



Paper Type: Original Article

Metaheuristic Optimization for Neuromorphic Computing Architectures

Fatemeh Zare Baghiabad^{1,*} , Viktor P. Sokolov²

¹ Department of Industrial Engineering, Faculty of Engineering, Ardakan University, Ardakan, Yazd, Iran; f.zare@ardakan.ac.ir.

² Department of Computer Science, Kazan Federal University, Kazan, Russia.

Citation:

Received: 04 February 2026

Revised: 20 April 2026

Accepted: 22 May 2026

Zare Baghiabad, F., & Sokolov, V. P. (2026). Metaheuristic optimization for Neuromorphic computing architectures. *Metaheuristic Algorithms with Applications*, 3(2), 195-216.

Abstract

Configuring Spiking Neural Network (SNN) topologies, synaptic parameters, and hardware mappings to minimize energy and latency on event-driven neuromorphic substrates is a decision problem characterized by multimodality, high dimensionality, mixed discrete–continuous variables, and noisy objective evaluations. These properties place the problem beyond the practical reach of classical exact methods and motivate metaheuristic search. This survey synthesizes 95 sources organized into three layers: The metaheuristic methodology itself (evolution-based, swarm-based, physics-based, and human/social-based families, together with hybrid and adaptive schemes), the neuromorphic hardware and SNN training context, and the comparatively sparse intersection literature that applies the former to the latter. Four contributions result: 1) a problem-oriented taxonomy links each algorithmic family to the stages of the neuromorphic design flow, anchored by a formal statement of the architecture optimization problem and a quantitative platform survey, 2) the empirical evidence is consolidated under a uniform assessment protocol (pairwise Wilcoxon signed-rank tests with Holm correction, Friedman ranking with Nemenyi critical-difference analysis, and budget-resolved performance measurement) exposing how strongly reported rankings depend on evaluation budgets, tuning symmetry, and benchmark selection, 3) documented weaknesses in hardware-aware evaluation, noise tolerance, and cross-study reproducibility are traced to their methodological roots, and 4) a five-point research roadmap (surrogate-assisted optimization, Adaptive Operator Selection (AOS), standardized neuromorphic benchmarks, reproducibility safeguards, and algorithm–hardware co-design) formulates concrete actions for each gap. The reviewed evidence indicates that operator-level innovation and methodological discipline, rather than the proliferation of nature-inspired metaphors, offer the most reliable route to gains in energy per inference, synaptic operations, and spike count.

Keywords: Spiking neural networks, Neuromorphic computing, Metaheuristic optimization, Hardware-aware optimization, Statistical comparison, Energy efficiency.

1 | Introduction

1.1 | Motivation

Neuromorphic computing has matured from a niche research program into one of the most actively pursued directions in energy-efficient computation [1], [2]. By mimicking the event-driven, sparse, and massively

 Corresponding Author: f.zare@ardakan.ac.ir

 <https://doi.org/10.48313/maa.vi.83>



Licensee System Analytics. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0>).

parallel character of biological nervous systems, neuromorphic architectures process information through spikes rather than dense matrix operations, offering substantial energy reductions for suitable workloads at matched accuracy [3], [4]. Large-scale platforms such as IBM TrueNorth [3], Intel Loihi [5], and SpiNNaker [6] have demonstrated spiking systems at deployment scale, while training methodologies imported from deep learning have steadily improved their task accuracy [7], [8]. Yet the very properties that make these systems efficient (their event-drivenness, distributed memory, and dependence on temporally coded signals) also make them extraordinarily difficult to configure.

The configuration task, which we call neuromorphic computing architecture optimization, involves jointly selecting network topologies, synaptic parameters, and hardware mappings so as to minimize energy and latency subject to accuracy and resource constraints. Its decision space mixes categorical choices (layer types, encoding schemes), integer-valued allocations (neuron counts per core, synaptic memory partitioning), and continuous parameters (thresholds, time constants, learning rates). Its objective landscape is rugged, noisy, and partially deceptive, because small perturbations of neuronal parameters can alter spike timing in discontinuous ways [7]. Classical exact and gradient-based methods are poorly suited to this setting, and the community has accordingly relied on metaheuristics, stochastic, derivative-free search procedures, in the sense established by Boussaïd et al. [9], with roots in evolutionary computation, swarm intelligence, and physics-inspired analogies [10].

1.2 | Scope, Methodology, and Relation to Prior Surveys

Several surveys have mapped the metaheuristic literature itself [9–11], and an extensive body of work has reviewed neuromorphic hardware and Spiking Neural Network (SNN) training [2], [7], [8]. These two strands, however, have developed largely in isolation: Metaheuristic surveys rarely account for the specific structure of neuromorphic objective functions, whereas neuromorphic surveys typically treat optimization as a solved background utility. The present survey occupies the intersection. It asks which metaheuristic families have been brought to bear on neuromorphic architecture optimization, what empirical evidence supports their reported advantages, and which methodological gaps most urgently constrain progress.

