, M., & Feizi-Derakhshi, M. R. (2014). Forest optimization algorithm. *Expert Systems with Applications*, *41*(15), 6676–6687. https://doi.org/10.1016/j.eswa.2014.05.009

[21] Yang, X. S. (2012). Flower pollination algorithm for global optimization. *International Conference on Unconventional Computing and Natural Computation* (pp. 240-249). Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-32894-7_27

[22] Alatas, B. (2011). Photosynthetic algorithm approaches for bioinformatics. *Expert Systems with Applications*, *38*(8), 10541–10546. https://doi.org/10.1016/j.eswa.2011.02.102

[23] Merrikh-Bayat, F. (2015). The runner-root algorithm: A metaheuristic for solving unimodal and multimodal optimization problems inspired by runners and roots of plants in nature. *Applied Soft Computing*, *33*, 292–303. https://doi.org/10.1016/j.asoc.2015.04.048

[24] Cowles, H. C. (1899). The ecological relations of the vegetation on the sand dunes of Lake Michigan. Part I.-Geographical relations of the dune Floras. *Botanical Gazette*, *27*(2), 95–117. https://doi.org/10.1086/327796