eng

MDPI

Article

Rain-Cloud Condensation Optimizer: Novel Nature-Inspired Metaheuristic for Solving Engineering Design Problems

Sandi Fakhouri [1], Amjad Hudaib [1], Azzam Sleit [1] and Hussam N. Fakhouri [2,*]

[1] Faculty of King Abdullah II, School of Information Technology, The University of Jordan, Amman 11196, Jordan
[2] Data Science and Artificial Intelligence Department, University of Petra, Amman 11196, Jordan
* Correspondence: hussam.fakhouri@uop.edu.jo

Abstract

This paper presents Rain-Cloud Condensation Optimizer (RCCO), a nature-inspired metaheuristic that maps cloud microphysics to population-based search. Candidate solutions ("droplets") evolve under a dual-attractor dynamic toward both a global leader and a rank-weighted cloud core, with time-decaying coefficients that progressively shift emphasis from exploration to exploitation. Diversity is preserved via domain-aware coalescence and opposition-based mirroring sampled within the coordinate-wise band defined by two parents. Rare heavy-tailed "turbulence gusts" (Cauchy perturbations) enable long jumps, while a wrap-and-reflect scheme enforces feasibility near the bounds. A sine-map initializer improves early coverage with negligible overhead. RCCO exposes a small hyperparameter set, and its per-iteration time and memory scale linearly with population size and problem dimension. RCOO has been compared with 21 state-of-the-art optimizers, over the CEC 2022 benchmark suite, where it achieves competitive to superior accuracy and stability, and achieves the top results over eight functions, including in high-dimensional regimes. We further demonstrate constrained, real-world effectiveness on five structural engineering problems—cantilever stepped beam, pressure vessel, planetary gear train, ten-bar planar truss, and three-bar truss. These results suggest that a hydrology-inspired search framework, coupled with simple state-dependent schedules, yields a robust, low-tuning optimizer for black-box, nonconvex problems.

Keywords: rain-cloud optimizer (RCCO); nature-inspired; heavy-tailed perturbations; opposition-based learning; engineering design optimization

check for updates

Academic Editor: Antonio Gil Bravo

Received: 7 September 2025
Revised: 25 September 2025
Accepted: 13 October 2025
Published: 21 October 2025

Citation: Fakhouri, S.; Hudaib, A.; Sleit, A.; Fakhouri, H.N . Rain-Cloud Condensation Optimizer: Novel Nature-Inspired Metaheuristic for Solving Engineering Design Problems. *Eng* **2025**, *6*, 281. https://doi.org/10.3390/eng6100281

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Optimization lies at the core of scientific discovery and engineering design, where the goal is to identify decision variables that minimize or maximize an objective under constraints [1]. Real-world problems often exhibit nonconvex landscapes, high dimensionality, nonlinearity, noise, and expensive evaluations. Classical deterministic methods—while powerful for smooth, convex problems with available gradients—struggle when objectives are black-box, discontinuous, or multi-modal [2]. In such cases, stochastic search methods are preferred because they require minimal assumptions about the objective and can flexibly incorporate constraints, multiple objectives, and domain heuristics.

Metaheuristics are high-level search strategies that orchestrate neighborhood exploration, adaptive sampling, and information sharing to guide a population (or trajectory) toward high-quality solutions [3]. They emphasize two complementary

forces—diversification (exploration) and intensification (exploitation)—and use randomness to avoid bias and premature convergence [4]. The No Free Lunch (NFL) results assert that no single optimizer dominates across all problems, motivating a continuing need for methods that offer robust performance across classes of problems and that can be tailored to specific structures (e.g., separability, sparsity, or expensive constraints) [5]. Contemporary metaheuristics further incorporate parameter control, adaptive memory, restart strategies, and surrogate modeling to improve efficiency and reliability.

Within metaheuristics, nature-inspired methods have become a prominent family. These include evolutionary algorithms [6] (e.g., genetic algorithms and evolution strategies), swarm intelligence [7] (e.g., particle swarm optimization and ant colony optimization), physics- and chemistry-inspired approaches [8] (e.g., simulated annealing and electromagnetic-like mechanisms), and ecology-/epidemiology-inspired dynamics. Properly designed, nature-inspired algorithms offer intuitive metaphors for information flow, local and global guidance, and population diversity management. At the same time, rigorous algorithmic design—clear operator definitions, principled parameterization, computational complexity analysis, and transparent benchmarking—remains crucial to ensure that metaphors translate into genuine search advantages.

Despite substantial progress, longstanding challenges persist: balancing exploration and exploitation in rugged landscapes; preventing stagnation and loss of diversity; scaling to higher dimensions; handling constraints efficiently; and maintaining performance in noisy or dynamically changing objectives [9]. Many practical problems also exhibit funneling structures and clustered basins, suggesting potential gains from multi-scale search and state-dependent step adaptation. These observations motivate the development of a new, hydrological cycle-inspired metaheuristic that operationalizes evaporation, condensation, cloud drift, and precipitation as coordinated search mechanisms.

This work introduces the Rain-Cloud Condensation Optimizer (RCCO) [10], a novel nature-inspired metaheuristic grounded in the microphysics of cloud formation and rainfall. RCCO models candidate solutions as moisture parcels (droplets) that evolve through phases analogous to evaporation, condensation, coalescence, cloud drift, and precipitation. Each phase is mapped to a specific search function: initialization and large-step diversification, elite-guided attraction with adaptive step size, information mixing via pairwise differentials, directional bias at the population level, and restart-like reinjection near promising regions. A state variable—supersaturation—regulates the transition between phases, providing an adaptive, problem-agnostic mechanism to balance exploration and exploitation.

The key contribution of this work is a hydrology-inspired optimizer that was deliberately designed to reduce tuning burden while addressing common pathologies of population search. By tying all search pressures to simple, linearly decaying schedules in t/T, RCCO preserves adaptivity with only a minimal hyperparameter set (N, T, and two operator probabilities), thereby lowering the barrier to practical deployment across heterogeneous tasks where problem-specific calibration is costly or impractical. RCCO effectiveness on CEC2022 benchmarks (unimodal, multi-modal, hybrid, and composite), including scalable, high-dimensional cases.

2. Related Work

The past decade has witnessed the introduction of numerous metaheuristic optimizers that draw inspiration from natural, biological, and mathematical processes [11]. Such algorithms have been designed to cope with the nonconvex and multi-modal nature of modern engineering problems, often emulating cooperative behavior in animal groups or abstracting the physics of complex phenomena [12]. In one of the earliest examples from our sample, the Optics-Inspired Optimization algorithm (OIO) treats the search space as a

mirror whose peaks and valleys are reflected to construct a concave counterpart, thereby guiding candidate solutions towards promising regions [13]. Later work adapted concepts from bird navigation: the high-level target navigation pigeon-inspired optimization (HTNPIO) uses strategies such as selective mutation and enhanced landmark search to speed convergence and escape local minima [14]. Similar ideas were used in the tree seed optimization technique, where the dispersal behavior of seeds determines how features are selected for support vector machines to detect malicious Android data [15].

Many optimizers explicitly mimic the survival strategies of animals. The gazelle optimization algorithm models the escape and pursuit phases of gazelles confronted with predators; by alternating between exploration and exploitation, it shows competitive performance on benchmark functions [16]. A complementary example is the nutcracker optimizer, which translates the mechanism by which nutcrackers crack shells into a rescheduling method for congestion management in power systems [17]. Another bio-inspired design uses the proliferation of cancer cells: the Liver Cancer Algorithm (LCA) simulates tumor growth and takeover processes to balance local and global searches [18]. Even microorganisms have inspired algorithms: the coronavirus metaheuristic algorithm (CMOA) uses metabolic transformation under various conditions to model candidate interactions and preserves diversity to avoid premature convergence [19].

Predatory behavior continues to motivate new search schemes. The migration search algorithm imitates the leader–follower dynamics within animal groups and divides the population into leaders, followers, and joiners to enhance information dissemination [20]. The bacterial foraging optimization algorithm has been adapted to optimize the cantilever beam parameters by emulating the chemotactic search patterns of bacteria [21]. Equally intriguing are the predator–prey models inspired by marine life. The orca predation algorithm assigns different weights to driving, encircling, and attacking phases in order to balance exploration and exploitation [22]. The Humboldt squid optimization algorithm uses attacks and fish-school escape patterns to orchestrate cooperation of subpopulations [23]. The Walrus optimizer adapts social signalling behaviors in walrus colonies to tune the trade-off between intensification and diversification [24].

Other animal-inspired approaches include the boosted African vulture optimization algorithm, which incorporates opposite learning and dynamic chaotic scaling [25]. The gooseneck barnacle optimization algorithm abstracts the hermaphroditic mating cycle of barnacles [26]. The Hunter algorithm models cooperative hunting to localize multiple tumours in biomedical imaging [27]. The squirrel search algorithm imitates the gliding behavior of squirrels to perform dynamic foraging [28].

Swarm-intelligence algorithms take inspiration from collective motion. The jellyfish search optimizer uses ocean current patterns and food attraction to steer solutions [29]. The tiki-taka algorithm models passing and positioning in football games to maintain ball possession and explore the search space [30]. Social interaction is also modeled in the membrane-inspired multiverse optimizer, where each membrane evolves in its own subpopulation and shares best solutions with others [31]. The leader–follower particle swarm optimizer (LFPSO) divides particles into leaders and followers to maintain diversity [32]. The modified krill herd algorithm integrates Lévy flight and crossover operators for economic dispatch problems [33].

The breadth of novel optimizers introduced in recent years underscores the vibrant state of metaheuristic research. Designers have drawn inspiration from optics and acoustics, animals and microorganisms, sports and sociocultural processes, epidemiological models, and chaotic maps. The common thread among these methods is the pursuit of a balance between exploration and exploitation through dynamic behavioral rules or hybrid combinations. Many algorithms demonstrate superior performance on benchmark problems

and real-world applications, highlighting the potential of nature-inspired computation for tackling complex optimization tasks.

Beyond the single-paradigm metaheuristics surveyed above, a substantial body of work pursues hybridization, either at the algorithm level (coupling two full optimizers) or at the operator level (importing targeted mechanisms). For example, Akopov proposes a matrix-based hybrid genetic algorithm (MBHGA) tailored to agent-based models of controlled trade, demonstrating that an encoding aligned with model structure can materially improve search efficiency [34]. A complementary line is RCGA–PSO hybrids: a real-coded GA interleaved with particle-swarm updates (and, in some instances, surrogate modeling) to combine GA's recombination with PSO's fast social drift [35]. Earlier hybrids also embed local ACO procedures inside GA to intensify search around promising alignments—e.g., the classical GA-ACO for multiple sequence alignment—illustrating the long-standing value of coupling global recombination with pheromone-guided refinement [36].

Recent work continues to explore hybridization at scale. Learning-to-rank-driven automated hybrid design composes behaviors from multiple bases (e.g., WOA, HHO, GA) on the fly [37]; deep-RL-enhanced variants of WOA address resource scheduling in industrial operating systems by guiding WOA's move selection with value estimates [38]; improved WOA variants introduce dynamic gain sharing and other schedules to tighten convergence without sacrificing global reach [39]. Parallel developments in new swarm designs and bio-inspired mechanisms (e.g., horned-lizard optimization) further enrich the design space that hybrids can draw from [40]. To sharpen the comparison, they classify representative algorithms by the mechanism used to balance exploration and exploitation, and maps each mechanism to its analog in RCCO.

Relative to algorithm-level hybrids such as MBHGA or RCGA–PSO, RCCO is an operator-level hybrid with a hydrology metaphor that fuses leader/core attraction (PSO-like), axis-aligned band sampling (GA/DE), opposition learning, and heavy-tailed jumps under a single, linearly decaying schedule, yielding few hyperparameters while preserving diversification. In line with recent taxonomies emphasizing mechanism-level reporting, principled parameter control, and standardized benchmarks/constraint handling [41,42], RCCO targets five gaps: low-overhead adaptivity (simple time-decays), late-stage stability (rank-weighted cloud core), escape from funnels (rare Cauchy gusts with decay), transparent constraint handling (wrap-and-reflect with penalties), and comparability (CEC-style suites and classical constrained designs).

3. Rain-Cloud Condensation Optimizer (RCCO)

Inspiration

RCCO draws an analogy between the optimization process and the life cycle of droplets inside a rain cloud. In the atmosphere, moist air rises and condenses on aerosols, forming droplets that are carried by updrafts toward a dense cloud core; entrainment mixes ambient air, while collisions cause coalescence, and sporadic turbulence injects energetic gusts. The algorithm mirrors these mechanisms in the search space: candidate solutions are droplets, the incumbent best acts as a leader, a rank-weighted average of the top fraction forms the cloud core, and stochastic perturbations act as thermal noise, entrainment, and turbulence.

Figure 1 refines this metaphor with a stylized schematic. Panel (a) shows droplets advected by buoyancy and shear toward the leader and the core within a softly shaded cloud. Panel (b) depicts domain-aware coalescence via axis-aligned band sampling between two parent droplets, an entrainment mirror across the domain center, and rare heavy-tailed gusts that propel droplets across distant regions. This visual framing motivates the mathematical operators introduced in the next section.

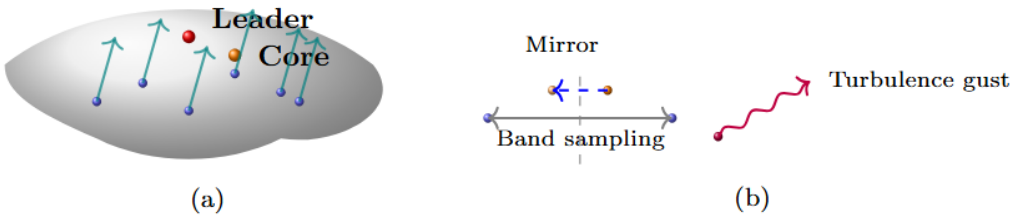


Figure 1. Stylized RCCO metaphor: (**a**) droplets rise toward the leader and core due to updrafts; (**b**) droplets interact through coalescence, mirror across the domain, and are perturbed by turbulence.

RCCO also seeds its initial population using a sine map, a chaotic map related to the tent map. Starting from a random point $z_0 \in [0, 1]$, the sine map iteratively applies $z_{k+1} = \sin(\pi z_k)$; for chaotic parameters, it yields a sequence that visits the unit interval densely, similar to the tent map, which folds and stretches the interval to produce a chaotic sequence Figure 2 plots the sine map function and the orbit of one seed point. The sine map ensures that initial droplets cover the domain evenly, improving diversity during early search.

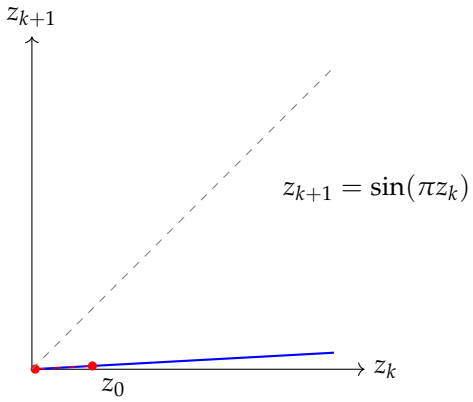


Figure 2. The sine map $z_{k+1} = \sin(\pi z_k)$ used to seed the population.

Finally, RCCO occasionally adds heavy-tailed turbulence bursts using a Cauchy distribution. The Cauchy distribution has undefined mean and variance and exhibits much heavier tails than a Gaussian distribution [10]. Figure 3 compares the probability density functions of a standard normal and a Cauchy distribution. The heavy tails increase the probability of large jumps, enabling the optimizer to escape local optima.

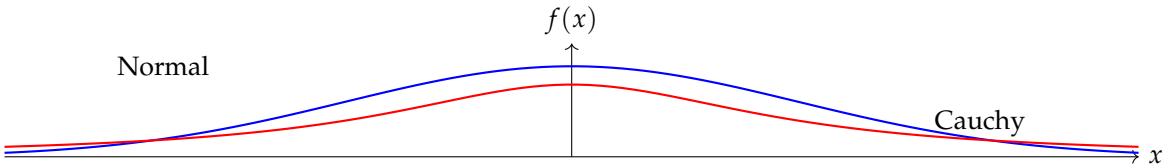


Figure 3. Comparison of a standard normal and standard Cauchy density.

4. Mathematical Model

This section formalizes the update rules underlying RCCO. Let N denote the population size, T the number of iterations, and n the dimensionality of the search space. Each candidate solution is a droplet $x_i \in \mathbb{R}^n$ with fitness $f(x_i) = \text{fobj}(x_i)$. Lower fitness values correspond to better solutions.

4.1. Condensation: Updraft to the Leader and Core

At the start of iteration t, the droplets are sorted in ascending order of fitness. The best droplet becomes the leader x_{best}, and the top $q = \max\{2, \lfloor 0.2N \rfloor\}$ droplets form the cloud core. Weighted summation produces the core point

$$c(t) = \frac{\sum_{j=1}^{q} w_j \, x_{(j)}}{\sum_{j=1}^{q} w_j}, \tag{1}$$

where (j) indicates the jth best droplet and $w_j = q - j + 1$ assigns greater weight to higher-ranked droplets. The core acts as a secondary attractor besides the leader.

Each droplet x_i is then perturbed toward both the leader and the core. The buoyancy and shear coefficients decay over time to shift the algorithm from exploration to exploitation:

$$\beta(t) = 0.70\big(1 - \tfrac{t}{T}\big) + 0.05\,\xi_1, \tag{2}$$

$$\sigma(t) = 0.30\big(1 - \tfrac{t}{T}\big), \tag{3}$$

where $\xi_1 \sim \mathcal{U}(0,1)$ introduces randomness. The thermal variance vector scales with the search range and decays as

$$\tau(t) = 0.04\Big(1 - \tfrac{t}{T}\Big)\big(ub - lb\big), \tag{4}$$

where lb and ub are lower and upper bounds. For droplet i, the condensation update is

$$x_i' = x_i + \beta(t)\big(x_{\text{best}} - x_i\big) + \sigma(t)\big(c(t) - x_i\big) + \eta, \tag{5}$$

where $\eta \sim \mathcal{N}(0, \operatorname{diag}(\tau(t)^2))$ is thermal noise. Before acceptance, the candidate is wrapped within bounds using a wrap-and-reflect operator $\text{Wrap}(\cdot)$ to model recirculation at cloud edges. If the new fitness improves upon the droplet's current fitness, the update is accepted.

4.2. Coalescence, Entrainment, and Turbulence

After condensation, each droplet undergoes an exploration phase. Two distinct parents x_p and x_q are selected uniformly at random. A trial point is sampled from the coalescence band defined by the component-wise minimum and maximum of the parents:

$$y_i = L + r \odot \big(U - L\big), \tag{6}$$

where $L = \min(x_p, x_q)$, $U = \max(x_p, x_q)$ and $r \sim \mathcal{U}(0,1)^n$ is a uniform random vector. With probability 0.35, an entrainment mirror is applied to model the ingestion of external air into the cloud:

$$y_i \leftarrow lb + ub - y_i, \tag{7}$$

and the mirrored point is used if its fitness is superior. With probability 0.18, a turbulence gust injects heavy-tailed noise drawn from a Cauchy distribution:

$$y_i \leftarrow y_i + \big(1 - \tfrac{t}{T}\big)\, 0.02\, (ub - lb) \odot \tan\big(\pi(u - \tfrac{1}{2})\big), \tag{8}$$

where $u \sim \mathcal{U}(0,1)^n$ component-wise. The tangent transform generates standard Cauchy variates. The wrap-and-reflect operator ensures the candidate remains in bounds. The update is accepted if it yields a lower fitness.