The synthesis covers literature from 2000 to 2025, with foundational precursors included where methodological clarity demands them (the statistical machinery, for instance, originates in the 1937–1979 period). Corpus construction proceeded in three stages: 1) the methodological layer was assembled from canonical algorithm-proposal papers, benchmark definitions, and statistical protocol sources, 2) the context layer was assembled from neuromorphic platform publications and SNN training methodology, including hardware-constrained training results that calibrate what optimized configurations achieve on real substrates [4], and 3) the intersection layer was assembled from studies applying population-based or trajectory-based search to neural network configuration, chiefly the neuroevolution lineage [12], [13]. Inclusion required either results on standardized benchmark suites or sufficient experimental detail for assessment; purely metaphorical algorithm proposals without empirical validation were excluded, following the critical standards articulated by Sörensen [14] and Campelo and Aranha [15]. The resulting corpus comprises 95 sources. A notable compositional fact, which the survey treats as a finding rather than a defect, is that the intersection layer is markedly thinner than either parent layer: The methodological literature dwarfs the domain-application literature, and several widely repeated claims about the superiority of specific algorithms on neuromorphic workloads rest on a small number of studies with heterogeneous protocols. Where reported results conflict, we attribute disagreement to identifiable protocol differences rather than adjudicating on aggregate win counts, a practice whose fragility is well documented in the no-free-lunch literature [16] and its refinements for realistic problem distributions [17].

1.3 | Contributions

The paper makes four contributions:

- I. Problem-oriented taxonomy: We organize the methodological corpus along four algorithmic families (evolution-based, swarm-based, physics-based, and human/social-based) and link each family to the stages of the neuromorphic design flow in which it is most plausibly deployed (Section 4, *Table 3*). The classification follows the inspiration-based conventions of [10], with porous boundaries acknowledged explicitly.
- II. A formal problem statement and quantitative platform survey: We give a consolidated mathematical formulation of neuromorphic architecture optimization that makes its objectives, platform-indexed constraints, stochastic evaluation, and mixed-variable structure explicit (Section 2.4), supported by a quantitative comparison of seven platform families (Section 2.3, *Table 1*).
- III. A uniform evidence-assessment protocol: We consolidate the comparative evidence under a single protocol (pairwise Wilcoxon signed-rank tests [18] with Holm correction [19], Friedman ranking with Nemenyi critical-difference analysis [20], [21], and budget-resolved measurement in the COCO tradition [22]) and use it to expose how strongly reported rankings depend on budgets, tuning symmetry, and benchmark selection (Section 5).
- IV. A five-point research roadmap: We distill the identified gaps into five interlocking challenges (surrogate-assisted optimization, Adaptive Operator Selection (AOS), standardized neuromorphic benchmarks, reproducibility safeguards, and algorithm–hardware co-design) and formulate concrete actions, metrics, and milestones for each (Section 7, *Table 5*).

1.4 | Organization

The remainder of the paper is organized as follows. Section 2 reviews neuromorphic computing foundations and formalizes the architecture optimization problem. Section 3 analyzes the structural properties that make this problem resistant to classical methods and reviews constraint handling. Section 4 develops the taxonomy of metaheuristic families and their hybrid and adaptive extensions. Section 5 consolidates the empirical evidence, emphasizing statistical methodology and cross-study comparability. Section 6 examines hardware-aware optimization and energy-efficiency analysis. Section 7 presents the research roadmap, and Section 8 concludes.

2 | Neuromorphic Computing Foundations

2.1 | Spiking Neural Networks and Neuron Models

SNNs are described as the third generation of neural network models [1], a designation reflecting their use of discrete events (spikes) rather than continuous activations as the unit of computation. The prevailing neuron abstraction is the Leaky Integrate-and-Fire (LIF) model, whose membrane potential V evolves as $\tau_m dV/dt = -(V - V_{\text{rest}}) + R I(t)$: Weighted input currents charge the membrane, leakage decays it toward the resting potential, a spike is emitted once V reaches the threshold θ , and V then resets, either abruptly to the resting potential or by continued integration below threshold, the reset mode itself being a configurable design choice with documented consequences for gradient-based training [7]. The LIF model strikes a pragmatic balance between biological plausibility and hardware cost, and its parameters (threshold, leak time constant, and reset behavior) are direct optimization targets. Information representation adds a further design layer: Rate coding communicates through spike frequency, temporal coding through precise spike times, and population coding through distributed activity patterns, and the choice among them is a topology-level decision that conditions both accuracy and energy [1], [8]. Because information is carried by spike timing as well as rate, small parameter changes can produce disproportionate changes in network behavior, a property that shapes the optimization landscape discussed in Section 3.

Training and configuration methodologies for SNNs span four families. Temporal coding with error backpropagation dates to SpikeProp [23], which navigated the non-differentiability of the spike function through spike-time encoding; modern surrogate-gradient methods circumvent the same obstacle by substituting smooth derivatives during the backward pass [7]. Conversion approaches transfer weights from pre-trained artificial networks into spiking equivalents [24], inheriting their accuracy at the cost of rate coding

and elevated spike counts. Local, biologically grounded plasticity, most prominently Spike-Timing-Dependent Plasticity (STDP) in its experimentally characterized form [25], [26], supports unsupervised feature learning without backpropagation [27]. Finally, hardware-constrained training optimizes network parameters directly under platform quantization and resource contracts [4]. These advances address a different problem from the one surveyed here: They optimize weights for a fixed architecture under a fixed hardware abstraction, whereas architecture optimization concerns the joint selection of topology, neuronal parameters, and hardware mapping. The two problems interact, training outcomes depend on architectural choices, so effective optimization methods must either account for training cost within their evaluation budgets or operate on training-free proxies [7], [8].

2.2 | Event-Driven Communication and Plasticity

Communication in neuromorphic systems is event-driven, though the realization differs by platform: Asynchronous Address-Event Representation (AER) links are the canonical mechanism for chip-to-chip spike exchange [6], [28], TrueNorth transports spikes across a synchronous on-chip network, Loihi employs an asynchronous mesh network-on-chip, and SpiNNaker routes multicast packets through source-programmed tables [3], [5], [6]. Whatever the transport, event-driven processing concentrates energy expenditure at active synapses, the principal source of the energy savings advertised by neuromorphic platforms, but it also couples network activity to communication load in ways that complicate optimization. Spike traffic is bursty, routing-dependent, and sensitive to placement decisions; two functionally equivalent mappings of the same network can differ substantially in latency and energy because of congestion effects that no topology-level metric captures [5], [6].

On-chip plasticity support is similarly heterogeneous: TrueNorth provides none (inference only) [3], SpiNNaker implements plasticity in software at the cost of compute [6], Loihi exposes programmable learning rules including STDP on-chip [5], and BrainScaleS-2 combines analog substrate dynamics with programmable digital plasticity processors [29]. This heterogeneity matters for optimization because the learning rule defines the trajectory along which a configuration's fitness unfolds: When STDP or another plasticity mechanism modifies synaptic state during the evaluation episode, potentiating causally timed pre-post pairs and depressing the rest, with an exponential dependence on spike-timing difference [25], [26], the objective is effectively a functional of a trajectory rather than a function of a point. Evaluations are therefore both expensive and noisy, reinforcing the case for derivative-free, noise-tolerant search [7].

2.3 | Hardware Platforms

Table 1 summarizes the seven platform families most relevant to architecture optimization, with the quantitative characteristics that define their resource contracts. Three design points organize the digital continuum. IBM TrueNorth demonstrated SNN inference at the million-neuron scale with roughly 70 mW typical power and approximately 26 pJ per synaptic operation, using 4,096 neurosynaptic cores, one million neurons, and 256 million shared, binary-weighted synapses, without on-chip learning [3]. Intel Loihi trades scale for programmability: 128 cores supporting approximately 130,000 neurons, on-chip programmable learning including STDP, an embedded x86 processor for management, and per-core synaptic memory partitioned from shared SRAM, fabricated in Intel's 14 nm process [5]; Loihi 2 subsequently introduced more flexible neuron models, a programmable event-driven learning framework, and the open-source Lava software stack [30]. SpiNNaker takes the opposite route: 18 general-purpose ARM968 cores per chip time-multiplex roughly a thousand neurons each, with 1024-entry multicast routing tables offering topological flexibility at the cost of table pressure and higher per-spike energy [6]; the design goal was flexible large-scale simulation rather than minimal energy per spike.

Analog and mixed-signal platforms extend the design space in the other direction. BrainScaleS implements neurons as analog circuits operating approximately $1,000\times$ faster than biological real time, enabling rapid evaluation of learning trajectories at the cost of device mismatch and parameter variability [31]; BrainScaleS-2 adds hybrid plasticity, pairing the analog substrate with digital plasticity processors [29]. The Neurogrid line

uses subthreshold analog neurons for extreme energy economy in the slow, sparse regime [32], and the DYNAP/ROLLS family demonstrates reconfigurable mixed-signal devices with 256 neurons and 128K synapses supporting on-line learning [28]. Tianjic takes a hybrid digital approach that natively supports both spiking and non-spiking computation within one core architecture, demonstrated on an unmanned bicycle experiment [33]. Finally, memristive and emerging-device substrates promise synaptic density beyond digital reach, at the price of cycle-to-cycle and device-to-device variability that turns every fitness evaluation into a random variable [2], [8].