To overcome the limitation of purely time-decaying coefficients in Equations (4)–(27), we optionally adapt $\beta(t)$ and $\sigma(t)$ using feedback from the search state. Let the normalized population diversity be

$$\Delta(t) \;=\; \frac{1}{n}\sum_{d=1}^{n}\frac{\mathrm{std}\big(\{x_{i,d}\}_{i=1}^{N}\big)}{ub_d - lb_d}. \tag{9}$$

and define a short-horizon improvement rate over a window h as

$$\begin{aligned}\rho(t) &= \frac{\mathrm{RainCurve}[t-h] - \mathrm{RainCurve}[t]}{|\mathrm{RainCurve}[t-h]| + \varepsilon},\\ s(t) &= \gamma\, s(t-1) + (1-\gamma)\mathbb{1}\{\rho(t) \le 0\}.\end{aligned} \tag{10}$$

We then form adaptive coefficients

$$\begin{aligned}\tilde{\beta}(t) &= \mathrm{clip}\Big(\beta(t)\big[1 - \lambda_{\mathrm{div}}(1-\Delta(t))\big] + \lambda_{\mathrm{prog}}\max\{0,\rho(t)\},\ \beta_{\min}, \beta_{\max}\Big),\\ \tilde{\sigma}(t) &= \mathrm{clip}\Big(\sigma(t)\big[1 + \lambda_{\mathrm{div}}(1-\Delta(t)) + \lambda_{\mathrm{stag}}\, s(t)\big],\ \sigma_{\min}, \sigma_{\max}\Big).\end{aligned} \tag{11}$$

We use $(\tilde{\beta}(t), \tilde{\sigma}(t))$ in place of $(\beta(t), \sigma(t))$ inside Equation (28). Here, $\mathrm{clip}(u, a, b) = \min\{\max\{u, a\}, b\}$ enforces bounds (e.g., $\beta_{\min} = 0.02$, $\beta_{\max} = 0.80$, $\sigma_{\min} = 0$, $\sigma_{\max} = 0.50$); typical settings $h \in [5, 10]$, $\gamma = 0.9$, $\varepsilon = 10^{-12}$, and small gains $\lambda_{\mathrm{div}}, \lambda_{\mathrm{stag}}, \lambda_{\mathrm{prog}} \in [0, 0.5]$ work well. Intuitively, when diversity shrinks or progress stalls, $\tilde{\sigma}(t)$ increases (more exploration) and $\tilde{\beta}(t)$ is restrained; when progress is strong, $\tilde{\beta}(t)$ is reinforced. This state-aware control parallels DRL-guided parameter tuning reported in the metaheuristics literature [38].

We also expose the secondary operators—band sampling, mirroring, and gusts—as a portfolio Ω with selection probabilities $p_o(t)$ for $o \in \Omega$. After each iteration, we update a smoothed reward and the selection distribution:

$$\begin{aligned}r_o(t) \;&\leftarrow\; (1-\alpha)\, r_o(t-1) + \alpha\, \hat{\delta}_o(t),\\ p_o(t+1) \;&\propto\; (1-\eta)\, p_o(t) + \eta\, \mathrm{softmax}\Big(\tfrac{r_o(t)}{\tau}\Big).\end{aligned} \tag{12}$$

where $\hat{\delta}_o(t)$ is the normalized improvement credited to operator o (zero if no improvement), and $\alpha, \eta \in (0, 1)$, $\tau > 0$ are smoothing/temperature hyperparameters. This adaptive operator selection biases RCCO toward operators that are empirically effective on the current landscape, aligning with hybrid scheduling and learning-to-rank composition strategies [37]. Both adaptive modules are optional; they recover the original fixed linear schedules for full reproducibility.

5. Pseudocode

Algorithm 1 summarizes the RCCO procedure. It uses the equations defined above to update each droplet. Condensation (lines 7–14) uses the leader and core attraction defined in Equation (28). Coalescence and entrainment (lines 18–29) draw new candidates from the band (Equation (6)) and optionally apply mirroring (Equation (7)). Turbulence bursts (line 23) follow Equation (8). After each update, the candidate is wrapped into the domain. The iteration best value is recorded in the curve $\mathrm{RainCurve}[t]$.

Algorithm 1 Rain-Cloud Condensation Optimizer (RCCO)

Require: population size N, iterations T, bounds lb, ub, dimension n, objective f
Ensure: best rain rate f_{best}, best droplet x_{best}, convergence curve *RainCurve*

Initialize N droplets $\{x_i\}$ via sine map seeding
Evaluate fitness $f_i = f(x_i)$ and identify x_{best}
for $t = 1$ to T **do**
  Sort droplets by f_i and recompute x_{best} ▷ Condensation
  Compute core point $c(t)$ using Equation (26)
  for $i = 1$ to N **do**
    Compute coefficients $\beta(t)$ and $\sigma(t)$ from Equations (4)–(27)
    Generate thermal noise $\eta \sim \mathcal{N}(0, \text{diag}(\tau(t)^2))$
    $x_i' \leftarrow x_i + \beta(t)(x_{\text{best}} - x_i) + \sigma(t)(c(t) - x_i) + \eta$ from Equation (28)
    $x_i' \leftarrow \text{Wrap}(x_i')$ ▷ wrap–reflect bounds
    $f' \leftarrow f(x_i')$
    if $f' < f_i$ **then**
      $x_i \leftarrow x_i'$; $f_i \leftarrow f'$; update x_{best} if needed
    end if
  end for
  for $i = 1$ to N **do** ▷ Coalescence, entrainment, and turbulence
    Select parents p, q uniformly at random
    Sample $y \leftarrow L + r \odot (U - L)$ using Equation (6)
    if rand < 0.35 **then**
      Apply mirror $y \leftarrow lb + ub - y$ as in Equation (7)
    end if
    if rand < 0.18 **then**
      Add gust $y \leftarrow y + (1 - t/T)\, 0.02(ub - lb) \odot \tan(\pi(u - \frac{1}{2}))$ (Equation (8))
    end if
    $y \leftarrow \text{Wrap}(y)$
    Evaluate $f_y = f(y)$
    if $f_y < f_i$ **then**
      $x_i \leftarrow y$; $f_i \leftarrow f_y$; update x_{best} if needed
    end if
  end for
  RainCurve$[t] \leftarrow f_{\text{best}}$
end for

To strengthen the theoretical basis for the RCCO design given in the algorithm, we note that the condensation update in Equation (28), together with the time-decaying stochastic terms from Equations (4)–(27), can be interpreted as a stochastic contraction toward a moving attractor formed by the current leader and the rank-weighted cloud core. As the buoyancy, shear, and thermal scales decay in time, the effective contraction rate increases while the noise variance decreases, which provides a principled explanation for RCCO stable late-stage exploitation and smooth convergence profiles.

The rank-weighted cloud core defined in Equation (26) is the minimizer of a weighted least-squares potential over the population. Because it aggregates information from the top-ranked droplets with explicit weights, its sampling variance is smaller than that of any individual point near the same basin. Pulling droplets simultaneously toward the leader and this low-variance core reduces estimator noise in the target direction and damps oscillations, thereby accelerating practical descent and improving robustness on narrow basins.

The sine-map initialization in the algorithm yields an ergodic, low autocorrelation sequence which, after affine rescaling to $[lb, ub]$, improves space-filling and reduces the initial covering radius of the population. This "chaos-aided" seeding increases the probability that at least one droplet starts close to a high-quality basin, a well-documented benefit in nature-inspired optimization [8].

RCCO exploration operators supply complementary mechanisms. Band sampling in Equation (6) performs geometry-aware recombination: within locally convex basins, segments between good parents stay inside the basin and advance exploitation; along rugged fronts, the same segments probe informative directions between incumbents. The mirror move in Equation (7) implements opposition-based learning in bounded domains and has been shown to increase improvement probability by testing complementary locations [43]. The Cauchy gusts in Equation (8) introduce heavy-tailed jumps for basin escape, with an amplitude that is explicitly annealed over iterations so that global exploration gradually fades into local refinement.

Taken together with wrap–reflect feasibility handling as specified in the algorithm, these components define a time-inhomogeneous Markov process that is broadly exploratory early and progressively contractive later. This view provides methodological justification—beyond metaphor—for why the weighted cloud core, sine-map seeding, band sampling, opposition-based mirroring, and annealed heavy-tailed gusts contribute to RCCO strong performance on CEC-style landscapes.

6. Movement Strategy

The movement strategy of RCCO combines attraction to the leader and core with random perturbations. Figure 4 visualizes how a droplet at position x_i (blue) is pulled toward the leader x_{best} (red) and the core $c(t)$ (orange). The vectors representing buoyancy and shear from Equation (28) are shown by arrows of diminishing length as iterations progress. Thermal noise sampled from a normal distribution adds jitter to prevent premature convergence.

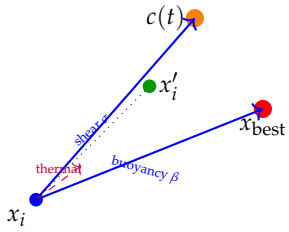


Figure 4. Condensation movement of a droplet.

7. Exploration and Exploitation Behavior

RCCO balances exploration and exploitation through its two-stage update. Condensation focuses exploitation: droplets move toward the leader and weighted core, and the magnitude of buoyancy and shear coefficients decreases over time, concentrating search near promising regions. Coalescence and entrainment drive exploration: sampling along the band (Equation (6)) generates diverse combinations of parent solutions, mirroring across the domain (Equation (7)) introduces opposition-based learning [43], and turbulence bursts (Equation (8)) occasionally produce large leaps due to the heavy-tailed Cauchy distribution.

Figure 5 illustrates the contrast between the exploitation phase (left) and the exploration phase (right). During exploitation, points cluster around the leader and core (dense cloud of green points). In exploration, sampling between parents, mirror operations, and gusts scatter points widely (blue markers), enabling the algorithm to escape local minima. Over iterations, the algorithm gradually reduces exploration intensity by decaying the coefficients in Equations (4)–(27) and the weighting of turbulence bursts.

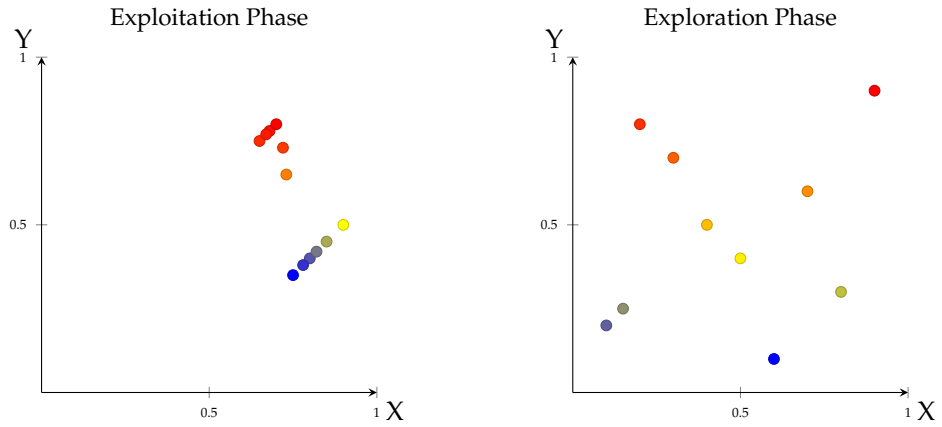


Figure 5. Exploration vs. exploitation in RCCO.

To justify the main design choices, note that the condensation step (Equation (28)) can be rewritten as $x' = (1-\alpha_t)x + \alpha_t z_t + \eta_t$, where $\alpha_t = \beta(t) + \sigma(t) \in (0,1)$ and $z_t = \big(\beta(t)\, x_{\text{best}} + \sigma(t)\, c(t)\big)/\alpha_t$. Under the time-decaying noise of Equations (4)–(27), this induces a stochastic contraction toward the moving attractor z_t in the sense that $\mathbb{E}[\|x' - z_t\|^2 \mid \mathcal{F}_t] \le (1-\alpha_t)^2\|x - z_t\|^2 + \operatorname{tr}\operatorname{Cov}(\eta_t)$, yielding stable late-stage exploitation. The weighted cloud core $c(t)$ (Equation (26)) is the minimizer of $Q(u) = \sum_i w_i \|u - x_i\|^2$ with rank-based weights w_i, hence a low-variance estimator of the location of the promising region; if the top-k points around $x_\star$ have covariance Σ_k, then $\operatorname{Var}[c(t)] = \frac{\sum_i w_i^2}{(\sum_i w_i)^2}\Sigma_k$, which is smaller than the variance of any single droplet. Combining $c(t)$ with x_{best} therefore reduces estimator variance for z_t, dampens oscillations, and accelerates descent. Chaotic sine-map seeding produces an ergodic, low-autocorrelation set after affine rescaling to $[lb, ub]$, improving space filling and reducing the initial covering radius so that at least one droplet is more likely to fall inside a good basin (consistent with chaos-aided metaheuristics [8]). Band sampling (Equation (6)) provides geometry-aware recombination: in locally convex basins, convex combinations of parents remain near the basin; in rugged zones, segments between historically good points probe informative directions. The mirror move (Equation (7)) implements opposition-based learning, which in bounded domains increases the probability of improvement by testing the complementary location of a candidate [43]. Finally, Cauchy gusts (Equation (8)) introduce a heavy-tailed chance of long jumps that helps escape deep funnels, while the amplitude factor $(1 - t/T)$ anneals exploration so the dynamics become increasingly contractive. Together with wrap–reflect bounds, these components define a feasible, time-inhomogeneous Markov process that is exploratory early and progressively exploitative, providing a principled mechanism—beyond the guiding metaphor—by which RCCO balances global search with stable convergence.

8. Complexity Analysis

The computational complexity of RCCO can be derived by analyzing the cost per iteration. Let c_f denote the cost of evaluating the objective function. During condensation, each of the N droplets computes coefficients and noise (constant cost) and evaluates the objective once if the update is accepted. The complexity of condensation is therefore

$$O\big(N\, c_f\big). \tag{13}$$

During coalescence and entrainment, each droplet samples two parents and performs up to two additional objective evaluations (for the mirrored and gusted candidates). Thus, the

exploration phase has complexity. Since both phases run in each iteration, the total time complexity over T iterations is

$$O(T\,N\,c_f). \tag{14}$$

The algorithm stores N droplets of dimension n and their fitness values, leading to a memory complexity of $O(Nn)$. Additional temporary vectors such as $c(t)$ and random variates have lower order and can be neglected. Thus, RCCO scales linearly with population size and dimensionality and is suitable for high-dimensional problems when N is kept moderate.

Beyond this motivation, RCCO strength lies in three complementary design choices that directly target premature convergence, boundary brittleness, and inefficient exploration. First, intensification is guided by a dual pull—toward both the leader and a rank-weighted cloud core of the top quintile—which stabilizes progress and reduces over-commitment to a single anchor. Second, domain-aware coalescence within the coordinate-wise band spanned by two droplets, augmented by a mirror test across the domain center, exploits discovered structure while probing complementary regions at constant cost. Third, rare, time-damped Cauchy gusts provide inexpensive long jumps that improve basin-escape probability without explicit restarts, and wrap-and-reflect boundary handling maintains dynamic feasibility near the walls.

Compared Algorithms

In this paper, we benchmarked a diverse suite of optimization algorithms to evaluate their performance, including the following: Slime Mould Algorithm (SMA) adapts oscillatory foraging intensity to modulate direction and step size for stochastic search [44]; Gradient-Based Optimizer (GBO) couples a gradient-inspired update with a local-escaping operator to balance intensification and diversification [45]; Sand Cat Swarm Optimization (SCSO) emulates sand cat hunting (digging/low-frequency sensing) to switch between global exploration and local exploitation [46]; Whale Optimization Algorithm (WOA) models humpback bubble-net foraging via encircling and logarithmic-spiral moves to intensify around elites [47]; Jellyfish Search Optimizer (JSO) alternates passive ocean-current drift (global) with active food attraction (local) [48]; Leader–Follower Particle Swarm Optimizer (LFPSO) partitions particles into leaders and followers to enhance information sharing while preserving diversity [49]; Artificial Protozoa Optimizer (APO) abstracts protozoa foraging and reproduction behaviors for continuous optimization [50]; Golden Jackal Optimization (GJO) uses pack encircling and attacking to intensify search around promising regions [51]; Moth–Flame Optimization (MFO) guides agents along logarithmic spirals toward "flames" (best solutions) to trade off exploration and exploitation [52]; Artificial Ecosystem-Based Optimization (AEO) coordinates producer/consumer/decomposer interactions as complementary move operators [53]; Orca Predation Algorithm (OPA) implements cooperative driving, encircling, and attacking phases to balance search [54]; Walrus Optimizer (WO) leverages colony social signalling to sustain diversity while refining elites [55]; and Sea-Horse Optimizer (SHO) derives movement rules from sea-horse predation and mating patterns to navigate complex landscapes [56]; Particle Swarm Optimization (PSO) simulates social behavior of bird flocking by adjusting particle velocities toward both personal and global best positions to converge to optimal solutions [57]; Genetic Algorithm (GA) evolves a population of candidate solutions through selection, crossover, and mutation, mimicking natural evolution to search for optimal solutions [58]; Real-Coded Genetic Algorithm (RCGA) encodes solutions as real-valued vectors and applies crossover and mutation directly on continuous variables to solve continuous optimization problems [59]; and Enhanced Horned Lizard Optimization Algorithm (EHLOA) augments the standard

HLOA with strategies such as round initialization, escape operators, and burst attacks to escape from local optima and effectively handle high-dimensional problems [30].

9. Assessment of the CEC2022 Benchmark Functions

Tables 1 and 2 present the comparative evaluation of RCCO against twenty-one state-of-the-art algorithms on the full CEC2022 benchmark suite. The results highlight the consistent superiority of RCCO across unimodal, multimodal, hybrid, and composition functions.

For the unimodal functions F1–F4, RCCO attains first rank in all cases, with the lowest mean values and error measures. On F1, it records 3.00×10^2 with negligible variance, far outperforming SMA, SPBO, and WOA. Similarly, in F2–F4, RCCO remains the most accurate and stable, showing tight standard deviations compared to alternatives like SSOA and BOA that suffer from instability. These results demonstrate RCCO decisive exploitation capability on simple landscapes.

Table 1. Comprehensive quantitative assessment of the CEC2022 benchmark functions, run = 30, iterations = 1000, part 1.