Table 1. Neuromorphic platform families and their resource contracts.

Platform (year, Reference)	Substrate	Scale	Learning Support	Resource Contract Relevant to Optimization
TrueNorth [3]	Digital, 4096 cores	1M neurons, 256M synapses, ~70 mW, ~26 pJ/SOP	None (inference)	Hard per-core synaptic memory caps; shared binary weights
Loihi [5]	Digital, 128 cores, 14 nm	~130k neurons	On-chip programmable incl. STDP	Per-core SRAM partitioning for synapses; mesh routing
Loihi 2 [30]	Digital, improved cores	~1M neurons/chip (multi-chip)	Programmable event-driven learning; Lava stack	More neuron models; mature software contract
SpiNNaker [6], [32]	Digital, 18 ARM968/chip	~1B neurons (boards)	Software-defined	1024-entry multicast routing tables; timer-driven ticks
BrainScaleS-1/2 [29], [31]	Analog accelerated (~1000× real time)	Wafer-scale (BSS-1); chips (BSS-2)	Hybrid plasticity (BSS-2)	Device mismatch; parameter variability; calibration overhead
Neurogrid/DY NAP [28], [32]	Mixed analog-digital	1M neurons (Neurogrid); 256 neurons (ROLLS)	On-line learning (DYNAP line)	Subthreshold analog variability; limited fan-out
Tianjic [33]	Hybrid digital (SNN+ANN)	Core array	Variable	Unified core supports both coding schemes

The platform dimension matters because it parameterizes the optimization problem itself. Resource budgets, communication topology, and supported neuron models differ across the rows of *Table 1*; consequently, a configuration that is optimal on one platform can be infeasible or markedly suboptimal on another [3], [5], [6]. Any meaningful cross-study comparison must therefore record the target platform as part of the experimental protocol, a requirement that Section 5 finds to be honored more often in principle than in practice.

2.4 | The Architecture Optimization Problem

The configuration task admits the following consolidated formulation. Let the decision vector $\mathbf{x} = (\mathbf{t}, \boldsymbol{\theta}, \mathbf{m})$ collect topology decisions $\mathbf{t}$ (layer graph, neuron populations, encoding scheme), neuronal and synaptic parameters $\boldsymbol{\theta}$ (thresholds, time constants, weight initialization distributions, plasticity-rule parameters), and hardware mapping decisions $\mathbf{m}$ (placement of populations onto cores, synaptic memory allocation, routing choices). Each candidate $\mathbf{x}$ is evaluated on a platform p with resource contract $\mathbf{g}_p$, and because evaluation involves stochastic dynamics and device variability, fitness is a random variable realized per evaluation episode ω . The objective is then

$$\begin{aligned} \text{minimize } F(\mathbf{x}) = E_{\omega} [(E_p(\mathbf{x}, \omega), R_p(\mathbf{x}, \omega), L_p(\mathbf{x}, \omega), -A_p(\mathbf{x}, \omega))] \text{ subject to} \\ \mathbf{g}_p(\mathbf{x}) \leq 0, \mathbf{h}_p(\mathbf{x}) = 0, \mathbf{x} \in \Omega, \end{aligned} \quad (1)$$

where E is energy per inference, R aggregates synaptic operations, L is hardware latency, and A is task accuracy. The constraint set g_p encodes the platform resource limits of *Table 1*, h_p encodes functional requirements (task completion, real-time bounds, and the ordering constraints that ensure spike propagation precedes dependent computation), and Ω captures the mixed discrete–continuous domain of x . Studies differ in whether they scalarize F through weighted aggregation or treat it as a genuine multi-objective problem, approximating the Pareto front with methods such as the Non-dominated Sorting Genetic Algorithm II (NSGA-II) [34] and the Multiobjective Evolutionary Algorithm based on Decomposition (MOEA/D) [35]. *Table 2* summarizes the objectives, constraint types, and measurement modalities reported across the surveyed literature.

Table 2. Objectives, constraints, and measurement modalities in neuromorphic architecture optimization.

Element	Symbol/Role	Typical Definition	Measurement Modality
Energy per inference	$E(x)$	Dynamic plus amortized static energy per processed sample	Board-level power instrumentation on Loihi systems [4], [5]; calibrated estimates for TrueNorth [3]
Synaptic operations	$R(x)$	Count of synaptic events per episode (or per second, if stated)	On-chip counters [5]; analytic activity models [4]
Hardware latency	$L(x)$	End-to-end processing delay including routing and queuing	Timing instrumentation; congestion-aware simulation [6]
Task accuracy	$A(x)$	Workload-dependent performance metric	Benchmark task evaluation [7], [8]
Resource constraints	$g_p(x)$	Synaptic memory per core, fan-out, bandwidth, routing table entries	Platform contract (Table 1)
Functional constraints	$h_p(x)$	Task completion, real-time bounds, causality of dependent computation	Evaluation protocol

Two features of this formulation deserve emphasis because they recur throughout the surveyed literature. First, the objective components are hierarchically dependent rather than independent: spike count partially determines synaptic operations, which drive dynamic energy and, through congestion, latency, so that perturbing a single decision variable propagates through a causal chain rather than a separable sum [2], [6]. Second, when plasticity is active during evaluation, $F(x)$ is strictly a functional of a learning trajectory rather than a point evaluation [5], [25], [26]; when plasticity is disabled, $F(x)$ reduces to an ordinary noisy function. Both features penalize methods that assume independent or stationary objectives, and both favor population methods whose parallel candidates sample the trajectory-induced variance [10], [36].

3 | The Optimization Landscape and Constraint Handling

3.1 | Why Classical Methods Falter

Four structural properties of neuromorphic architecture optimization jointly resist classical exact and gradient-based approaches: 1) the landscape is multimodal: Equivalent accuracy is achievable by many distinct topologies and parameter regimes, and the energy and latency surfaces carry local minima induced by discrete mapping choices [10], 2) the dimensionality is high and mixed: Realistic configurations involve hundreds to thousands of decision variables combining categorical, integer, and continuous types, so enumeration is infeasible and smoothness assumptions fail at the discrete–continuous interface, 3) evaluations are expensive and noisy: A single fitness assessment may require simulating or deploying the network, running an inference or learning episode, and measuring hardware counters, with run-to-run variance arising from stochastic spike generation, analog device variability on mixed-signal substrates [29], [31], and scheduling nondeterminism on asynchronous platforms, and 4) constraints are coupled: Synaptic memory limits bind placement decisions, which alter communication patterns and hence latency, creating couplings that penalty-based scalarization handles awkwardly [37], [38].

These properties explain the field's convergence on metaheuristics, which trade global-optimality guarantees for robustness to multimodality, noise, and mixed variables [10], [11]. They do not, however, imply that gradient-based and model-based alternatives are irrelevant. Differentiable neural architecture search relaxes

the discrete search over architectures into a continuous optimization amenable to gradient descent [39], and platform-aware formulations of such searches incorporate latency models into the objective [40]; within SNN training, surrogate gradients extend the same philosophy across the spike non-differentiability [7]. Bayesian optimization, meanwhile, directly targets the expensive-black-box regime with principled uncertainty quantification [41] and is the natural competitor for metaheuristics under tight budgets (Section 6.1). The honest positioning is therefore comparative: Gradient-based methods dominate when the objective is differentiable or can be made so, model-based methods dominate when evaluations are scarce and the dimension is moderate, and metaheuristics dominate when discreteness, multimodality, and population-level diversity requirements co-occur, as they do in full architecture optimization. The choice among metaheuristics cannot be settled once and for all: The no-free-lunch theorems establish that averaged over all problem distributions no search algorithm outperforms any other [16], and refined analyses show that practically relevant problem classes can still admit uniformly superior methods [17]. The operative question, addressed in Sections 5 and 6, is which families align with which structural features of neuromorphic instances, and under which evaluation protocols their advantages persist.

3.2 | Constraint-Handling Portfolio

The surveyed studies deploy a recognizable portfolio of constraint-handling techniques, reviewed comprehensively in the evolutionary computation literature [37], [38], [42]. Penalty functions append constraint violations to the objective with coefficients that must themselves be tuned, a nontrivial burden in expensive-evaluation settings. Feasibility rules, notably Deb's [37] tournament principle, select feasible individuals over infeasible ones and rank infeasible ones by violation magnitude, thereby avoiding penalty coefficients entirely. Repair operators project infeasible configurations onto feasible ones, natural for mapping problems where capacity violations can be resolved by greedy reassignment, though repairs risk channeling search toward particular regions of the feasible set. Stochastic ranking balances objective and violation information probabilistically, and ϵ -constraint methods progressively tighten feasibility tolerances, both surveyed in [42]. Finally, multi-objective reformulation treats constraint violation as an additional objective to be minimized alongside the original criteria [38], fitting naturally into frameworks already handling multiple objectives [34], [35]. The choice among these techniques interacts with the metaheuristic family: Population-based methods with tournament selection pair naturally with feasibility rules, whereas decomposition-based schemes absorb constraints as additional components of subproblems, with MOEA/D's constraint handling operating on the decomposed scalar subobjectives [35].