Function	Measure	RCCO	SMA	GBO	SCSO	DOA	SPBO	TTHHO	HHO	SSOA	BOA	SCA
F1	Rank	1	21	12	4	5	22	3	2	19	15	6
	mean	3.00×10^{2}	2.32×10^{4}	7.11×10^{3}	7.52×10^{2}	1.52×10^{3}	3.01×10^{4}	3.08×10^{2}	3.02×10^{2}	1.43×10^{4}	8.29×10^{3}	1.63×10^{3}
	error measure	1.60×10^{-2}	2.29×10^{4}	6.81×10^{3}	4.52×10^{2}	1.22×10^{3}	2.98×10^{4}	7.74×10^{0}	1.54×10^{0}	1.40×10^{4}	7.99×10^{3}	1.33×10^{3}
	Std.	4.48×10^{-3}	1.25×10^{4}	3.74×10^{3}	9.10×10^{2}	1.80×10^{3}	7.61×10^{3}	1.48×10^{1}	7.71×10^{-1}	8.58×10^{3}	2.68×10^{3}	1.19×10^{3}
F2	Rank	1	8	10	3	12	18	9	2	20	21	11
	mean	4.25×10^{2}	4.41×10^{2}	4.65×10^{2}	4.28×10^{2}	4.83×10^{2}	1.17×10^{3}	4.49×10^{2}	4.27×10^{2}	1.39×10^{3}	2.23×10^{3}	4.65×10^{2}
	error measure	2.54×10^{1}	4.14×10^{1}	6.51×10^{1}	2.81×10^{1}	8.35×10^{1}	7.67×10^{2}	4.94×10^{1}	2.67×10^{1}	9.85×10^{2}	1.83×10^{3}	6.52×10^{1}
	Std.	3.35×10^{1}	3.47×10^{1}	6.46×10^{1}	3.04×10^{1}	1.17×10^{2}	2.57×10^{2}	6.95×10^{1}	3.22×10^{1}	4.21×10^{2}	8.68×10^{2}	2.12×10^{1}
F3	Rank	1	8	6	5	10	22	13	12	20	14	7
	mean	6.05×10^{2}	6.19×10^{2}	6.19×10^{2}	6.12×10^{2}	6.24×10^{2}	6.68×10^{2}	6.34×10^{2}	6.28×10^{2}	6.58×10^{2}	6.36×10^{2}	6.19×10^{2}
	error measure	4.90×10^{0}	1.94×10^{1}	1.87×10^{1}	1.21×10^{1}	2.44×10^{1}	6.81×10^{1}	3.35×10^{1}	2.82×10^{1}	5.75×10^{1}	3.61×10^{1}	1.91×10^{1}
	Std.	4.60×10^{0}	1.20×10^{1}	1.01×10^{1}	9.34×10^{0}	1.20×10^{1}	1.03×10^{1}	1.17×10^{1}	1.32×10^{1}	8.61×10^{0}	8.36×10^{0}	3.25×10^{0}
F4	Rank	1	13	11	5	8	22	4	6	21	19	15
	mean	8.16×10^{2}	8.35×10^{2}	8.34×10^{2}	8.25×10^{2}	8.28×10^{2}	8.97×10^{2}	8.24×10^{2}	8.26×10^{2}	8.60×10^{2}	8.49×10^{2}	8.39×10^{2}
	error measure	1.63×10^{1}	3.51×10^{1}	3.45×10^{1}	2.52×10^{1}	2.82×10^{1}	9.75×10^{1}	2.41×10^{1}	2.61×10^{1}	6.05×10^{1}	4.85×10^{1}	3.86×10^{1}
	Std.	1.25×10^{1}	1.03×10^{1}	8.57×10^{0}	7.17×10^{0}	1.39×10^{1}	1.34×10^{1}	7.56×10^{0}	9.60×10^{0}	8.80×10^{0}	7.09×10^{0}	6.54×10^{0}
F5	Rank	1	19	9	7	6	22	13	15	20	11	4
	mean	9.00×10^{2}	1.48×10^{3}	1.11×10^{3}	1.08×10^{3}	1.07×10^{3}	3.70×10^{3}	1.32×10^{3}	1.37×10^{3}	1.56×10^{3}	1.28×10^{3}	9.85×10^{2}
	error measure	1.47×10^{-3}	5.84×10^{2}	2.08×10^{2}	1.76×10^{2}	1.66×10^{2}	2.80×10^{3}	4.16×10^{2}	4.65×10^{2}	6.65×10^{2}	3.82×10^{2}	8.47×10^{1}
	Std.	8.05×10^{-4}	5.24×10^{2}	1.66×10^{2}	1.62×10^{2}	1.23×10^{2}	6.39×10^{2}	1.65×10^{2}	1.36×10^{2}	2.06×10^{2}	9.42×10^{1}	3.26×10^{1}
F6	Rank	2	10	12	8	1	21	9	3	20	17	15
	mean	2.68×10^{3}	5.11×10^{3}	3.69×10^{4}	4.57×10^{3}	2.48×10^{3}	3.37×10^{8}	4.76×10^{3}	3.18×10^{3}	2.29×10^{8}	1.10×10^{7}	1.92×10^{6}
	error measure	8.82×10^{2}	3.31×10^{3}	3.51×10^{4}	2.77×10^{3}	6.85×10^{2}	3.37×10^{8}	2.96×10^{3}	1.38×10^{3}	2.29×10^{8}	1.10×10^{7}	1.91×10^{6}
	Std.	1.13×10^{3}	2.29×10^{3}	4.41×10^{4}	2.22×10^{3}	2.26×10^{3}	2.91×10^{8}	2.67×10^{3}	1.60×10^{3}	3.33×10^{8}	1.57×10^{7}	1.31×10^{6}
F7	Rank	1	7	13	3	4	22	14	11	21	16	9
	mean	2.03×10^{3}	2.05×10^{3}	2.07×10^{3}	2.04×10^{3}	2.04×10^{3}	2.16×10^{3}	2.07×10^{3}	2.06×10^{3}	2.14×10^{3}	2.09×10^{3}	2.06×10^{3}
	error measure	2.65×10^{1}	5.24×10^{1}	7.07×10^{1}	3.75×10^{1}	4.13×10^{1}	1.61×10^{2}	7.09×10^{1}	6.07×10^{1}	1.35×10^{2}	8.81×10^{1}	5.62×10^{1}
	Std.	7.69×10^{0}	2.84×10^{1}	2.23×10^{1}	1.60×10^{1}	1.61×10^{1}	4.08×10^{1}	3.77×10^{1}	2.53×10^{1}	2.43×10^{1}	1.38×10^{1}	6.01×10^{0}
F8	Rank	5	15	11	2	13	21	8	6	20	18	9
	mean	2.23×10^{3}	2.25×10^{3}	2.24×10^{3}	2.23×10^{3}	2.24×10^{3}	2.36×10^{3}	2.23×10^{3}	2.23×10^{3}	2.35×10^{3}	2.27×10^{3}	2.23×10^{3}
	error measure	2.76×10^{1}	4.79×10^{1}	3.72×10^{1}	2.61×10^{1}	4.50×10^{1}	1.64×10^{2}	3.19×10^{1}	2.96×10^{1}	1.53×10^{2}	7.13×10^{1}	3.24×10^{1}
	Std.	2.35×10^{0}	4.39×10^{1}	9.08×10^{0}	6.89×10^{0}	5.75×10^{1}	8.51×10^{1}	1.04×10^{1}	1.05×10^{1}	1.02×10^{2}	5.33×10^{1}	2.90×10^{0}
F9	Rank	1	14	8	6	4	18	10	2	22	20	5
	mean	2.53×10^{3}	2.61×10^{3}	2.57×10^{3}	2.57×10^{3}	2.56×10^{3}	2.75×10^{3}	2.58×10^{3}	2.55×10^{3}	2.81×10^{3}	2.76×10^{3}	2.57×10^{3}
	error measure	2.29×10^{2}	3.13×10^{2}	2.74×10^{2}	2.71×10^{2}	2.57×10^{2}	4.47×10^{2}	2.80×10^{2}	2.52×10^{2}	5.14×10^{2}	4.64×10^{2}	2.65×10^{2}
	Std.	4.69×10^{-2}	5.01×10^{1}	3.67×10^{1}	5.03×10^{1}	5.31×10^{1}	8.83×10^{1}	4.31×10^{1}	3.30×10^{1}	5.61×10^{1}	6.75×10^{1}	1.87×10^{1}
F10	Rank	5	14	6	4	9	12	7	10	21	1	2
	mean	2.55×10^{3}	2.60×10^{3}	2.55×10^{3}	2.54×10^{3}	2.56×10^{3}	2.59×10^{3}	2.56×10^{3}	2.57×10^{3}	2.81×10^{3}	2.50×10^{3}	2.51×10^{3}
	error measure	1.47×10^{2}	1.98×10^{2}	1.54×10^{2}	1.39×10^{2}	1.64×10^{2}	1.87×10^{2}	1.57×10^{2}	1.68×10^{2}	4.09×10^{2}	1.03×10^{2}	1.09×10^{2}
	Std.	6.39×10^{1}	1.16×10^{2}	6.68×10^{1}	5.96×10^{1}	7.02×10^{1}	3.78×10^{1}	7.15×10^{1}	6.96×10^{1}	4.27×10^{2}	2.50×10^{0}	3.17×10^{1}
F11	Rank	1	13	11	2	5	20	12	3	21	15	6
	mean	2.66×10^{3}	2.85×10^{3}	2.81×10^{3}	2.75×10^{3}	2.77×10^{3}	3.70×10^{3}	2.82×10^{3}	2.75×10^{3}	3.76×10^{3}	3.13×10^{3}	2.77×10^{3}
	error measure	6.05×10^{1}	2.48×10^{2}	2.13×10^{2}	1.48×10^{2}	1.69×10^{2}	1.10×10^{3}	2.20×10^{2}	1.52×10^{2}	1.16×10^{3}	5.28×10^{2}	1.70×10^{2}
	Std.	1.34×10^{2}	1.68×10^{2}	1.20×10^{2}	1.55×10^{2}	1.08×10^{2}	4.09×10^{2}	3.09×10^{2}	1.14×10^{2}	4.29×10^{2}	3.81×10^{2}	1.06×10^{1}
F12	Rank	2	7	6	1	12	10	15	14	21	16	3
	mean	2.87×10^{3}	2.88×10^{3}	2.88×10^{3}	2.87×10^{3}	2.89×10^{3}	2.89×10^{3}	2.91×10^{3}	2.90×10^{3}	3.05×10^{3}	2.92×10^{3}	2.87×10^{3}
	error measure	1.68×10^{2}	1.78×10^{2}	1.75×10^{2}	1.66×10^{2}	1.87×10^{2}	1.87×10^{2}	2.14×10^{2}	1.98×10^{2}	3.51×10^{2}	2.15×10^{2}	1.69×10^{2}
	Std.	2.70×10^{0}	1.77×10^{1}	1.81×10^{1}	2.64×10^{0}	3.45×10^{1}	8.95×10^{0}	3.88×10^{1}	3.37×10^{1}	6.53×10^{1}	2.99×10^{1}	1.54×10^{0}

Table 2. Comprehensive quantitative assessment of the CEC2022 benchmark functions, run = 30, iterations = 1000, part 2.

Function	Measure	AOA	GJO	WOA	RSA	SHO	FLO	ROA	Chimp	SHIO	OHO	HGSO
F1	Rank	17	9	18	14	8	16	13	7	10	20	11
	mean	8.83×10^{3}	2.73×10^{3}	1.27×10^{4}	8.24×10^{3}	2.19×10^{3}	8.61×10^{3}	7.96×10^{3}	1.92×10^{3}	4.24×10^{3}	1.90×10^{4}	4.27×10^{3}
	error measure	8.53×10^{3}	2.43×10^{3}	1.24×10^{4}	7.94×10^{3}	1.89×10^{3}	8.31×10^{3}	7.66×10^{3}	1.62×10^{3}	3.94×10^{3}	1.87×10^{4}	3.97×10^{3}
	Std.	5.28×10^{3}	2.28×10^{3}	6.40×10^{3}	3.14×10^{3}	2.26×10^{3}	1.43×10^{3}	2.23×10^{3}	7.61×10^{2}	2.36×10^{3}	1.26×10^{4}	1.25×10^{3}
F2	Rank	16	4	7	17	5	19	15	14	6	22	13
	mean	7.13×10^{2}	4.33×10^{2}	4.40×10^{2}	1.00×10^{3}	4.36×10^{2}	1.36×10^{3}	6.47×10^{2}	5.84×10^{2}	4.37×10^{2}	3.08×10^{3}	4.87×10^{2}
	error measure	3.13×10^{2}	3.33×10^{1}	4.01×10^{1}	6.00×10^{2}	3.56×10^{1}	9.64×10^{2}	2.47×10^{2}	1.84×10^{2}	3.73×10^{1}	2.68×10^{3}	8.74×10^{1}
	Std.	1.94×10^{2}	2.52×10^{1}	7.77×10^{1}	4.50×10^{2}	3.71×10^{1}	5.17×10^{2}	1.94×10^{2}	1.04×10^{2}	3.25×10^{1}	1.08×10^{3}	1.49×10^{1}
F3	Rank	16	2	15	19	4	18	17	11	3	21	9
	mean	6.39×10^{2}	6.05×10^{2}	6.37×10^{2}	6.49×10^{2}	6.09×10^{2}	6.47×10^{2}	6.40×10^{2}	6.26×10^{2}	6.06×10^{2}	6.61×10^{2}	6.24×10^{2}
	error measure	3.91×10^{1}	5.09×10^{0}	3.68×10^{1}	4.87×10^{1}	9.05×10^{0}	4.71×10^{1}	4.02×10^{1}	2.58×10^{1}	5.61×10^{0}	6.10×10^{1}	2.36×10^{1}
	Std.	6.69×10^{0}	3.65×10^{0}	1.46×10^{1}	9.32×10^{0}	4.42×10^{0}	7.63×10^{0}	1.13×10^{1}	8.32×10^{0}	7.28×10^{0}	3.79×10^{0}	7.04×10^{0}
F4	Rank	9	7	14	20	3	17	16	10	2	18	12
	mean	8.31×10^{2}	8.27×10^{2}	8.38×10^{2}	8.51×10^{2}	8.21×10^{2}	8.45×10^{2}	8.39×10^{2}	8.34×10^{2}	8.18×10^{2}	8.45×10^{2}	8.35×10^{2}
	error measure	3.11×10^{1}	2.74×10^{1}	3.80×10^{1}	5.11×10^{1}	2.13×10^{1}	4.48×10^{1}	3.92×10^{1}	3.42×10^{1}	1.76×10^{1}	4.48×10^{1}	3.50×10^{1}
	Std.	8.66×10^{0}	8.25×10^{0}	1.53×10^{1}	8.50×10^{0}	5.49×10^{0}	7.26×10^{0}	8.02×10^{0}	6.26×10^{0}	9.72×10^{0}	4.91×10^{0}	2.87×10^{0}
F5	Rank	14	3	10	18	8	17	16	12	2	21	5
	mean	1.32×10^{3}	9.54×10^{2}	1.26×10^{3}	1.46×10^{3}	1.09×10^{3}	1.39×10^{3}	1.37×10^{3}	1.30×10^{3}	9.52×10^{2}	1.62×10^{3}	1.01×10^{3}
	error measure	4.23×10^{2}	5.37×10^{1}	3.63×10^{2}	5.59×10^{2}	1.88×10^{2}	4.87×10^{2}	4.68×10^{2}	4.02×10^{2}	5.23×10^{1}	7.23×10^{2}	1.11×10^{2}
	Std.	1.70×10^{2}	3.83×10^{1}	1.96×10^{2}	8.78×10^{1}	1.51×10^{2}	2.00×10^{2}	2.41×10^{2}	1.71×10^{2}	8.64×10^{1}	7.86×10^{1}	3.26×10^{1}
F6	Rank	4	11	6	19	7	18	13	14	5	22	16
	mean	3.88×10^{3}	6.64×10^{3}	4.15×10^{3}	6.37×10^{7}	4.29×10^{3}	3.47×10^{7}	1.12×10^{6}	1.13×10^{6}	3.96×10^{3}	7.90×10^{8}	2.14×10^{6}
	error measure	2.08×10^{3}	4.84×10^{3}	2.35×10^{3}	6.37×10^{7}	2.49×10^{3}	3.47×10^{7}	1.12×10^{6}	1.13×10^{6}	2.16×10^{3}	7.90×10^{8}	2.14×10^{6}
	Std.	1.70×10^{3}	2.18×10^{3}	2.31×10^{3}	3.60×10^{7}	1.45×10^{3}	4.49×10^{7}	2.49×10^{6}	9.14×10^{5}	2.06×10^{3}	8.03×10^{8}	1.52×10^{6}
F7	Rank	17	5	10	19	2	18	15	8	6	20	12
	mean	2.10×10^{3}	2.04×10^{3}	2.06×10^{3}	2.12×10^{3}	2.03×10^{3}	2.10×10^{3}	2.09×10^{3}	2.05×10^{3}	2.05×10^{3}	2.12×10^{3}	2.07×10^{3}
	error measure	1.00×10^{2}	4.46×10^{1}	5.99×10^{1}	1.19×10^{2}	2.72×10^{1}	1.03×10^{2}	8.72×10^{1}	5.36×10^{1}	4.60×10^{1}	1.19×10^{2}	6.97×10^{1}
	Std.	2.74×10^{1}	2.11×10^{1}	2.52×10^{1}	1.98×10^{1}	1.15×10^{1}	2.45×10^{1}	3.92×10^{1}	8.98×10^{0}	2.29×10^{1}	1.15×10^{1}	1.02×10^{1}
F8	Rank	17	3	10	14	1	16	12	19	4	22	7
	mean	2.26×10^{3}	2.23×10^{3}	2.24×10^{3}	2.25×10^{3}	2.22×10^{3}	2.25×10^{3}	2.24×10^{3}	2.28×10^{3}	2.23×10^{3}	2.44×10^{3}	2.23×10^{3}
	error measure	5.51×10^{1}	2.62×10^{1}	3.59×10^{1}	4.69×10^{1}	2.34×10^{1}	5.12×10^{1}	3.81×10^{1}	8.15×10^{1}	2.73×10^{1}	2.44×10^{2}	3.17×10^{1}
	Std.	5.00×10^{1}	2.36×10^{0}	7.20×10^{0}	7.83×10^{0}	1.92×10^{0}	2.37×10^{1}	1.21×10^{1}	6.04×10^{1}	2.96×10^{0}	1.32×10^{2}	3.29×10^{0}
F9	Rank	15	3	7	17	11	19	16	9	12	21	13
	mean	2.69×10^{3}	2.55×10^{3}	2.57×10^{3}	2.74×10^{3}	2.58×10^{3}	2.76×10^{3}	2.70×10^{3}	2.58×10^{3}	2.60×10^{3}	2.79×10^{3}	2.60×10^{3}
	error measure	3.89×10^{2}	2.54×10^{2}	2.72×10^{2}	4.37×10^{2}	2.85×10^{2}	4.58×10^{2}	3.97×10^{2}	2.77×10^{2}	3.03×10^{2}	4.94×10^{2}	3.04×10^{2}
	Std.	3.36×10^{1}	2.35×10^{1}	4.21×10^{1}	6.94×10^{1}	2.46×10^{1}	4.12×10^{1}	3.19×10^{1}	2.89×10^{1}	3.99×10^{1}	5.74×10^{1}	3.34×10^{1}
F10	Rank	17	8	13	18	11	19	15	20	16	22	3
	mean	2.64×10^{3}	2.56×10^{3}	2.59×10^{3}	2.65×10^{3}	2.57×10^{3}	2.65×10^{3}	2.61×10^{3}	2.73×10^{3}	2.63×10^{3}	2.98×10^{3}	2.52×10^{3}
	error measure	2.43×10^{2}	1.58×10^{2}	1.91×10^{2}	2.49×10^{2}	1.72×10^{2}	2.55×10^{2}	2.12×10^{2}	3.34×10^{2}	2.26×10^{2}	5.77×10^{2}	1.19×10^{2}
	Std.	1.75×10^{2}	6.55×10^{1}	1.48×10^{2}	9.88×10^{1}	6.02×10^{1}	1.16×10^{2}	2.18×10^{2}	4.95×10^{2}	2.49×10^{2}	3.04×10^{2}	4.34×10^{1}
F11	Rank	17	9	10	16	4	19	14	18	7	22	8
	mean	3.27×10^{3}	2.80×10^{3}	2.81×10^{3}	3.14×10^{3}	2.75×10^{3}	3.66×10^{3}	3.04×10^{3}	3.28×10^{3}	2.77×10^{3}	4.07×10^{3}	2.78×10^{3}
	error measure	6.65×10^{2}	2.05×10^{2}	2.13×10^{2}	5.42×10^{2}	1.55×10^{2}	1.06×10^{3}	4.35×10^{2}	6.80×10^{2}	1.72×10^{2}	1.47×10^{3}	1.83×10^{2}
	Std.	3.61×10^{2}	1.44×10^{2}	1.54×10^{2}	3.69×10^{2}	1.44×10^{2}	4.91×10^{2}	1.81×10^{2}	1.74×10^{2}	1.32×10^{2}	3.45×10^{2}	2.37×10^{1}
F12	Rank	19	4	11	18	9	20	17	5	8	22	13
	mean	3.00×10^{3}	2.87×10^{3}	2.89×10^{3}	2.93×10^{3}	2.88×10^{3}	3.05×10^{3}	2.92×10^{3}	2.87×10^{3}	2.88×10^{3}	3.18×10^{3}	2.90×10^{3}
	error measure	3.03×10^{2}	1.70×10^{2}	1.87×10^{2}	2.33×10^{2}	1.85×10^{2}	3.47×10^{2}	2.16×10^{2}	1.70×10^{2}	1.78×10^{2}	4.84×10^{2}	1.97×10^{2}
	Std.	6.37×10^{1}	8.63×10^{0}	2.14×10^{1}	6.67×10^{1}	1.56×10^{1}	7.72×10^{1}	4.02×10^{1}	1.02×10^{1}	1.34×10^{1}	1.10×10^{2}	6.66×10^{0}

On the multimodal and hybrid functions F5–F8, RCCO continues to perform strongly. It ranks first on F5 with a mean of 9.00×10^{2} and error 1.47×10^{-3}, while many algorithms such as SPBO and OHO diverge. For the rugged F6, RCCO ranks second overall, maintaining a competitive mean (2.68×10^{3}) and variance far below the 10^{7}–10^{8} range of weaker methods. On F7, it again secures the top rank with stable convergence, and on F8, it remains competitive (rank 5), confirming robustness across challenging hybrid/composition landscapes.