4 | A Taxonomy of Metaheuristic Families

We organize the methodological corpus into four families following the inspiration-based classification conventions of the metaheuristic literature [10], [11], while stressing that family boundaries are porous; several algorithms admit defensible placements in multiple families, and we flag the borderline cases explicitly. *Table 3* summarizes the families, representative algorithms, and their principal deployment roles in the neuromorphic design flow. *Fig. 1* shows the temporal distribution of the 95 surveyed sources by category, and two compositional patterns are worth noting before the family-by-family discussion. First, the swarm-based family dominates the 2016–2020 period, reflecting the proliferation of nature-inspired algorithms during those years rather than a corresponding growth of validation evidence. Second, the methodological and benchmark literature (the rightmost series in each period) persists across all periods at a steady rate, whereas sources specific to neuromorphic platforms and SNN training accelerate visibly from the early 2010s onward—an asymmetry that quantifies the methodological imbalance this survey documents.

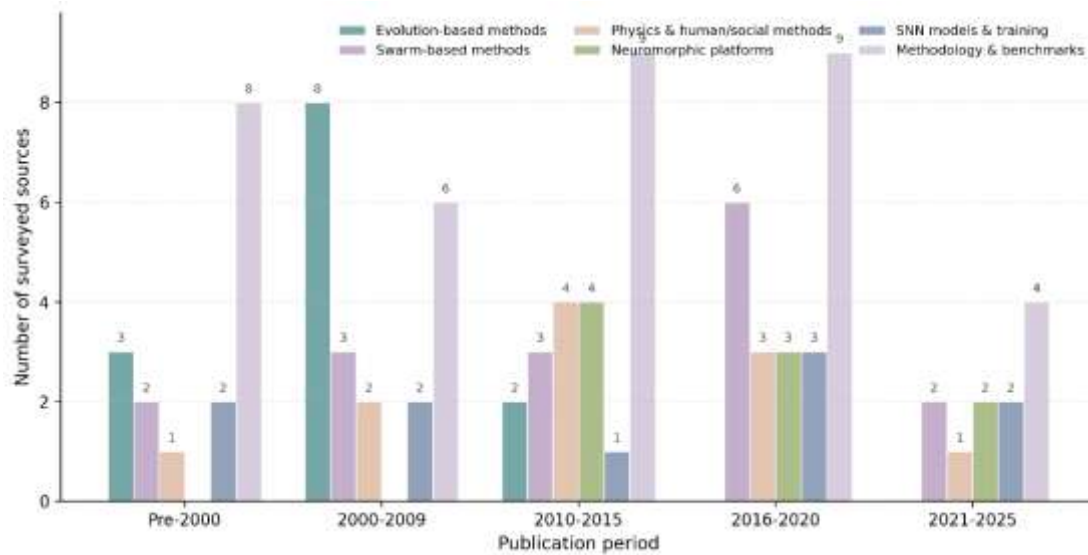


Fig. 1. Temporal distribution of the 95 surveyed sources by category and publication period.

Table 3. Problem-oriented taxonomy of metaheuristic families applied to neuromorphic architecture optimization.

Family	Representative Algorithms (Year)	Operator Structure	Principal Deployment in the Neuromorphic Design Flow
Evolution-based	GA [12], [43]; CMA-ES [36]; DE [44]; SaDE [45]; JADE [46]; SHADE [47]; L-SHADE [48]; NSGA-II [34]; MOEA/D [35]; BBO [49]; NEAT [13]	Selection, recombination, mutation; endogenous or success-history parameter adaptation	Topology search and neuroevolution (GA, NEAT); continuous parameter tuning (CMA-ES, DE variants); multi-objective energy–accuracy fronts (NSGA-II, MOEA/D)
Swarm-based	PSO [50]; ACO [51]; ABC [52]; GWO [53]; WOA [54]; ALO [55]; MFO [56]; SCA [57]; DA [58]; SSA [59]; GOA [60]; HHO [61]; BFA [62]; CS [63]; AO [64]; TSRO [65]	Social attraction/repulsion, stigmergy, hierarchy-based position updates	Continuous synaptic parameter tuning under tight budgets (PSO); constructive mapping and placement (ACO); hyperparameter selection and pruning (GWO, WOA, HHO successors)
Physics-based	SA [66]; GSA [67]; FA [68]; BA [69]; FPA [70]; WCA [71]; EO [72]; MVO [73] and binary variants [74]; water optimization algorithm [75]	Annealing schedules, force-based attraction, mode switching between local/global phases	Trajectory-based refinement of population solutions (SA); exploration-heavy global phases in hybrid schemes
Human/social-based	TLBO [76]	Teaching/learning phases; classroom analogies	Low-tuning-budget settings exploiting parameter-free design (TLBO)

Note. Assignments follow the inspiration-based conventions of [10] where applicable; borderline cases are acknowledged: FA and BA are classified here as physics-based following their light- and sound-propagation analogies, though their bio-inspiration makes swarm-based placements equally defensible; Biogeography-Based Optimization (BBO) is placed with evolution-based methods given its migration-and-mutation operator structure; Trees Social Relations Optimization (TSRO) self-describes as swarm-based [65]; and the two water-inspired algorithms (Water Cycle Algorithm (WCA) and the water optimization algorithm) are both grouped under physics-based following their hydraulic framing [71], [75]. The deployment column reflects usage patterns reported in the surveyed literature rather than exclusivity.

4.1 | Evolution-Based Methods

Evolution-based methods maintain a population of candidate configurations that undergo selection, recombination, and mutation. Genetic algorithms, in both their canonical form [43] and applied toolchain

elaborations [12], were the first family applied to neural architecture problems, and Neuroevolution through Augmenting Topologies (NEAT) [13] demonstrated that topology and weight evolution could proceed jointly through speciation and incremental complexification, a lineage directly ancestral to modern architecture search. Evolution strategies occupy the opposite end of the representational spectrum: By adapting mutation strength endogenously, the Covariance Matrix Adaptation Evolution Strategy (CMA-ES) learns the local covariance structure of the continuous parameter subspace [36], an advantage when tuning neuronal time constants and thresholds whose sensitivities are strongly anisotropic. Differential Evolution (DE) interpolates between these poles through difference-vector mutation [44]; its modern variants (strategy adaptation (SaDE) [45], external archives (JADE) [46], and the success-history based adaptive DE (SHADE) [47] with its linearly size-reduced extension L-SHADE [48]) have established themselves as consistently strong performers on the IEEE Congress on Evolutionary Computation (CEC) benchmark suites [77–79], which Section 5 revisits. BBO [49] rounds out the family with migration operators that admit natural island interpretations for placement problems. For the multi-objective formulations that neuromorphic energy–accuracy trade-offs demand, NSGA-II's fast non-dominated sorting [34] and MOEA/D's decomposition of the objective space [35] are the two reference frameworks throughout the surveyed period.

4.2 | Swarm-Based Methods

Swarm-based methods coordinate populations through social rather than reproductive operators. Particle Swarm Optimization (PSO) [50], the family's founding member, updates particle velocities from personal and social attractors and is attractive for continuous parameter tuning because of its low per-iteration cost and rapid early convergence, at the price of premature convergence on rotated multimodal functions [79]. Ant Colony Optimization (ACO) [51] and Artificial Bee Colony (ABC) [52] extend the family to discrete construction and foraging-style exploration, with ACO historically attractive for mapping and placement problems whose structure admits constructive solutions. The past decade has added a large cohort of predator–prey and hierarchy-inspired algorithms: Grey Wolf Optimizer (GWO) [53], Whale Optimization Algorithm (WOA) [54], Ant Lion Optimizer (ALO) [55], Moth-Flame Optimization (MFO) [56], Sine Cosine Algorithm (SCA) [57], Dragonfly Algorithm (DA) [58], Salp Swarm Algorithm (SSA) [59], Grasshopper Optimization Algorithm (GOA) [60], Harris Hawks Optimization (HHO) [61], Bacterial Foraging Optimization (BFA) [62], Cuckoo Search (CS) [63], Aquila Optimizer (AO) [64], and TSRO [65]. Reports of their application to SNN hyperparameter selection and pruning appear throughout the engineering literature, although typically under the protocol limitations that the critique literature has documented for this class of algorithms [14], [15]. The empirical assessment in Section 5, however, tempers expectations about aggregate superiority in two further respects: The validation practices of the metaphor-driven stream have drawn recurrent criticism [14], [15], and the intersection studies that would substantiate domain-specific advantage are fewer and less uniform in protocol than the volume of algorithm proposals suggests.