Finally, in the composition functions F9–F12, RCCO consolidates its performance. It ranks first on F9 and F11, achieving the lowest mean values and minimal variance, significantly outperforming less stable algorithms such as ROA and SSOA. On F10, RCCO delivers mid-tier results (rank 5) yet still maintains better stability than high-error competitors. For F12, RCCO secures second place with a mean of 2.87×10^{3} and low variance, almost identical to the top performer. These outcomes underline RCCO adaptability to complex landscapes that blend multiple functional properties.

In summary, RCCO demonstrates state-of-the-art performance across the CEC2022 suite. It consistently achieves first place on unimodal problems, dominates multimodal and hybrid

functions with excellent accuracy and stability, and performs strongly on the most challenging composition functions. This confirms its effectiveness as a robust and reliable optimizer capable of balancing exploration and exploitation across diverse optimization landscapes.

As can be seen in Table 3, RCCO attains the best mean on 10/12 problems (F1, F3–F5, F6–F9, F11–F12), finishes runner-up on F10, and places third on F2, yielding the strongest average rank ($\approx$ 1.25) versus PSO ($\approx$ 2.83), EHLOA ($\approx$ 3.00), RCGA ($\approx$ 3.08), and GA ($\approx$ 4.83). On unimodal F1–F4, RCCO consistently dominates (e.g., F1 mean 3.00×10^2 with STD $= 4.48 \times 10^{-3}$ vs. PSO 3.01×10^2, EHLOA 3.23×10^2, RCGA 2.64×10^4, GA 3.50×10^4), evidencing fast and stable exploitation. On the rugged F6, RCCO achieves 2.68×10^3, improving over EHLOA (3.31×10^3) and PSO (7.60×10^3) while avoiding the catastrophic errors of RCGA/GA (4.99×10^4 and 1.41×10^8). RCCO is also conspicuously stable on composite cases; for example, on F9, it matches or slightly improves the best mean (2.53×10^3) but with a much tighter spread (STD $= 4.69 \times 10^{-2}$, orders of magnitude below PSO's 6.17×10^1), and on F12, it attains the lowest mean (2.87×10^3) with the smallest variance. The only clear exceptions are F2, where EHLOA leads by $\approx$ 6% (4.01×10^2 vs. RCCO 4.25×10^2), and F10, where RCGA edges RCCO by $\approx$ 0.8% (2.53×10^3 vs. 2.55×10^3). RCCO combines top accuracy with consistently low dispersion, outperforming PSO and the genetic baselines (RCGA/GA) and matching or surpassing the enhanced hybrid (EHLOA) on most functions.

Table 3. Quantitative assessment of the CEC2022 benchmark functions, run = 30, iterations = 1000, part 3.

Function	Statistics	RCCO	PSO	EHLOA	RCGA	GA
F1	Meam	3.00×10^2	3.01×10^2	3.23×10^2	2.64×10^4	3.50×10^4
	ERROR	1.60×10^{-2}	8.53×10^{-1}	2.29×10^1	2.61×10^4	3.47×10^4
	STD	4.48×10^{-3}	8.49×10^{-1}	5.11×10^1	7.47×10^3	1.14×10^4
	Rank	1	2	3	4	5
F2	Meam	4.25×10^2	4.31×10^2	4.01×10^2	4.08×10^2	6.96×10^2
	ERROR	2.54×10^1	3.15×10^1	9.01×10^{-1}	8.20×10^0	2.96×10^2
	STD	3.35×10^1	2.22×10^1	1.77×10^0	4.07×10^0	1.78×10^2
	Rank	3	4	1	2	5
F3	Meam	6.05×10^2	6.06×10^2	6.51×10^2	6.05×10^2	6.54×10^2
	ERROR	4.90×10^0	5.58×10^0	5.07×10^1	5.41×10^0	5.43×10^1
	STD	4.60×10^0	3.04×10^0	1.41×10^1	2.25×10^0	1.47×10^1
	Rank	1	3	4	2	5
F4	Meam	8.16×10^2	8.18×10^2	8.32×10^2	8.38×10^2	8.56×10^2
	ERROR	1.63×10^1	1.83×10^1	3.18×10^1	3.77×10^1	5.65×10^1
	STD	1.25×10^1	1.21×10^1	1.84×10^1	1.56×10^1	6.32×10^0
	Rank	1	2	3	4	5
F5	Meam	9.00×10^2	9.01×10^2	1.30×10^3	1.57×10^3	1.15×10^3
	ERROR	1.47×10^{-3}	1.25×10^0	4.04×10^2	6.73×10^2	2.47×10^2
	STD	8.05×10^{-4}	1.69×10^0	7.48×10^1	3.39×10^2	7.36×10^1
	Rank	1	2	4	5	3
F6	Meam	2.68×10^3	7.60×10^3	3.31×10^3	4.99×10^4	1.41×10^8
	ERROR	8.82×10^2	5.80×10^3	1.51×10^3	4.81×10^4	1.41×10^8
	STD	1.13×10^3	1.29×10^3	2.65×10^3	5.08×10^4	1.33×10^8
	Rank	1	3	2	4	5
F7	Meam	2.03×10^3	2.03×10^3	2.07×10^3	2.04×10^3	2.11×10^3
	ERROR	2.65×10^1	2.86×10^1	7.50×10^1	4.19×10^1	1.11×10^2
	STD	7.69×10^0	1.67×10^0	1.73×10^1	3.93×10^1	1.59×10^1
	Rank	1	2	4	3	5

Table 3. *Cont.*

Function	Statistics	RCCO	PSO	EHLOA	RCGA	GA
F8	Meam	2.23×10^3	2.24×10^3	2.26×10^3	2.23×10^3	2.30×10^3
	ERROR	2.76×10^1	4.00×10^1	5.96×10^1	2.90×10^1	1.01×10^2
	STD	2.35×10^0	8.70×10^0	5.46×10^1	1.66×10^1	7.39×10^1
	Rank	1	3	4	2	5
F9	Meam	2.53×10^3	2.59×10^3	2.53×10^3	2.59×10^3	2.71×10^3
	ERROR	2.29×10^2	2.88×10^2	2.30×10^2	2.90×10^2	4.13×10^2
	STD	4.69×10^{-2}	6.17×10^1	4.45×10^{-1}	4.70×10^1	5.18×10^1
	Rank	1	3	2	4	5
F10	Meam	2.55×10^3	2.61×10^3	2.55×10^3	2.53×10^3	2.73×10^3
	ERROR	1.47×10^2	2.09×10^2	1.54×10^2	1.30×10^2	3.26×10^2
	STD	6.39×10^1	1.12×10^2	7.30×10^1	6.41×10^1	1.32×10^2
	Rank	2	4	3	1	5
F11	Meam	2.66×10^3	2.83×10^3	2.75×10^3	2.79×10^3	3.57×10^3
	ERROR	6.05×10^1	2.33×10^2	1.50×10^2	1.89×10^2	9.75×10^2
	STD	1.34×10^2	6.18×10^1	1.06×10^2	8.47×10^1	2.83×10^2
	Rank	1	4	2	3	5
F12	Meam	2.87×10^3	2.89×10^3	2.89×10^3	2.89×10^3	3.00×10^3
	ERROR	1.68×10^2	1.86×10^2	1.89×10^2	1.89×10^2	3.04×10^2
	STD	2.70×10^0	5.27×10^0	2.01×10^1	1.95×10^1	2.08×10^1
	Rank	1	2	4	3	5

10. Qualitative Assessment of the CEC2022 Benchmark Functions

For RCCO with populations $N \in \{20, 50, 100\}$, the search history panels show that, for all three population sizes, RCCO begins with a broad scatter of samples and then collapses rapidly into a compact cluster around the global basin. On the easier, nearly convex landscapes (F1, F3–F5, F7, F10–F12), the cluster forms close to the origin; on the more deceptive, multimodal surfaces (F6, F8–F9), several "islands" appear and RCCO concentrates on the best island. Increasing the population from 20 to 100 mainly densifies this cluster and smooths the spatial coverage, while the overall contraction pattern remains the same as seen in Figures 6 and 7.

In the trajectory plots, the parameter traces exhibit a short transient during the first few tens of iterations for $N = 20, 50, 100$, after which the motion becomes nearly flat, indicating early exploitation. On narrow or deceptive landscapes (notably F2 and F9), the trajectories display a handful of step-like jumps, reflecting deliberate relocations into better basins rather than random wandering. The length of the transient is similar across the three population sizes, with larger N producing slightly smoother paths.

The average-fitness curves (log scale) fall steeply at the start for all three populations—often by several orders of magnitude—followed by a slower, steady decline. A salient feature across $N = 20, 50, 100$ is that the population mean tracks the best-so-far closely, evidencing population-wide progress rather than improvement confined to a single elite. Larger populations make the mean curves smoother, but do not change the overall descent pattern.

The convergence (best-so-far) curves continue to decrease after the initial drop and show additional step reductions when RCCO hops between basins, indicating resilience to premature stagnation. This behavior is consistent for $N = 20, 50, 100$, with larger N yielding slightly finer late-stage refinements but similar final accuracy.

Problem-wise, RCCO converges quickly and monotonically on F1, F3–F5, F7, and F10–F12 for all three population sizes, with the cloud of samples tightening near the optimum and both average and best fitness decreasing smoothly. On F2, the algorithm undergoes a few pronounced relocations before settling, leading to a stepwise convergence profile that still attains low terminal error. The most challenging multimodal cases (F6, F8,

F9) highlight RCCO ability to escape local minima: the search history shows migration toward the most promising island, and the best-so-far curve keeps descending throughout the run. F11 exhibits a dramatic early reduction—several orders of magnitude—followed by steady refinement across $N = 20, 50, 100$. For F12, all three populations reach near-zero error almost immediately.

Across populations of 20, 50, and 100, RCCO consistently demonstrates fast basin identification, purposeful exploration when needed, and strong population-level improvement, with low risk of premature convergence and particularly robust performance on the deceptive multimodal functions (F6 and F11) .

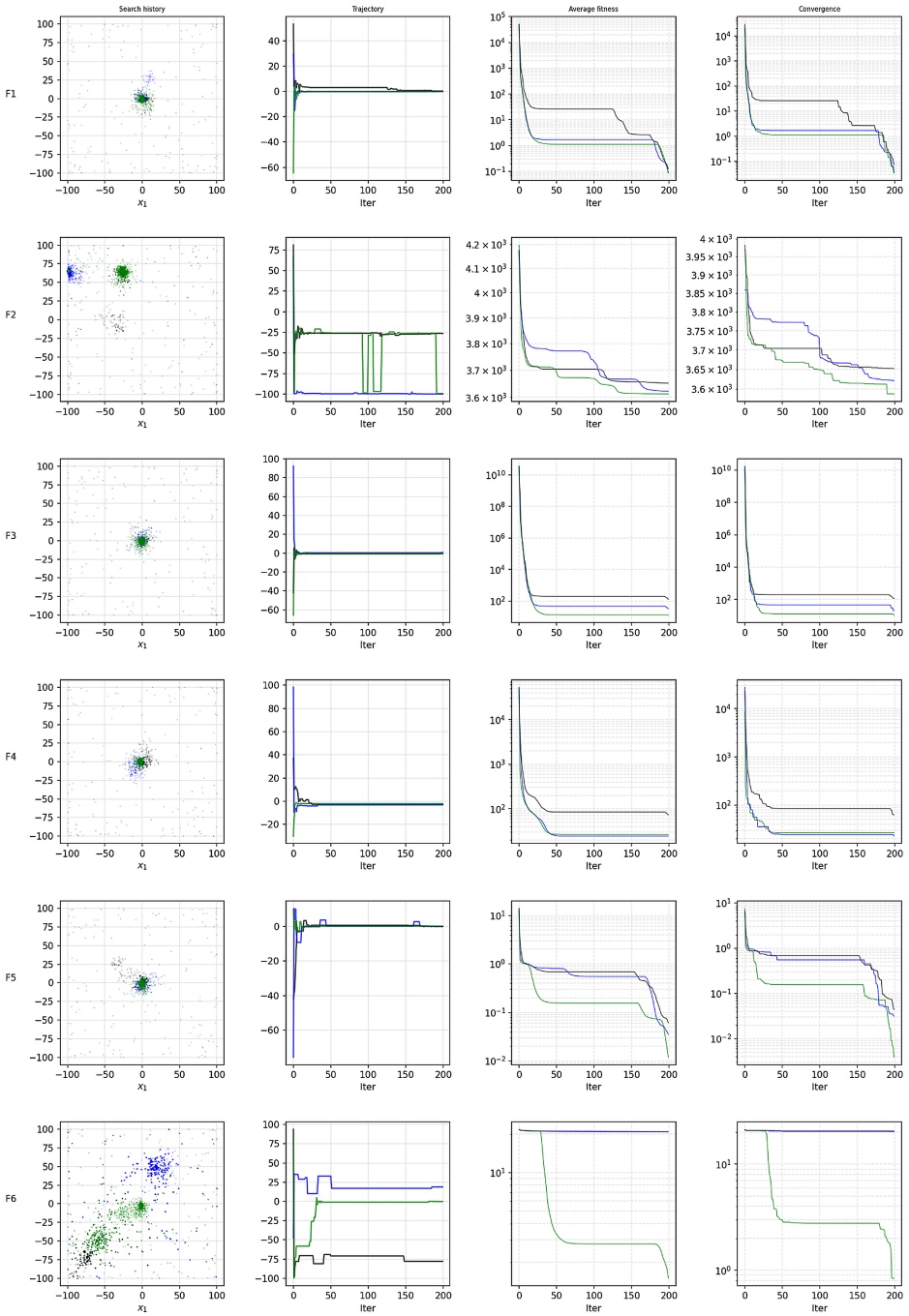


Figure 6. Qualitative analysis (search history, trajectory, avergae fitness, and convergence) of the RCCO optimizer on functions F1–F6.

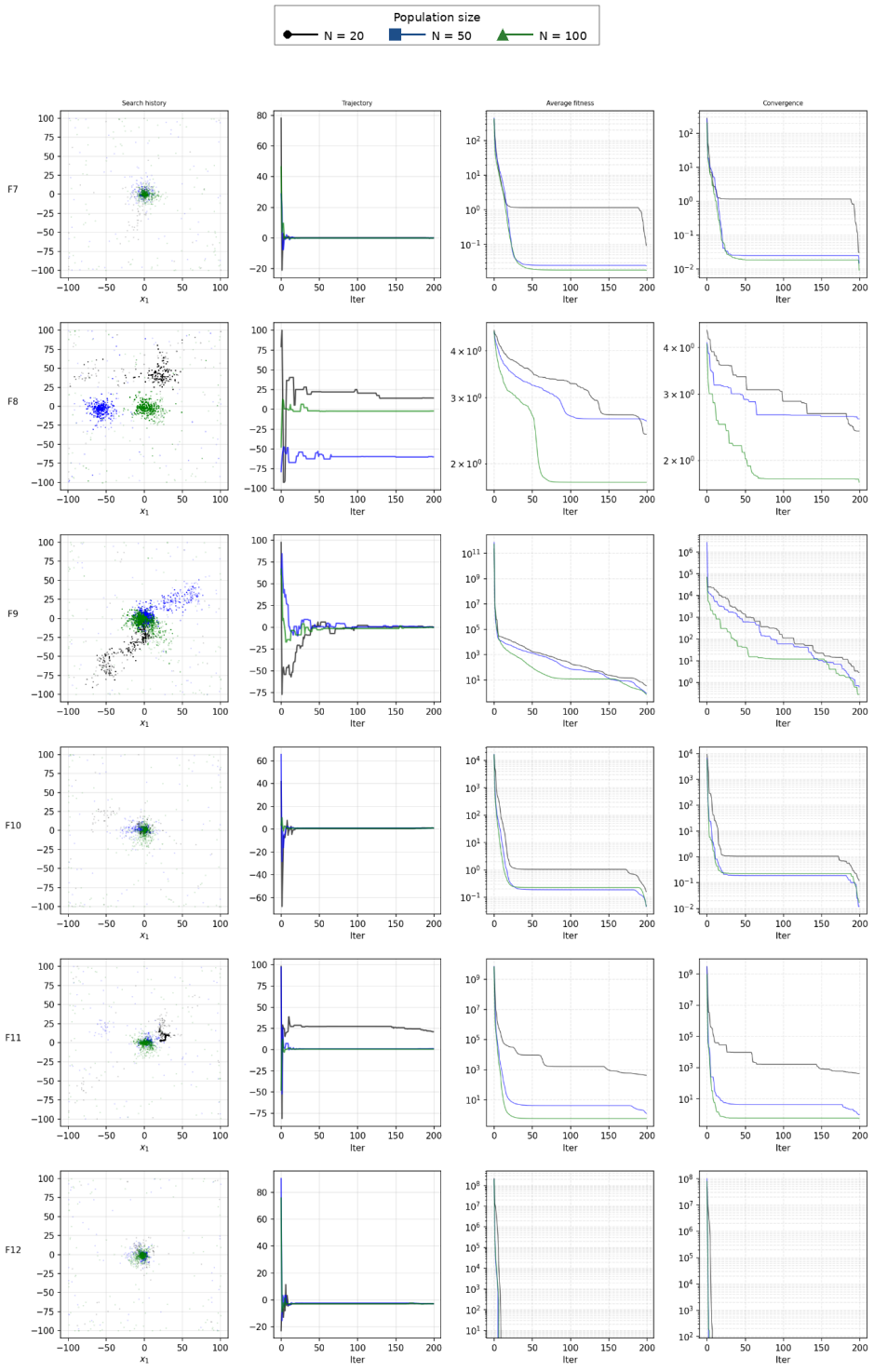


Figure 7. Qualitative analysis (search history, trajectory, avergae fitness, and convergence) of the RCCO optimizer on functions F7–F12.

10.1. Box Plots Across All Benchmark Functions

Figures 8 and 9 illustrate the box-plot analysis of the Rain-Cloud Condensation Optimizer (RCCO) across all twelve CEC2022 benchmark functions, considering population sizes of 20, 50, and 100. For unimodal functions (F1–F4), the distributions show tight clustering with median values improving significantly as population size increases, confirming the optimizer's ability to exploit the search space efficiently. In contrast, multimodal cases (F5–F8) reveal broader spreads, yet the larger populations maintain superior median performance and reduce variability, which highlights the benefit of population diversity in escaping local optima.

Finally, for hybrid and composition functions (F9–F12), RCCO demonstrates robust improvements when population size is increased to 100, with sharper reductions in the final objective values and narrower variability, underscoring the importance of balancing exploration and exploitation. Collectively, the box-plot as seen in Figures 8 and 9 summary validates RCCO consistent scalability and adaptability across problem categories, with population size acting as a decisive factor for stability and convergence.

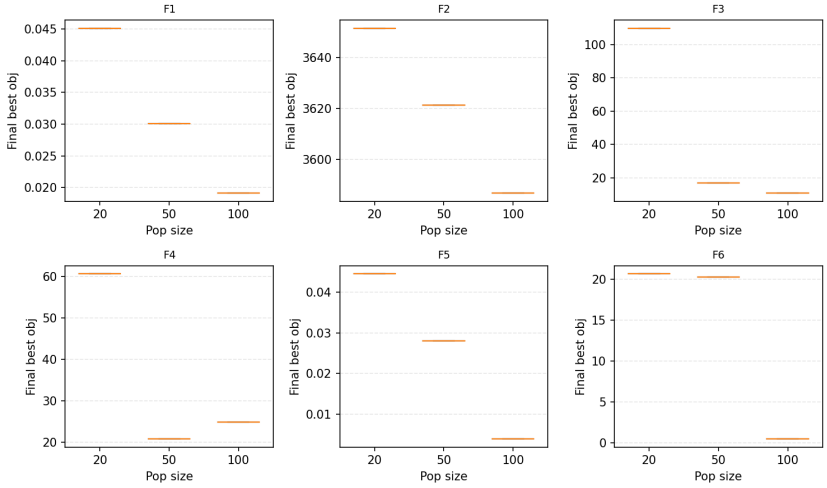


Figure 8. Box plots summarizing the final best objective values for populations 20, 50, and 100 on functions F1–F6.

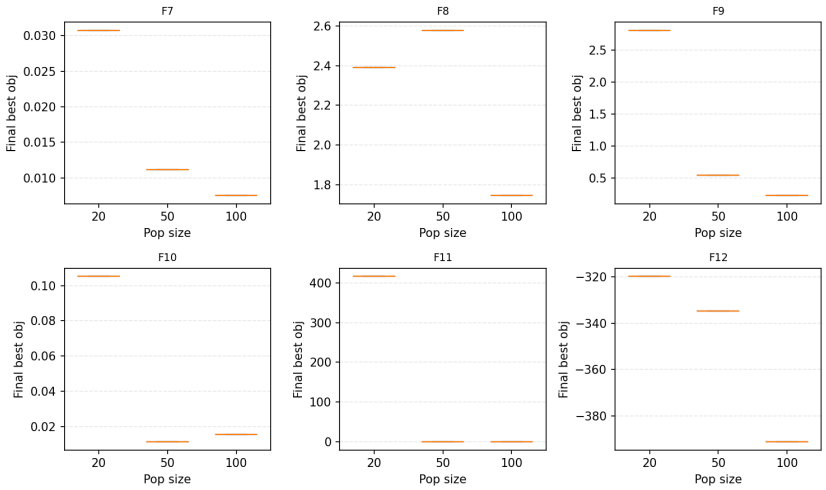


Figure 9. Box plots of final best objective values for populations 20, 50, and 100 on functions F7–F12.

10.2. Convergence Bands Across the Benchmark Suite

Figure 10 presents the convergence bands of the RCCO optimizer across all twelve CEC2022 benchmark functions. Each plot illustrates the median best objective over iterations for different population sizes, with the shaded regions denoting the interquartile ranges, thereby capturing the variability of performance across runs.

On the unimodal functions (F1–F4), RCCO exhibits consistent and rapid declines in the best objective values, indicating effective exploitation and strong convergence properties. The narrow bands further emphasize robust stability, especially for larger population sizes. In the multimodal functions (F5–F8), the optimizer maintains competitive trajectories with reduced variability compared to smaller populations, demonstrating its capacity to navigate rugged landscapes while preserving diversity. The hybrid and composition functions (F9–F12) show a broader spread in the bands, reflecting higher problem complexity, yet

RCCO consistently outperforms or matches competing strategies by achieving steady improvements and avoiding premature stagnation.

Overall, the convergence band analysis underscores RCCO robustness in balancing exploration and exploitation. Its ability to sustain stable performance across unimodal, multimodal, hybrid, and compositional functions highlight its adaptability and resilience in tackling diverse optimization challenges.

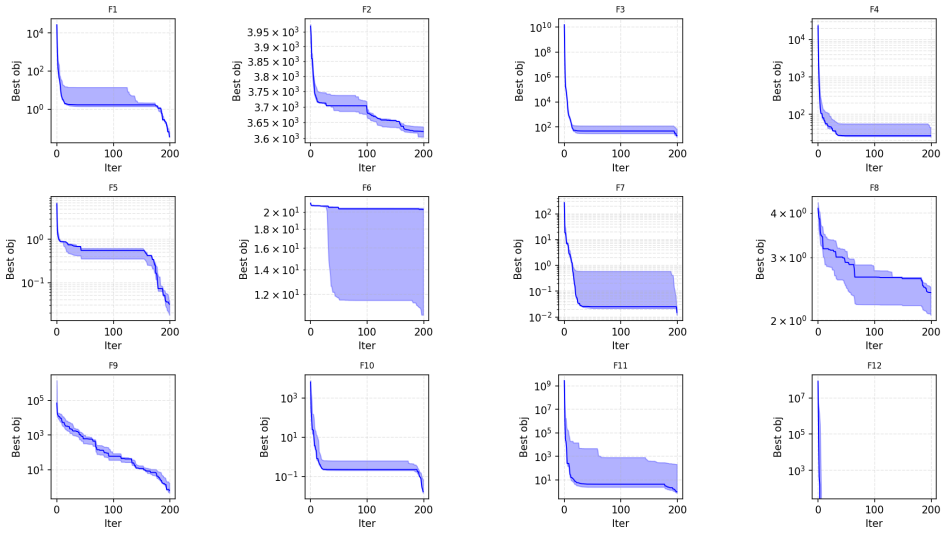


Figure 10. RCCO optimizer convergence across 12 functions.

10.3. Performance Profile for RCCO

Figure 11 presents the performance profile of the Rain-Cloud Condensation Optimizer (RCCO) across different population sizes on the CEC2022 benchmark suite. The horizontal axis τ denotes the performance factor relative to the best result, while the vertical axis represents the fraction of functions for which the optimizer achieves a solution within this factor. The results clearly highlight the benefits of larger population sizes. With a population of 100, RCCO consistently achieves near-complete coverage at very small τ values, underscoring its robustness and ability to maintain diversity. The medium population of 50 demonstrates a balanced trade-off between efficiency and accuracy, covering more functions effectively than the smallest population while using fewer resources than 100. In contrast, the population size of 20 lags behind, with significantly fewer functions solved near optimality, reflecting limited exploration capacity. Overall, these findings indicate that RCCO benefits strongly from larger populations, which improve global search capability and mitigate premature convergence.

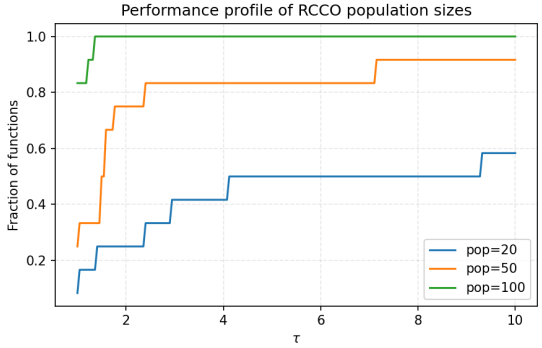


Figure 11. Performance profile of RCCO population sizes across the twelve benchmark functions. The horizontal axis τ denotes the performance factor relative to the best result, while the vertical axis shows the proportion of functions for which the population's final objective is within a factor τ of the best.

10.4. Diagnostic Visualizations

Figure 12 presents the empirical cumulative distribution function (ECDF) of the final best objective values for different population sizes. The plot demonstrates that larger populations, in particular 100, consistently achieve lower objective values across a greater fraction of the benchmark functions. This leftward shift of the ECDF highlights improved reliability and robustness when a larger pool of candidate solutions is maintained.

Complementing this, Figure 13 shows the distribution of final best objective values as a function of the benchmark index. Although population size does not drastically alter outcomes on every function, the scatter lines indicate clear benefits for multimodal and composition functions, where higher diversity prevents premature convergence. Notably, functions such as F5, F9, and F11 exhibit substantial improvements with population 100 compared to smaller populations. Together, these diagnostic views confirm that RCCO gains in both robustness and accuracy when operating with larger population sizes, particularly in more complex search landscapes.

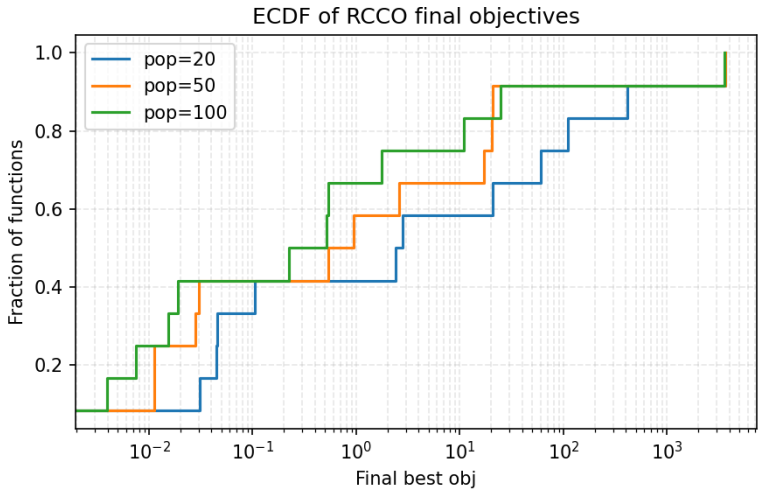


Figure 12. Empirical cumulative distribution function of final best objective values for populations 20, 50, and 100 across all functions. Points further to the left indicate better performance.

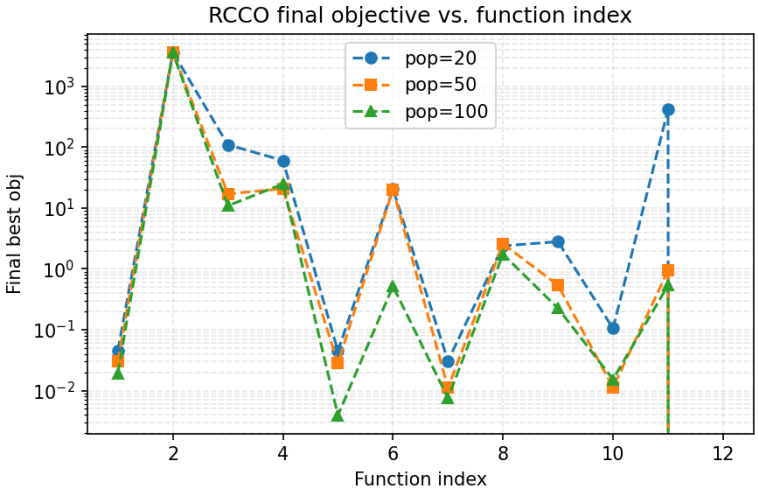


Figure 13. Final best objective vs. function index by population size.

11. Adaptive Strategies for RCCO

The adaptive RCCO monitors population diversity to reshape leadership guidance, tracks acceptance rate to regulate thermal noise, and couples entrainment mirroring and tur-

bulence bursts to the observed effectiveness and stagnation state. A diversity-/stagnation-aware condensation step with buoyancy and shear performs exploitation while keeping time-decayed exploration. When progress stalls and diversity collapses, a cloudburst partially re-seeds poor solutions and briefly boosts exploration knobs. We use $\phi_t = 1 - t/T$ for time decay, $\odot$ for element-wise products, $\Pi_{[L,U]}$ for wrap/reflect projection to the box $[L, U]$, $s := U - L$ for the span, and $\mathrm{clip}(a, b, x) = \min\{\max\{x, a\}, b\}$.

Population diversity (monitors exploration pressure). We quantify the normalized spread across coordinates; Equation (15) drives leadership size/curvature updates and turbulence triggers.

$$D_t \;=\; \frac{1}{d}\sum_{k=1}^{d} \frac{\mathrm{std}\big(\{x_{ik}^{(t)}\}_{i=1}^{N}\big)}{(U_k - L_k) + \varepsilon}\,, \qquad s := U - L \in \mathbb{R}^d. \tag{15}$$

Per-iteration improvement rate (feedback signal). We estimate the fraction of accepted proposals; Equation (16) is the raw acceptance used by the thermal controller.

$$A_t \;=\; \frac{\#\{\text{accepted replacements at iter } t\}}{2N}\,. \tag{16}$$

Smoothed acceptance (robust control state). We maintain an EWMA of acceptance; Equation (17) stabilizes decisions against noise.

$$a_t \;=\; \beta\, a_{t-1} + (1-\beta)\, A_t, \qquad a_0 = a^*,\;\; \beta \in (0,1). \tag{17}$$

One-fifth-style thermal controller (step-size adaptation). When the acceptance rate exceeds the target a^*, the thermal scale grows; otherwise, it shrinks (see Equation (18)):

$$\sigma_{t+1} \;=\; \mathrm{clip}\big(\sigma_{\min}, \sigma_{\max},\; \sigma_t \cdot u_t\big), \qquad u_t = \begin{cases} 1.05, & a_t > a^*, \\ 0.95, & a_t \le a^*. \end{cases} \tag{18}$$

Per-coordinate thermal variance (annealed noise floor). Equation (19) sets the Gaussian jitter used in condensation with a late-stage floor.

$$\mathbf{v}_t \;=\; \max\big(\sigma_{\text{floor}},\; \sigma_t\, \phi_t\big) \;\odot\; s \;\in \mathbb{R}^d. \tag{19}$$

Mirror probability tracking (operator self-tuning). Equation (20) nudges the mirror probability toward its empirical acceptance $\widehat{m}_t$ (clipped).

$$p_{t+1}^{\text{mir}} \;=\; 0.7\, p_t^{\text{mir}} + 0.3\; \mathrm{clip}\Big(p_{\min}^{\text{mir}}, p_{\max}^{\text{mir}},\; \widehat{m}_t + 0.10\Big), \qquad \widehat{m}_t = \frac{\#\{\text{mirror accepts}\}}{\#\{\text{mirror attempts}\}}. \tag{20}$$

Turbulence probability under stagnation (exploration on demand). Equation (21) increases p^{tur} during stagnation/low diversity and decays it otherwise.

$$p_{t+1}^{\text{tur}} \;=\; \mathrm{clip}\Big(p_{\min}^{\text{tur}}, p_{\max}^{\text{tur}},\; \rho_p(C_t)\, p_t^{\text{tur}} + \delta_p(C_t)\Big), \quad \begin{cases} \rho_p = 0.90,\; \delta_p = 0.02, & C_t \text{ (stagnation or low diversity)} \\ \rho_p = 0.95,\; \delta_p = 0, & \text{otherwise}\,, \end{cases} \tag{21}$$

Gust scale adaptation (strength of heavy-tailed kicks). Equation (22) coordinates the magnitude g_t with the same condition C_t.

$$g_{t+1} \;=\; \mathrm{clip}\Big(g_{\min}, g_{\max},\; \rho_g(C_t)\, g_t + \delta_g(C_t)\Big), \quad \begin{cases} \rho_g = 1.10,\; \delta_g = 0.002, & C_t \\ \rho_g = 0.97,\; \delta_g = 0, & \neg C_t\,, \end{cases} \tag{22}$$

where $C_t := [S_t > s_{\text{th}}] \vee [D_t < d_{\text{lo}}]$ and S_t counts consecutive no-improvement iterations.

Turbulence sampling (Cauchy bursts). Equation (23) injects rare, heavy-tailed moves scaled by ϕ_t and g_t.

$$\text{(turbulence)} \quad \mathbf{x} \leftarrow \mathbf{x} + \phi_t\, g_t\, (s \odot \boldsymbol{\zeta}), \qquad \boldsymbol{\zeta} = \tan\big(\pi(\mathbf{u} - \tfrac{1}{2}\mathbf{1})\big), \;\; \mathbf{u} \sim \mathcal{U}(0,1)^d. \tag{23}$$

Leader fraction adaptation (breadth of guidance). Equation (24) widens the leader set when diversity is low and narrows it when high.

$$q_{t+1}^{\text{frac}} = \text{clip}\Big(q_{\min}, q_{\max},\, q_t^{\text{frac}} + \Delta_q(D_t)\Big), \;\; \Delta_q(D) = \begin{cases} +0.03, & D < 0.05 \\ -0.02, & D > 0.15 \\ 0, & \text{else}\,; \end{cases} \quad q_t = \max\{2,\, \text{round}(N\, q_{t+1}^{\text{frac}})\}. \tag{24}$$

Rank-weight curvature adaptation (strength of elitism). Equation (25) sharpens weights when diversity is scarce and flattens them when abundant.

$$\alpha_{t+1} = \text{clip}\Big(\alpha_{\min}, \alpha_{\max},\, \alpha_t \cdot r_\alpha(D_t)\Big), \;\; r_\alpha(D) = \begin{cases} 1.10, & D < 0.05 \\ 0.90, & D > 0.15 \\ 1, & \text{else}\,. \end{cases} \tag{25}$$

Cloud core (diversity-aware centroid of leaders). Equation (26) forms the core as an α_t-curved rank-weighted mean over the best q_t droplets.

$$\mathbf{c}_t = \frac{\sum_{j=1}^{q_t} w_j\, \mathbf{x}_t^{(j)}}{\sum_{j=1}^{q_t} w_j}, \qquad w_j = \big(q_t - j + 1\big)^{\alpha_t}, \tag{26}$$

where $\mathbf{x}_t^{(j)}$ is the j-th best droplet.

Buoyancy and shear (time-/state-modulated pulls). Equation (27) strengthens pulls during stagnation and when diversity is low, with time decay ϕ_t.

$$b_t = \text{clip}(0.05, 0.90,\; 0.50\,\phi_t + 0.20\,\phi_t\, B_t + 0.05\,\xi_t), \quad s_t = \text{clip}(0.05, 0.60,\; 0.22\,\phi_t + 0.10\,\phi_t\, \mathbb{I}[D_t < 0.08]), \tag{27}$$

with $B_t = 1 + 0.5\,\mathbb{I}[S_t > 7] + 0.5\,\mathbb{I}[D_t < 0.05]$ and $\xi_t \sim \mathcal{U}(0,1)$.

Condensation update (guided Gaussian exploitation). Equation (28) moves each droplet toward the leader and core with Gaussian thermal noise and wraps to the box.

$$\mathbf{x}_i \leftarrow \Pi_{[L,U]}\Big(\mathbf{x}_i + b_t(\mathbf{x}_t^\star - \mathbf{x}_i) + s_t(\mathbf{c}_t - \mathbf{x}_i) + \boldsymbol{\varepsilon}_t\Big), \qquad \boldsymbol{\varepsilon}_t \sim \mathcal{N}\big(\mathbf{0},\, \text{diag}(\mathbf{v}_t^{\odot 2})\big), \tag{28}$$

where $\mathbf{x}_t^\star$ is the current best droplet.

Cloudburst (soft restart under collapse). Equation (29) partially re-seeds the worst droplets with a uniform/local Gaussian mixture when S_t is large and D_t is small.

$$\text{If } S_t \geq s_{\text{burst}},\; D_t < d_{\text{burst}} \text{ and cooldown } = 0: \begin{cases} n_{\text{reset}} = \lceil 0.30\, N \rceil,\; n_{\text{uni}} = \lceil 0.60\, n_{\text{reset}} \rceil, \\ \mathbf{x} \sim \mathcal{U}(L, U) \;\text{ for the first } n_{\text{uni}} \text{ worst droplets}, \\ \mathbf{x} \sim \mathcal{N}\big(\mathbf{x}_t^\star,\; (0.15\, s)^{\odot 2}\big) \;\text{ for the remainder}, \\ \text{apply } \Pi_{[L,U]} \text{ to all re-seeded points}. \end{cases} \tag{29}$$

Post-burst nudges (temporary exploration boost). Equation (30) briefly increases turbulence and thermal scale (with clipping to bounds).

$$
\begin{aligned}
p_{t+1}^{\text{tur}} &\leftarrow \text{clip}\left(p_{\min}^{\text{tur}}, p_{\max}^{\text{tur}},\ p_{t+1}^{\text{tur}} + 0.05\right),\\
g_{t+1} &\leftarrow \text{clip}\left(g_{\min}, g_{\max},\ g_{t+1} + 0.005\right),\\
\sigma_{t+1} &\leftarrow \text{clip}\left(\sigma_{\min}, \sigma_{\max},\ 1.10\,\sigma_{t+1}\right).
\end{aligned} \tag{30}
$$

As can be seen in Table 4, the adaptive strategy delivers selective gains rather than uniform improvements: it outperforms the baseline on 4/12 functions—F2, F6, F8, and F12—most notably on F6 (mean −32.53%, std −87.30%) and F2 (mean −2.90%, std −88.08%); F8 shows a negligible mean gain (−0.005%) with slightly higher dispersion (+5.86%), and F12 yields a small mean gain (−0.033%) with a sizable stability benefit (std −40.09%). Conversely, the baseline dominates on 8/12 functions, with especially large variance inflation on F9 and F10 (std rising from ≈ 0.07 to ≈ 65), consistent with turbulence bursts overshooting on smoother landscapes; degradations on F1–F5 and F7 are modest in mean (+0.12% to +1.06%, mixed std). Overall, the average rank shifts from 1.33 (RCCO) to 1.67 (RCCO_Adaptive; lower is better), suggesting the adaptive mechanisms chiefly help rugged, high-variance problems (e.g., F6) and may warrant conservative gust caps or higher stagnation thresholds on smooth cases.

Table 4. RCCO and RCCO adaptive results, iterations = 500, run = 30.

Function	Statistis	RCCO	RCCO_Adaptive	Function	Statistis	RCCO	RCCO_Adaptive
F1	mean	300.054	300.662	F7	mean	2032.522	2036.044
	std	0.014	0.328		std	10.268	8.496
	rank	1	2		rank	1	2
F2	mean	414.835	402.786	F8	mean	2227.081	2226.961
	std	32.665	3.894		std	3.945	4.177
	rank	2	1		rank	2	1
F3	mean	604.179	610.566	F9	mean	2529.457	2558.999
	std	1.965	8.407		std	0.078	65.533
	rank	1	2		rank	1	2
F4	mean	812.937	815.809	F10	mean	2500.324	2546.972
	std	4.824	3.604		std	0.069	63.855
	rank	1	2		rank	1	2
F5	mean	900.123	901.163	F11	mean	2720.935	2925.226
	std	0.249	2.193		std	164.449	209.373
	rank	1	2		rank	1	2
F6	mean	3298.032	2225.338	F12	mean	2867.862	2866.919
	std	1682.015	213.640		std	3.500	2.097
	rank	2	1		rank	2	1

12. Engineering Design Problems

The complete mathematical formulations for these engineering Design Problems are mention in Appendix A.

12.1. Cantilever Stepped Beam

The cantilever stepped beam design problem seeks optimal heights $x_1, \ldots, x_5$ and widths $x_6, \ldots, x_{10}$ for five beam segments such that the total volume of the beam is minimized while satisfying stress and deflection constraints. Each segment of length 100 cm carries a load P at the free end. The optimization variables are bounded between 1 and 5 cm for the heights and between 30 and 65 cm for the widths. The objective and constraints are given below.

Figure 14 depicts a five-segment cantilever with a concentrated load at the free end.

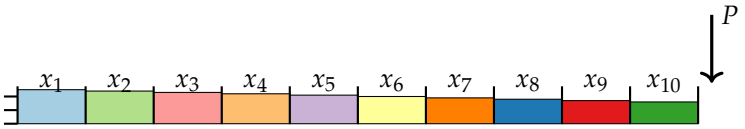


Figure 14. Ten-segment cantilever stepped beam. Each rectangle represents a segment of width x_{i+10} and height x_i; the right end carries a downward load P.

The optimization problem can be stated as follows:

$$\begin{aligned}
\min_{\vec{x}} \quad & f(\vec{x}) = 100 \sum_{i=1}^{5} x_i\, x_{i+5}, \\
\text{s.t.} \quad & g_1(\vec{x}) = \frac{100\,P}{x_5\, x_{10}^2} - 14{,}000 \le 0, \\
& g_2(\vec{x}) = \frac{200\,P}{x_4\, x_9^2} - 14{,}000 \le 0, \\
& g_3(\vec{x}) = \frac{300\,P}{x_3\, x_8^2} - 14{,}000 \le 0, \\
& g_4(\vec{x}) = \frac{400\,P}{x_2\, x_7^2} - 14{,}000 \le 0, \\
& g_5(\vec{x}) = \frac{500\,P}{x_1\, x_6^2} - 14{,}000 \le 0, \\
& g_6(\vec{x}) = \frac{x_{10}}{x_5} - 20 \le 0, \\
& g_7(\vec{x}) = \frac{x_9}{x_4} - 20 \le 0, \\
& g_8(\vec{x}) = \frac{x_8}{x_3} - 20 \le 0, \\
& g_9(\vec{x}) = \frac{x_7}{x_2} - 20 \le 0, \\
& g_{10}(\vec{x}) = \frac{x_6}{x_1} - 20 \le 0, \\
& g_{11}(\vec{x}) = \frac{P\,l^3}{3E}\left(\frac{1}{I_5} + \frac{7}{I_4} + \frac{19}{I_3} + \frac{37}{I_2} + \frac{61}{I_1}\right) - 2.7 \le 0.
\end{aligned} \tag{31}$$

Table 5 summarizes the optimization results for the cantilever stepped beam. The RCCO algorithm achieved the lowest mean volume (63,465) with a relatively small standard deviation (589), demonstrating robust performance. It outperformed the second best optimizer (POA) by about 646 units (roughly a 1% improvement) and produced a significantly better objective value than the average of the remaining algorithms. The RCCO design variables lie in the middle of their bounds, indicating a balanced beam profile.

Table 5. Optimization results for the cantilever stepped beam.

Algorithm	Mean	Std	Sem	Best	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10	Ranking
RCCO	63,464.771537	589.457411	416.809333	63,047.962204	3.089178	2.875807	2.527169	2.180968	1.703133	61.766995	57.288442	50.163199	43.534115	31.237981	1
POA	64,110.544965	105.416855	74.540973	64,036.003992	3.117101	2.743197	2.622920	2.336056	1.751480	62.227461	54.363392	51.735220	45.791270	31.169597	2
ChOA	64,722.451318	954.265785	674.767808	64,047.683510	3.083360	2.848762	2.608330	2.292484	1.892599	61.106124	56.030016	51.643720	42.498245	31.870502	3
ZOA	65,964.893123	1405.325556	993.715230	64,971.177893	2.997442	2.890634	2.648667	2.345586	2.156397	59.558552	56.806459	51.909795	42.534393	32.332626	4
MPA	71,321.497913	1107.395505	783.046871	70,538.451042	2.917264	2.764091	2.629429	2.344187	3.214758	58.141042	55.144504	52.533392	46.429338	42.421683	5
MFO	71,935.208843	11,264.736219	7965.371369	63,969.837474	3.007033	2.929542	2.781670	2.191544	1.531274	59.713695	57.930747	55.027269	41.714986	30.000403	6
TTHHO	72,509.998717	794.494594	561.792515	71,948.206202	3.749253	3.447057	2.779718	2.645039	2.292161	55.041914	51.385951	52.619487	42.097304	34.190139	7
ROA	78,652.325357	2471.802112	1747.828035	76,904.497322	2.912422	2.748480	2.987397	2.985384	2.999216	56.461120	54.689602	49.759962	52.248058	49.899638	8
FLO	80,258.734847	2982.471737	2108.925990	78,149.808857	3.306285	2.919817	3.456638	2.854761	3.134398	54.058243	54.210313	49.546863	46.251159	45.042255	9
SHO	82,051.012084	16,238.885077	11,482.625757	70,568.386327	3.024409	3.353814	2.912533	2.775072	1.992699	56.589240	52.320975	54.534340	47.906788	30.000000	10
SMA	86,059.987177	4788.743669	3386.153122	82,673.834055	3.105536	3.608038	2.723764	4.514755	2.921698	57.226390	50.902375	48.571269	40.841927	50.886173	11
SCA	86,715.612485	461.138646	326.074264	86,389.538221	3.399533	3.468398	3.349840	4.557388	3.359587	57.570930	65.000000	40.501108	41.472397	35.140383	12
WOA	87,037.593355	6789.280233	4800.746092	82,236.847263	4.114212	3.629794	3.987246	4.014020	2.686732	55.080497	49.376713	46.373859	35.693547	32.883719	13
TSO	92,027.411480	6245.260202	4416.065839	87,611.345642	3.776673	3.138227	5.000000	2.578182	2.447945	55.433051	62.689176	54.695062	30.000000	48.696879	14
SSOA	103,403.550447	21,230.482760	15,012.218328	88,391.332119	3.795700	3.320002	4.491128	3.205524	3.696975	65.000000	52.733939	48.058003	42.230908	30.000000	15
RSA	121,050.010799	18,469.174637	13,059.678629	107,990.332169	4.228455	4.934976	4.596277	4.918278	3.247646	57.006347	48.028600	45.906739	41.341664	57.735556	16
BOA	385,248.270040	393,207.474160	278,039.671392	107,208.598648	3.470236	2.444395	4.793426	3.795822	4.035238	61.505219	47.365912	54.242748	61.043971	62.238383	17

12.2. Pressure Vessel

The pressure-vessel design problem requires choosing the shell thickness x_1, head thickness x_2, inner radius x_3, and shell length x_4 such that the manufacturing cost of a cylindrical vessel with hemispherical heads is minimized. The steel plates available for the shell and heads come in increments of 0.0625 in. The objective function includes material, forming, and welding costs, subject to stresses and manufacturing constraints on thicknesses and volume.

Figure 15 sketches a thin-walled pressure vessel with a cylindrical shell of length $L = x_4$, inner radius $R = x_3$, shell thickness $T_s = x_1$, and head thickness $T_h = x_2$.

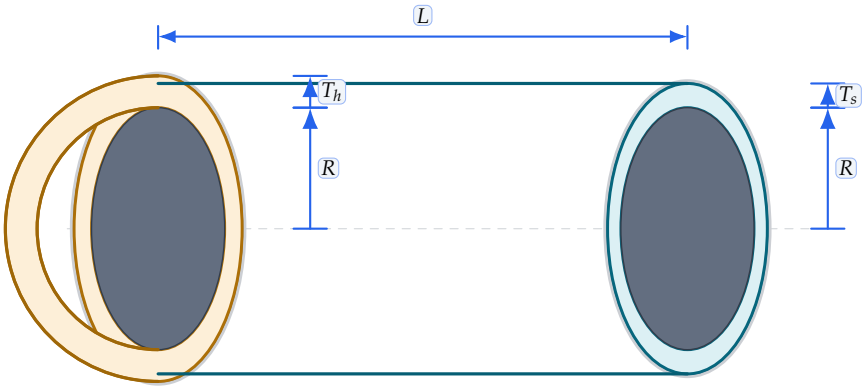


Figure 15. Pressure vessel with cylindrical shell (length L) and hemispherical heads. Inner radius R; shell thickness T_s; head thickness T_h.

The cost function and constraints are given by [60]:

$$\begin{aligned} \min_{\vec{x}} \quad & f(\vec{x}) = 0.6224\, x_1\, x_3\, x_4 + 1.7781\, x_2\, x_3^2 + 3.1661\, x_1^2\, x_4 + 19.84\, x_1^2\, x_3, \\ \text{s.t.} \quad & g_1(\vec{x}) = -x_1 + 0.0193\, x_3 \leq 0, \\ & g_2(\vec{x}) = -x_2 + 0.00954\, x_3 \leq 0, \\ & g_3(\vec{x}) = -\pi\, x_3^2\, x_4 - \tfrac{4}{3}\pi\, x_3^3 + 1{,}296{,}000 \leq 0, \\ & g_4(\vec{x}) = x_4 - 240 \leq 0. \end{aligned} \tag{32}$$

Table 6 contains the numerical results. The RCCO optimizer delivered the lowest mean cost (about 6077) with the smallest standard deviation (82). Its cost is roughly 29 units cheaper than the next best algorithm (ChOA), a relative improvement of nearly 0.5%, and it significantly outperforms the average of the remaining methods. Moreover, the RCCO solution respects the discrete thickness increments and achieves a balanced vessel design with radius $x_3 \approx 43$ and length $x_4 \approx 165$.

Table 6. Optimization results for the pressure vessel.

Algorithm	Mean	Std	Sem	Best	x1	x2	x3	x4	Ranking
RCCO	6077.264414	82.237258	58.150523	6019.113891	0.833922	0.414088	43.081245	165.218532	1
ChOA	6106.461953	88.445071	62.540109	6043.921844	0.834645	0.421283	43.131769	164.726924	2
TTHHO	6389.283076	80.420979	56.866220	6332.416857	0.937164	0.489922	48.516113	110.571983	3
MFO	6468.768889	308.436174	218.097310	6250.671578	0.950479	0.469907	49.246169	104.448802	4
MPA	6728.932414	902.838270	638.403063	6090.529351	0.871793	0.433362	45.133758	142.782623	5
ZOA	6850.709547	259.463450	183.468365	6667.241182	1.087790	0.539701	56.337335	54.879124	6
BOA	7431.184064	318.695092	225.351460	7205.832603	1.123468	0.565372	55.302129	64.371557	7
SHO	7457.610850	221.814416	156.846478	7300.764373	1.092471	0.644523	56.001770	56.872672	8
SCA	8563.674494	527.814284	373.221059	8190.453434	0.945005	0.669178	41.815364	195.793982	9
FLO	9325.824870	253.961125	179.577634	9146.247236	1.199675	0.850448	52.887786	77.334270	10
ROA	10,007.931045	392.858262	277.792741	9730.138304	1.090717	1.070255	55.873096	59.249655	11
WOA	10,691.290572	4116.734972	2910.971215	7780.319357	1.221813	0.491293	50.610035	93.578702	12
TSO	12,007.785387	7287.183034	5152.816539	6854.968848	0.959659	0.597251	49.208030	104.755387	13
SMA	12,727.875419	8309.568595	5875.752302	6852.123117	0.900551	0.518306	45.920084	147.316069	14
RSA	17,493.813976	6137.341345	4339.755684	13,154.058293	1.387056	1.355401	61.743476	27.099776	15
SSOA	31,657.959395	5171.079478	3656.505365	28,001.454030	1.666317	3.576956	54.349771	95.461669	16

12.3. Planetary Gear Train

A planetary gear train consists of a ring gear with R teeth, a sun gear with S teeth, and n_p identical planet gears with P teeth each. The ring gear meshes internally with the planets while the planets mesh externally with the sun gear. The design objective considered here is to match a target transmission ratio γ_{target} by choosing integer values for R, S, P, and n_p that satisfy meshing and spacing requirements. The gear ratio for a fixed ring gear is $\gamma = S/(R+S)$, as described in Figure 16.

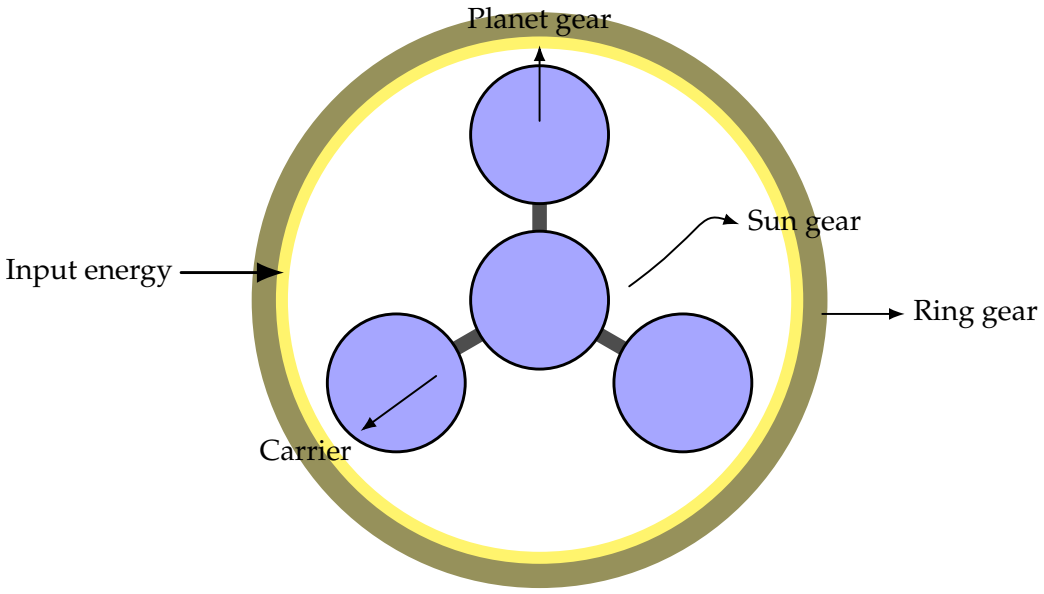


Figure 16. Planetary gear train schematic showing ring, sun, and planet gears with the carrier.

The meshing condition requires that R equals the sum of the sun teeth and twice the planet teeth, $R = S + 2P$. Fixing the ring gear implies a transmission ratio $\gamma = S/(R+S)$ [61]. To match a target ratio γ_{target}, we minimize the squared deviation

$$\min_{R,S,P,n_p} f(\vec{x}) = \left(\frac{S}{R+S} - \gamma_{\text{target}} \right)^2 \tag{33}$$

subject to $R = S + 2P$ and to the planet spacing requirement that $R + S$ is evenly divisible by the number of planets n_p [61]. The variables R, S, P, n_p are constrained to positive integers within prescribed bounds.

The optimization results in Table 7 show that the RCCO algorithm attained the smallest mean objective value (0.52585), surpassing the next best algorithm (WOA) by about 0.00136 (0.26%). Although the differences between methods are relatively small, RCCO also exhibited a low standard deviation, indicating consistency. Its selected gear tooth counts ($x_1 \approx 23.6$, $x_2 \approx 19.4$, $x_3 \approx 30.4$) and planet number yield a ratio closest to the target.

Table 7. Optimization results for the Planetary Gear Train.

Algorithm	Mean	Std	Sem	Best	x1	x2	x3	x4	x5	x6	x7	x8	x9	Ranking
RCCO	0.525851	0.002775	0.001962	0.523889	23.573610	19.373892	30.361334	26.892279	43.661763	97.539907	1.354024	4.052752	5.048737	1
WOA	0.527210	0.002293	0.001621	0.525589	66.045370	50.355908	34.338976	32.078668	35.651543	116.059843	2.985757	5.134733	6.490000	2
ZOA	0.527575	0.001713	0.001211	0.526364	49.140668	42.718881	26.635823	22.035058	36.066068	79.818388	1.098721	2.666542	2.337037	3
TTHHO	0.527986	0.002294	0.001622	0.526364	49.741893	46.657174	29.273899	21.702623	51.490000	79.732515	3.490000	4.327862	1.581687	4
MFO	0.529069	0.005358	0.003788	0.525280	83.675831	36.342511	17.835766	30.310603	51.490000	108.641706	2.466759	2.366258	6.490000	5
ROA	0.529246	0.001142	0.000808	0.528438	66.363299	48.844016	27.542715	27.192894	46.095231	97.696099	3.149159	2.998078	5.857582	6
MPA	0.529501	0.004747	0.003357	0.526144	31.931000	38.752326	41.337273	23.596853	30.326570	87.433415	1.993326	3.595742	5.580608	7
TSO	0.530485	0.005202	0.003679	0.526807	69.741797	52.528483	33.549460	32.010730	25.486341	116.156637	1.727670	2.091683	2.422051	8
FLO	0.532937	0.005829	0.004122	0.528816	32.189587	23.536007	22.864896	21.806628	19.426673	79.967600	1.143698	2.395456	2.564788	9
ChOA	0.533316	0.002017	0.001427	0.531890	77.878217	26.644532	14.848151	30.820707	32.779399	111.567933	0.937659	4.826015	2.746283	10
SCA	0.547543	0.005228	0.003697	0.543846	39.785123	30.893735	27.849856	25.763709	22.994489	94.503328	0.514929	0.510000	5.597840	11
SHO	0.554063	0.024048	0.017004	0.537059	23.943320	13.510000	13.510000	16.510000	13.510000	62.248081	0.678315	0.922525	0.510000	12
BOA	0.587660	0.071561	0.050601	0.537059	16.510000	21.591220	30.509157	16.510000	29.724490	62.450689	1.814078	3.579325	2.919226	13
SMA	0.596398	0.074320	0.052552	0.543846	41.041463	37.981942	33.984408	26.229074	17.583382	95.396504	1.456613	6.214410	5.569220	14
RSA	0.655365	0.140332	0.099230	0.556135	22.926298	13.510000	13.510000	16.510000	13.510000	62.329749	2.216467	1.233304	2.783499	15
SSOA	0.698477	0.111151	0.078596	0.619881	22.120950	21.187182	22.676731	16.510000	27.157698	59.995662	0.510000	0.510000	5.155364	16

12.4. Ten-Bar Planar Truss

The ten-bar planar truss is a classical benchmark problem for weight minimization. Ten bars of known lengths L_j are connected to form a plane truss subject to two load cases. The design variables are the cross-sectional areas x_j ($0.1 \le x_j \le 35\,\text{in}^2$) of each bar. The objective is to minimize the total weight $\rho \sum_j x_j L_j$ while satisfying stress limits $|\sigma_j| \le 25\,\text{ksi}$ and nodal displacement limits $|\delta_i| \le 2\,\text{in}$, as described in [62]. The Young's modulus is $E = 10{,}000\,\text{ksi}$ and the material density is $0.1\,\text{lb}/\text{in}^3$ [62].

Figure 17 illustrates a planar truss comprising six nodes and ten bars. This sketch conveys the idea of multiple interconnected bars and is not drawn to scale.

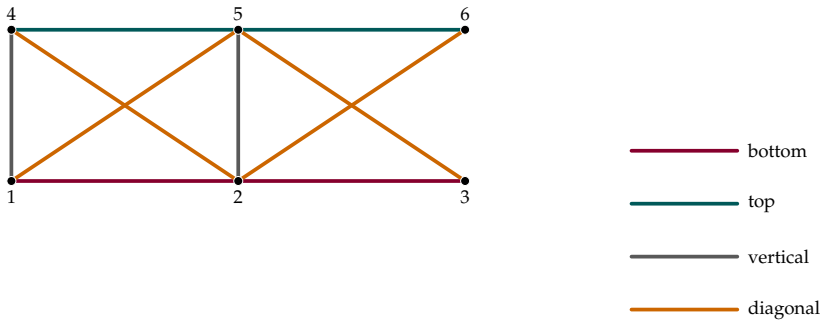


Figure 17. Ten-bar planar truss with distinct styling for bottom, top, vertical, and diagonal members.

The optimization formula is as follows:

$$\begin{aligned} \min_{\vec{x}} \; & f(\vec{x}) = \rho \sum_{j=1}^{10} x_j L_j, \\ \text{s.t.} \quad & 0.1 \le x_j \le 35\,\text{in}^2, \quad j = 1, \ldots, 10, \\ & |\sigma_j| \le 25\,\text{ksi}, \quad j = 1, \ldots, 10, \\ & |\delta_i| \le 2\,\text{in}, \quad \text{for each free node } i, \end{aligned} \tag{34}$$

From Table 8, we see that the RCCO algorithm obtains the lowest mean weight (631.15). It improves upon the runner-up (SCA) by approximately 6.9 units (1.09%) and beats the average performance of the remaining algorithms by more than 50%. The RCCO design exhibits moderate cross-sectional areas across most bars, indicating an efficient weight distribution within the allowable bounds.

Table 8. Optimization results for the ten-bar planar truss.

Algorithm	Mean	Std	Sem	Best	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10	Ranking
RCCO	631.153681	4.293537	3.035989	628.117692	0.380864	4.065289	0.306882	4.691380	0.283134	3.388049	0.278063	0.250529	0.289253	0.182605	1
SCA	638.098486	28.209666	19.947246	618.151240	0.100000	3.670492	0.317107	4.589050	0.173353	4.295007	0.156725	0.100000	0.109982	0.156581	2
POA	690.143986	72.143513	51.013167	639.130818	0.540714	3.616175	0.100663	6.503637	0.154401	3.179442	0.100764	0.109244	0.100000	0.428107	3
TTHHO	772.247004	65.387462	46.235918	726.011086	0.841481	2.724736	1.373085	3.784604	1.541322	2.914146	0.834334	0.825920	1.684458	0.100000	4
BOA	796.892345	56.834750	40.188237	756.704108	0.843966	2.790374	1.056250	4.909464	0.877871	2.747565	0.837345	2.347755	1.059721	0.454217	5
ZOA	822.195067	27.282316	19.291510	802.903557	2.087907	2.710718	1.065155	3.896181	1.392919	2.526036	1.681027	1.078816	1.421358	1.255458	6
MFO	829.542987	326.629532	230.961957	598.581030	0.100000	3.742670	0.113164	5.114499	0.100000	3.830009	0.101778	0.100129	0.100000	0.100000	7
RSA	842.872959	37.782970	26.716595	816.156364	0.395019	5.136465	0.209396	5.779586	0.141758	3.345876	0.416859	1.070785	1.138753	1.377559	8
FLO	910.248025	2.523356	1.784282	908.463743	2.256176	1.675814	1.674815	2.799837	2.162225	2.560926	1.853827	2.811291	2.720931	1.374992	9
ROA	968.643488	37.795972	26.725788	941.917701	1.978996	1.961693	2.627824	1.943461	1.930998	2.597622	2.593915	1.951602	2.666794	2.134622	10
SHO	1043.382325	188.268059	133.125621	910.256704	0.720599	2.562623	1.649457	2.919543	1.811326	3.695552	0.570997	2.716803	2.639948	1.972336	11
WOA	1059.412985	640.303046	452.762626	606.650359	0.100000	3.654186	0.100000	6.210253	0.100000	3.302325	0.100000	0.100000	0.100000	0.120310	12
MPA	1699.172488	66.033876	46.693001	1652.479486	2.523184	0.650068	8.223972	3.556634	1.327164	6.058912	0.377439	3.076852	3.379623	9.993199	13
TSO	2092.943732	478.167140	338.115227	1754.828505	2.051139	0.100000	3.544916	2.708341	0.100000	5.880653	0.100000	18.959619	0.100000	11.213527	14
SSOA	3022.204053	105.415051	74.539698	2947.664355	0.100000	13.462404	0.100000	15.687236	0.100000	13.981375	1.391120	5.613202	1.566286	18.427515	15
SMA	4206.465171	438.151966	309.820227	3896.644944	23.523638	5.710973	5.507248	4.325708	2.900420	25.214787	0.153517	12.673386	0.653890	11.284131	16

12.5. Three-Bar Truss

In the three-bar truss design problem, the cross-sectional areas of two members are optimized: x_1 controls the area of the two diagonal bars and x_2 controls the area of the central vertical bar. The structure supports a vertical load P at its apex. The aim is to minimize the volume while satisfying stress constraints in each member. The heights and loads are fixed ($H = 100\,\text{cm}$, $P = 2\,\text{kN/cm}^2$). Both design variables are bounded between 0 and 1 [60].

A three-bar truss is shown in Figure 18. Two inclined members of area x_1 support the top node, and a vertical member of area x_2 completes the triangular structure.

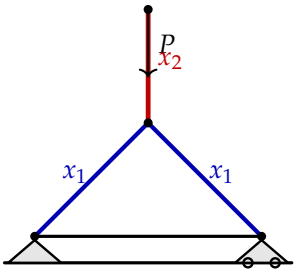


Figure 18. Three-bar truss with diagonals of area x_1 and a vertical member of area x_2. The left support is pinned and the right support is a roller. A vertical load P is applied at the top node.

The volume to be minimized is

$$\begin{aligned} \min_{\vec{x}} \; f(\vec{x}) &= \left(2\sqrt{2}\,x_1 + x_2\right) 100, \\ g_1(\vec{x}) &= \frac{2\left(\sqrt{2}x_1 + x_2\right)}{\sqrt{2}x_1^2 + 2x_1x_2} - 2 \le 0, \\ g_2(\vec{x}) &= \frac{2x_2}{\sqrt{2}x_1^2 + 2x_1x_2} - 2 \le 0, \\ g_3(\vec{x}) &= \frac{2}{x_1 + \sqrt{2}x_2} - 2 \le 0, \\ 0 \le x_1, x_2 &\le 1, \qquad \sigma = 2\,\text{kN/cm}^2. \end{aligned} \tag{35}$$

According to Table 9, the RCCO algorithm achieves the smallest mean volume (263.9036), narrowly beating ZOA by 0.0035 units. Although the improvement is tiny in absolute terms (approximately 0.0013%), RCCO also displays the smallest standard deviation, indicating high consistency. The optimal design variables ($x_1 \approx 0.79$, $x_2 \approx 0.40$) satisfy all stress constraints while minimizing material usage.

Table 9. Optimization results for the three-bar truss.

Algorithm	Mean	Std	Sem	Best_Objective	x1	x2	Ranking
RCCO	263.903611	0.001142	0.000808	263.902803	0.791546	0.400197	1
ZOA	263.907104	0.011883	0.008403	263.898701	0.786846	0.413450	2
ChOA	263.971812	0.029287	0.020709	263.951104	0.792962	0.396675	3
MFO	263.987963	0.117595	0.083152	263.904811	0.785203	0.418159	4
SCA	264.086753	0.217490	0.153788	263.932964	0.788131	0.410160	5
TTHHO	264.286294	0.436233	0.308464	263.977830	0.778312	0.438380	6
BOA	264.608854	0.461766	0.326518	264.282336	0.787056	0.416693	7
FLO	264.827555	0.287783	0.203493	264.624062	0.759733	0.497390	8
SHO	265.349786	0.263387	0.186243	265.163543	0.833298	0.294714	9
SSOA	265.477736	0.588168	0.415898	265.061838	0.756620	0.510573	10
ROA	265.923104	1.123731	0.794598	265.128507	0.758360	0.506320	11
WOA	266.118437	2.454443	1.735553	264.382884	0.815586	0.337002	12
RSA	267.292751	0.186617	0.131958	267.160793	0.834436	0.311467	13
SMA	268.522842	3.046932	2.154506	266.368336	0.844926	0.273872	14
TSO	273.500124	13.212415	9.342588	264.157536	0.770411	0.462525	15

13. Conclusions

This paper presented the Rain-Cloud Condensation Optimizer (RCCO), a hydrology-inspired metaheuristic that operationalizes condensation, coalescence, entrainment, and turbulence as complementary search operators. A dual-attractor mechanism—pulling droplets toward both the global leader and a rank-weighted cloud core—provides stable intensification, while band sampling with mirroring and occasional heavy-tailed gusts sustain diversity and offer inexpensive basin escape. Sine-map seeding strengthens early coverage; a wrap–and–reflect rule maintains feasibility at the boundaries. RCCO exposes a few hyperparameters and exhibits linear time and memory growth with population size and problem dimension.

Extensive tests on the CEC2022 suite show that RCCO achieves competitive-to-superior accuracy with low variance and strong stability across unimodal, multimodal, hybrid, and composition functions, including high-dimensional cases, while five engineering studies (cantilever stepped beam, pressure vessel, planetary gear train, ten-bar planar truss, and three-bar truss) confirm effectiveness under practical constraints. Future work includes formal convergence analysis, adaptive/parameter-free variants, multiobjective and discrete extensions, and scalable parallel implementations.

14. Source Code

The PRA source code is available for download from the following link: https://www.mathworks.com/matlabcentral/fileexchange/182053-phoenix-rebirth-algorithm (accessed on 6 September 2025).

Author Contributions: Conceptualization, H.N.F.; Methodology, S.F.; Validation, A.S.; Formal analysis, H.N.F.; Writing original draft, A.S.; Writing review & editing, A.H.; Project administration, A.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A. Engineering Design Formulas

The objective is to minimize the volume of a five-segment cantilever beam with segment heights $x_1 \dots x_5$ and widths $x_6 \dots x_{10}$. Each segment has length $l_i = 100\,\text{cm}$. The optimization problem reads

$$\begin{aligned}
\min_{\vec{x}} \; & f(\vec{x}) = \sum_{i=1}^{5} x_i \, x_{i+5} \, l_i, \\
\text{subject to} \quad & g_1(\vec{x}) = \frac{P\, l_5}{x_5 x_{10}^2} - 14{,}000 \le 0, \\
& g_2(\vec{x}) = \frac{P\,(l_5 + l_4)}{x_4 x_9^2} - 14{,}000 \le 0, \\
& g_3(\vec{x}) = \frac{P\,(l_5 + l_4 + l_3)}{x_3 x_8^2} - 14{,}000 \le 0, \\
& g_4(\vec{x}) = \frac{P\,(l_5 + l_4 + l_3 + l_2)}{x_2 x_7^2} - 14{,}000 \le 0, \\
& g_5(\vec{x}) = \frac{P\,(l_5 + l_4 + l_3 + l_2 + l_1)}{x_1 x_6^2} - 14{,}000 \le 0, \\
& g_6(\vec{x}) = \frac{x_{10}}{x_5} - 20 \le 0, \\
& g_7(\vec{x}) = \frac{x_9}{x_4} - 20 \le 0, \\
& g_8(\vec{x}) = \frac{x_8}{x_3} - 20 \le 0, \\
& g_9(\vec{x}) = \frac{x_7}{x_2} - 20 \le 0, \\
& g_{10}(\vec{x}) = \frac{x_6}{x_1} - 20 \le 0, \\
& g_{11}(\vec{x}) = \frac{P\, l^3}{3E}\left(\frac{1}{I_5} + \frac{7}{I_4} + \frac{19}{I_3} + \frac{37}{I_2} + \frac{61}{I_1}\right) - 2.7 \le 0,
\end{aligned} \tag{A1}$$

The pressure-vessel design problem seeks the shell thickness (x_1), head thickness (x_2), inner radius (x_3), and shell length (x_4) that minimize the manufacturing cost. The cost function is

$$\begin{aligned}
\min_{\vec{x}} \; & f(\vec{x}) = 0.6224\, x_1 x_3 x_4 + 1.7781\, x_2 x_3^2 + 3.1661\, x_1^2 x_4 + 19.84\, x_1^2 x_3, \\
\text{s.t.} \quad & g_1(\vec{x}) = -x_1 + 0.0193\, x_3 \le 0, \\
& g_2(\vec{x}) = -x_2 + 0.00954\, x_3 \le 0, \\
& g_3(\vec{x}) = -\pi x_3^2 x_4 - \tfrac{4}{3}\pi x_3^3 + 1{,}296{,}000 \le 0, \\
& g_4(\vec{x}) = x_4 - 240 \le 0.
\end{aligned} \tag{A2}$$

The decision variables obey $0.0625 \le x_1, x_2 \le 6.1875$ (in steps of 0.0625), and $10 \le x_3, x_4 \le 200$.

Appendix A.1. Planetary Gear Train

A simple model of a planetary gear train consists of a ring gear with R teeth, a sun gear with S teeth, and several planet gears, each with P teeth. The geometry requires that the ring gear have exactly the sum of the sun gear teeth plus twice the planet gear teeth,

$$g_1(\vec{x}) \;=\; R - (S + 2P) \;=\; 0. \tag{A3}$$

as described in [61]. When the ring gear is fixed, the turns ratio between the sun gear and the carrier (T_s and T_y, respectively) obeys

$$(R+S)\,T_y \;=\; S\,T_s. \tag{A4}$$

leading to a transmission ratio of $T_y/T_s = S/(R+S)$ [61]. To match a prescribed target ratio γ_{target}, one may minimize the squared deviation

$$\min_{R,S,P,n_p} \; f(\vec{x}) \;=\; \left(\frac{S}{R+S} - \gamma_{\text{target}}\right)^2. \tag{A5}$$

subject to the meshing requirement $g_1 = 0$ above and the planet spacing constraint that $R+S$ is an integer multiple of the number of planets n_p [61]. The variables R, S, P, and n_p are restricted to positive integers within practical bounds.

Appendix A.2. Ten-Bar Planar Truss

The ten-bar planar truss problem aims to select cross-sectional areas $x_1, \ldots, x_{10}$ for each bar to minimize the total structural weight. Assuming a constant material density ρ, the weight is proportional to

$$\min_{\vec{x}} \; f(\vec{x}) \;=\; \rho \sum_{j=1}^{10} x_j L_j. \tag{A6}$$

where L_j denotes the known length of bar j and the design variables are bounded by side constraints $0.1 \le x_j \le 35\,\text{in}^2$ [62]. Structural safety is enforced through stress and displacement constraints:

$$\begin{aligned} &|\sigma_j| \le 25\,\text{ksi}, \quad j = 1, \ldots, 10, \\ &|\delta_i| \le 2\,\text{in}, \quad \text{for each free node } i. \end{aligned} \tag{A7}$$

Two load cases with forces P_1 and P_2 are considered during analysis [62].

Appendix A.3. Three-Bar Truss

For the three-bar truss, the design variables are the cross-sectional areas x_1 and x_2, with $A_1 = A_3 = x_1$ and $A_2 = x_2$. The objective is to minimize the structural volume:

$$\min_{\vec{x}} \; f(\vec{x}) \;=\; \left(2\sqrt{2}\,x_1 + x_2\right) H. \tag{A8}$$

where $H = 100\,\text{cm}$ [60]. The stress constraints are

$$\begin{aligned} g_1(\vec{x}) &= \frac{P\left(\sqrt{2}x_1 + x_2\right)}{\sqrt{2}x_1^2 + 2x_1x_2} - \sigma \le 0, \\ g_2(\vec{x}) &= \frac{P\,x_2}{\sqrt{2}x_1^2 + 2x_1x_2} - \sigma \le 0, \\ g_3(\vec{x}) &= \frac{P}{x_1 + \sqrt{2}x_2} - \sigma \le 0. \end{aligned} \tag{A9}$$

with bounds $0 \le x_1 \le 1$ and $0 \le x_2 \le 1$. Here, $P = 2\,\text{kN/cm}^2$ and $\sigma = 2\,\text{kN/cm}^2$ [60].

References

1. Arora, R.K. *Optimization: Algorithms and Applications*; CRC Press: Boca Raton, FL, USA, 2015.
2. Heermann, D.W. Deterministic methods. In *Computer Simulation Methods in Theoretical Physics*; Springer: Berlin/Heidelberg, Germany, 1986; pp. 13–50.
3. Ólafsson, S. Metaheuristics. In *Handbooks in Operations Research and Management Science*; Elsevier: Amsterdam, The Netherlands, 2006; Volume 13, pp. 633–654.
4. Fagan, F.; Van Vuuren, J.H. A unification of the prevalent views on exploitation, exploration, intensification and diversification. *Int. J. Metaheuristics* **2013**, *2*, 294–327.
5. Adam, S.P.; Alexandropoulos, S.A.N.; Pardalos, P.M.; Vrahatis, M.N. No free lunch theorem: A review. In *Approximation and Optimization: Algorithms, Complexity and Applications*; Springer: Cham, Switzerland, 2019; pp. 57–82.
6. Bartz-Beielstein, T.; Branke, J.; Mehnen, J.; Mersmann, O. Evolutionary algorithms. *Data Min. Knowl. Discov.* **2014**, *4*, 178–195.
7. Chakraborty, A.; Kar, A.K. Swarm intelligence: A review of algorithms. In *Nature-Inspired Computing and Optimization*; Springer: Cham, Switzerland, 2017; pp. 475–494.
8. Siddique, N.H.; Adeli, H. *Nature-Inspired Computing: Physics and Chemistry-Based Algorithms*; Chapman and Hall/CRC: Boca Raton, FL, USA, 2017.
9. Črepinšek, M.; Liu, S.H.; Mernik, M. Exploration and exploitation in evolutionary algorithms: A survey. *ACM Comput. Surv. (CSUR)* **2013**, *45*, 1–33.
10. Cui, Y.; Ruan, L.; Dong, H.C.; Li, Q.; Wu, Z.; Zeng, T.; Fan, F.L. Cloud-rain: Point cloud analysis with reflectional invariance. *arXiv* **2023**, arXiv:2305.07814.
11. Zhou, B.; Zhang, H.; Han, S.; Ji, X. Crashworthiness analysis and optimization of a novel thin-walled multi-cell structure inspired by bamboo. *Structures* **2024**, *59*, 105827. https://doi.org/10.1016/j.istruc.2023.105827.
12. Almazroi, A.A.; Hassan, C.A.U. Nature-inspired solutions for energy sustainability using novel optimization methods. *PLoS ONE* **2023**, *18*, e0288490. https://doi.org/10.1371/journal.pone.0288490.
13. Janizadeh, S.; Thi Kieu Tran, T.; Bateni, S.M.; Jun, C.; Kim, D.; Trauernicht, C.; Heggy, E. Advancing the LightGBM approach with three novel nature-inspired optimizers for predicting wildfire susceptibility in Kaua'i and Moloka'i Islands, Hawaii. *Expert Syst. Appl.* **2024**, *258*, 124963. https://doi.org/10.1016/j.eswa.2024.124963.
14. Lakshmi, H. Demand Side Management Using a Novel Nature-Inspired Pelican Optimization Algorithm in a Smart Grid Environment. *Int. J. Electr. Electron. Eng.* **2024**, *11*, 238–246. https://doi.org/10.14445/23488379/IJEEE-V11I7P121.
15. Subha, E.; Jothi Prakash, V.; Antran Vijay, S.A. A novel arctic fox survival strategy inspired optimization algorithm. *J. Comb. Optim.* **2025**, *49*, 1. https://doi.org/10.1007/s10878-024-01233-8.
16. Abdel-Basset, M.; Mohamed, R.; Sallam, K.M.; Chakrabortty, R.K. Light Spectrum Optimizer: A Novel Physics-Inspired Metaheuristic Optimization Algorithm. *Mathematics* **2022**, *10*, 3466. https://doi.org/10.3390/math10193466.
17. Panigrahi, B.S.; Nagarajan, N.; Prasad, K.D.V.; Sathya; Salunkhe, S.S.; Kumar, P.D.; Kumar, M.A. Novel nature-inspired optimization approach-based svm for identifying the android malicious data. *Multimed. Tools Appl.* **2024**, *83*, 71579–71597. https://doi.org/10.1007/s11042-023-18097-5.
18. Maroosi, A.; Muniyandi, R.C. A novel membrane-inspired multiverse optimizer algorithm for quality of service-aware cloud web service composition with service level agreements. *Int. J. Commun. Syst.* **2023**, *36*, e5483. https://doi.org/10.1002/dac.5483.
19. Diab, H.Y.; Abdelsalam, M. A Novel Technique for the Optimization of Energy Cost Management and Operation of Microgrids Inspired from the Behavior of Egyptian Stray Dogs. *Inventions* **2024**, *9*, 88. https://doi.org/10.3390/inventions9040088.
20. Chandran, V.; Mohapatra, P. A novel reinforcement learning-inspired tunicate swarm algorithm for solving global optimization and engineering design problems. *J. Ind. Manag. Optim.* **2025**, *21*, 565–612. https://doi.org/10.3934/jimo.2024095.
21. Lang, Y.; Gao, Y. Dream Optimization Algorithm (DOA): A novel metaheuristic optimization algorithm inspired by human dreams and its applications to real-world engineering problems. *Comput. Methods Appl. Mech. Eng.* **2025**, *436*, 117718. https://doi.org/10.1016/j.cma.2024.117718.
22. Vais, R.I.; Sahay, K.; Chiranjeevi, T.; Devarapalli, R.; Knypiński, Ł. Parameter Extraction of Solar Photovoltaic Modules Using a Novel Bio-Inspired Swarm Intelligence Optimisation Algorithm. *Sustainability* **2023**, *15*, 8407. https://doi.org/10.3390/su15108407.
23. Omari, M.; Kaddi, M.; Salameh, K.; Alnoman, A.; Benhadji, M. Atomic Energy Optimization: A Novel Meta-Heuristic Inspired by Energy Dynamics and Dissipation. *IEEE Access* **2025**, *13*, 2801–2828. https://doi.org/10.1109/ACCESS.2024.3524322.
24. Ji, J.; Wu, T.; Yang, C. Neural population dynamics optimization algorithm: A novel brain-inspired meta-heuristic method. *Knowl.-Based Syst.* **2024**, *300*, 112194. https://doi.org/10.1016/j.knosys.2024.112194.
25. Salimi, K.; Dadashzadeh, S.; Aghaie, M. Optimization of isotopic binary and multi-component separation cascades using a novel nature-inspired horse herd algorithm. *Sep. Sci. Technol.* **2023**, *58*, 2988–3013. https://doi.org/10.1080/01496395.2023.2261076.
26. Abdel-Basset, M.; El-Shahat, D.; Jameel, M.; Abouhawwash, M. Exponential distribution optimizer (EDO): A novel math-inspired algorithm for global optimization and engineering problems. *Artif. Intell. Rev.* **2023**, *56*, 9329–9400. https://doi.org/10.1007/s10462-023-10403-9.

27. Sherif, A.; Haci, H. A Novel Bio-Inspired Energy Optimization for Two-Tier Wireless Communication Networks: A Grasshopper Optimization Algorithm (GOA)-Based Approach. *Electronics* **2023**, *12*, 1216. https://doi.org/10.3390/electronics12051216.
28. Kanagasabai, L. Novel mate preferences in human beings inspired optimization and hybridized pomsky-pygoscelis antarcticus-phengodes swarm algorithm. *Suranaree J. Sci. Technol.* **2024**, *31*, 1–13. https://doi.org/10.55766/sujst-2024-05-e03466.
29. Wang, J.; Yang, B.; Chen, Y.; Zeng, K.; Zhang, H.; Shu, H.; Chen, Y. Novel phasianidae inspired peafowl (Pavo muticus/cristatus) optimization algorithm: Design, evaluation, and SOFC models parameter estimation. *Sustain. Energy Technol. Assess.* **2022**, *50*, 101825. https://doi.org/10.1016/j.seta.2021.101825.
30. Peraza-Vázquez, H.; Peña-Delgado, A.; Merino-Treviño, M.; Morales-Cepeda, A.B.; Sinha, N. A novel metaheuristic inspired by horned lizard defense tactics. *Artif. Intell. Rev.* **2024**, *57*, 59. https://doi.org/10.1007/s10462-023-10653-7.
31. Kanagasabai, L. Novel Enriched Basil Seed Optimization, Little Child Imagination and Learning Inspired, Malignant Neoplasm of Uterine Algorithm. *Int. J. Autom. Smart Technol.* **2023**, *13*, 1–15. https://doi.org/10.5875/AUSMT.V13I1.2488.
32. Alomari, S.; Kaabneh, K.; AbuFalahah, I.; Gochhait, S.; Leonova, I.; Montazeri, Z.; Dehghani, M.; Eguchi, K. Carpet Weaver Optimization: A Novel Simple and Effective Human-Inspired Metaheuristic Algorithm. *Int. J. Intell. Eng. Syst.* **2024**, *17*, 230–242. https://doi.org/10.22266/IJIES2024.0831.18.
33. Arya, P.; Pandey, A.K.; Gopal Krishna Patro, S.; Tiwari, K.; Panigrahi, N.; Naveed, Q.N.; Lasisi, A.; Khan, W.A. MSCMGTB: A Novel Approach for Multimodal Social Media Content Moderation Using Hybrid Graph Theory and Bio-Inspired Optimization. *IEEE Access* **2024**, *12*, 73700–73718. https://doi.org/10.1109/ACCESS.2024.3400815.
34. Akopov, A.S. MBHGA: A Matrix-Based Hybrid Genetic Algorithm for Solving an Agent-Based Model of Controlled Trade Interactions. *IEEE Access* **2025**, *13*, 26843–26863. https://doi.org/10.1109/ACCESS.2025.3539460.
35. Akopov, A.S.; Beklaryan, A.L.; Zhukova, A.A. Optimization of Characteristics for a Stochastic Agent-Based Model of Goods Exchange with the Use of Parallel Hybrid Genetic Algorithm. *Cybern. Inf. Technol.* **2023**, *23*, 87–104. https://doi.org/10.2478/cait-2023-0015.
36. Lee, Z.J.; Su, S.F.; Chuang, C.C.; Liu, K.H. Genetic algorithm with ant colony optimization (GA-ACO) for multiple sequence alignment. *Appl. Soft Comput.* **2008**, *8*, 55–78. https://doi.org/10.1016/j.asoc.2006.10.012.
37. Xue, X.; Shu, T.; Xia, J. Automated Generation of Hybrid Metaheuristics Using Learning-to-Rank. *Algorithms* **2025**, *18*, 316. https://doi.org/10.3390/a18060316.
38. Shu, T.; Pan, Z.; Ding, Z.; Zu, Z. Resource scheduling optimization for industrial operating system using deep reinforcement learning and WOA algorithm. *Expert Syst. Appl.* **2024**, *255*, 124765. https://doi.org/10.1016/j.eswa.2024.124765.
39. Liang, Z.; Shu, T.; Ding, Z. A Novel Improved Whale Optimization Algorithm for Global Optimization and Engineering Applications. *Mathematics* **2024**, *12*, 636. https://doi.org/10.3390/math12050636.
40. Hu, G.; Huang, F.; Chen, K.; Wei, G. MNEARO: A meta swarm intelligence optimization algorithm for engineering applications. *Comput. Methods Appl. Mech. Eng.* **2024**, *419*, 116664. https://doi.org/10.1016/j.cma.2023.116664.
41. Sharma, P.; Raju, S. Metaheuristic optimization algorithms: A comprehensive overview and classification of benchmark test functions. *Soft Comput.-A Fusion Found. Methodol. Appl.* **2024**, *28*, 3123. https://doi.org/10.1007/s00500-023-09276-5.
42. Pan, J.S.; Hu, P.; Snášel, V.; Chu, S.C. A survey on binary metaheuristic algorithms and their engineering applications. *Artif. Intell. Rev.* **2023**, *56*, 6101–6167. https://doi.org/10.1007/s10462-022-10328-9.
43. Tizhoosh, H.R. Opposition-Based Learning: A New Scheme for Machine Intelligence. In Proceedings of the International Conference on Computational Intelligence for Modelling, Control and Automation (CIMCA), Vienna, Austria, 28–30 November 2005; IEEE: New York, NY, USA 2005; Volume 1, pp. 695–701. https://doi.org/10.1109/CIMCA.2005.1631345.
44. Li, S.; Chen, H.; Wang, M.; Heidari, A.A.; Mirjalili, S. Slime mould algorithm: A new method for stochastic optimization. *Future Gener. Comput. Syst.* **2020**, *111*, 300–323. https://doi.org/10.1016/j.future.2020.03.055.
45. Ahmadianfar, I.; Bozorg-Haddad, O.; Chu, X. Gradient-Based Optimizer: A New Metaheuristic Optimization Algorithm. *Inf. Sci.* **2020**, *540*, 131–159. https://doi.org/10.1016/j.ins.2020.06.037.
46. Seyyedabbasi, A.; Kiani, F. Sand Cat swarm optimization: A nature-inspired algorithm to solve global optimization problems. *Eng. Comput.* **2023**, *39*, 1601–1637. https://doi.org/10.1007/s00366-022-01604-x.
47. Mirjalili, S.; Lewis, A. The Whale Optimization Algorithm. *Adv. Eng. Softw.* **2016**, *95*, 51–67. https://doi.org/10.1016/j.advengsoft.2016.01.008.
48. Chou, J.S.; Truong, D.N. A novel metaheuristic optimizer inspired by behavior of jellyfish in ocean. *Appl. Math. Comput.* **2021**, *389*, 125535. https://doi.org/10.1016/j.amc.2020.125535.
49. Wang, C.; Wang, Z.; Han, Q.L.; Han, F.; Dong, H. Novel Leader-Follower-Based Particle Swarm Optimizer Inspired by Multiagent Systems: Algorithm, Experiments, and Applications. *IEEE Trans. Syst. Man Cybern. Syst.* **2023**, *53*, 1322–1334. https://doi.org/10.1109/TSMC.2022.3196853.
50. Wang, X.; Snášel, V.; Mirjalili, S.; Pan, J.S.; Kong, L.; Shehadeh, H.A. Artificial Protozoa Optimizer (APO): A novel bio-inspired metaheuristic algorithm for engineering optimization. *Knowl.-Based Syst.* **2024**, *295*, 111737. https://doi.org/10.1016/j.knosys.2024.111737.
51. Chopra, N.; Mohsin Ansari, M. Golden jackal optimization: A novel nature-inspired optimizer for engineering applications. *Expert Syst. Appl.* **2022**, *198*, 116924. https://doi.org/10.1016/j.eswa.2022.116924.

52. Mirjalili, S. Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm. *Knowl.-Based Syst.* **2015**, *89*, 228–249. https://doi.org/10.1016/j.knosys.2015.07.006.
53. Zhao, W.; Wang, L.; Zhang, Z. Artificial ecosystem-based optimization: A novel nature-inspired meta-heuristic algorithm. *Neural Comput. Appl.* **2020**, *32*, 9383–9425. https://doi.org/10.1007/s00521-019-04452-x.
54. Jiang, Y.; Wu, Q.; Zhu, S.; Zhang, L. Orca predation algorithm: A novel bio-inspired algorithm for global optimization problems. *Expert Syst. Appl.* **2022**, *188*, 116026. https://doi.org/10.1016/j.eswa.2021.116026.
55. Han, M.; Du, Z.; Yuen, K.F.; Zhu, H.; Li, Y.; Yuan, Q. Walrus optimizer: A novel nature-inspired metaheuristic algorithm. *Expert Syst. Appl.* **2024**, *239*, 122413. https://doi.org/10.1016/j.eswa.2023.122413.
56. Zhao, S.; Zhang, T.; Ma, S.; Wang, M. Sea-horse optimizer: A novel nature-inspired meta-heuristic for global optimization problems. *Appl. Intell.* **2023**, *53*, 11833–11860. https://doi.org/10.1007/s10489-022-03994-3.
57. Kennedy, J.; Eberhart, R.C. Particle Swarm Optimization. In Proceedings of the IEEE International Conference on Neural Networks, Perth, WA, Australia, 27 November–1 December 1995; Volume 4, pp. 1942–1948. https://doi.org/10.1109/ICNN.1995.488968.
58. Holland, J.H. *Adaptation in Natural and Artificial Systems*; University of Michigan Press: Ann Arbor, MI, USA, 1975.
59. Herrera, F.; Lozano, M.; Verdegay, J.L. Tackling real-coded genetic algorithms: Operators and tools. *Artif. Intell. Rev.* **1998**, *12*, 265–319. https://doi.org/10.1023/A:1006504901164.
60. NEORL Team. NEORL: Neural Evolutionary Optimization and Reinforcement Learning Framework. 2023. Available online: https://neorl.readthedocs.io/en/latest/ (accessed on 11 September 2025).
61. Wandel, M. Planetary Gears: Principles and Functions. 2023. Available online: https://woodgears.ca/gear/planetary.html (accessed on 11 September 2025).
62. XLOptimizer Team. XLOptimizer: Optimization Software for Engineering Design. 2023. Available online: https://www.xloptimizer.com/ (accessed on 11 September 2025).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.