4.3 | Physics-Based and Human/Social-Based Methods

Physics-based methods derive operators from physical annealing, force, or propagation analogies. Simulated Annealing (SA) [66], the oldest member, remains relevant as a trajectory-based refinement mechanism, particularly in hybrid schemes refining solutions produced by population-based search; indeed, SA's own pedigree lies in combinatorial optimization. Gravitational Search Algorithm (GSA) [67] accelerates masses toward better-performing agents; Firefly Algorithm (FA) [68] and Bat Algorithm (BA) [69] implement light-attraction and echolocation analogies respectively; Flower Pollination Algorithm (FPA) [70] alternates local and global pollination modes; WCA [71] models runoff and evaporation; Equilibrium Optimizer (EO) [72] frames search as convergence to equilibrium states; and Multi-Verse Optimization (MVO) [73], including its binary variant applied to flow-shop scheduling [74], complete the family together with the water optimization algorithm [75], a recent hydraulic-framing addition whose validation remains preliminary. The family's characteristic strength is controlled exploration through physically interpretable schedules; its characteristic weakness is sensitivity to schedule parameters, which in expensive-evaluation neuromorphic settings is

difficult to calibrate because each schedule assessment consumes scarce evaluations [80]. Human- and social-based methods abstract learning and social interaction instead of natural law. Teaching–Learning-Based Optimization (TLBO) [76] alternates teacher and learner phases and is notable for requiring no algorithm-specific control parameters beyond population size, easing the tuning burden that Section 5 identifies as a major confound; it remains the family's principal representative in the surveyed corpus.

4.4 | Hybridization, Adaptive Operator Selection, and Algorithm Selection

Three cross-cutting developments cut across the four families: 1) hybridization, combining population-based global search with trajectory-based refinement, or operators from different families, consistently reports improved solution quality on neuromorphic objectives at the cost of additional design parameters that must themselves be tuned [80], [11], 2) AOS and parameter control replace fixed operator and parameter choices with dynamic allocation: The AOS framework of Pham et al. [81] credits operators by recent success, and the parameter-control taxonomy of Eiben et al. [82] and Karafotias et al. [83] distinguishes what is adapted, how, and when. The DE lineage demonstrates the approach's maturity: SaDE adapts strategy probabilities [45], JADE adapts mutation and crossover parameters with an external archive [46], and SHADE/L-SHADE maintain success-history memories [47], [48]. Section 7 argues that parameter control is among the most promising responses to the heterogeneous, time-varying structure of neuromorphic search landscapes, and 3) the algorithm selection problem, choosing the right solver for a given instance rather than designing a universal one [84], has developed a dedicated literature for continuous optimization [85], [86] whose feature-based instance characterization could transfer naturally to neuromorphic workloads, where problem features (dimension, constraint density, noise level, budget scale) are in principle observable in advance.

5 | Empirical Evidence: Benchmarks, Statistics, and Synthesis

5.1 | Benchmark Suites and Protocol Heterogeneity

Metaheuristic comparisons in this domain rest on two layers of benchmarks. The first layer comprises numerical optimization suites, CEC 2017 [77] and CEC 2020 [78], whose shifted, rotated, and hybrid functions probe multimodality, ill-conditioning, and linkage. The systematic CEC 2017 assessment by Werth et al. [79] found that DE-family methods occupied the leading ranks while many newer nature-inspired methods did not consistently outperform their predecessors; the scope of that finding is bounded by the study's algorithm set and budget. Complementary to the fixed-budget CEC format, the COCO platform measures performance as a function of evaluation budget, reporting Expected Running Time (ERT) and Empirical Cumulative Distribution Functions (ECDF) that expose budget-conditioned orderings directly [22]. The second layer comprises domain-specific evaluations on neuromorphic workloads, where objectives are energy per inference, synaptic operations, latency, and accuracy measured in simulation or on hardware, over datasets that have themselves consolidated in the past decade: N-MNIST for saccade-converted digit recognition [87], DVS128 Gesture for event-based gesture recognition [88], and the Heidelberg spiking datasets for systematic SNN evaluation [89]. This layer carries the ecological validity that numerical suites lack, but it is afflicted by protocol heterogeneity: Target platforms differ (*Table 1*), evaluation budgets vary by orders of magnitude, and tuning effort is seldom reported. Benchmarking initiatives such as NeuroBench [90] and the Loihi 2 ecosystem's evaluation stack [30] are beginning to address this fragmentation, but no community standard yet governs these evaluations for architecture optimization; the absence of such a standard is, in our assessment, the single largest obstacle to cumulative progress in this field.

5.2 | Statistical Comparison Protocol

Sound comparison requires explicit statistical machinery, for which the tutorial protocol of Derrac et al. [90] (pairwise Wilcoxon signed-rank tests [18] with Holm correction [19], complemented by Friedman [20] ranking with Nemenyi critical-difference analysis, following the framework systematized for classifier comparison by Demsar [21]) has become the de facto standard. The Wilcoxon signed-rank test detects consistent pairwise

superiority under mild assumptions, paired observations and, for exact significance levels, symmetry of the difference distribution under the null hypothesis, rather than being fully assumption-free, a caveat worth stating because difference distributions on benchmark suites are frequently skewed. Friedman ranking aggregates performance across functions or instances into a single ordering, with the Nemenyi critical difference delimiting which rank gaps are statistically meaningful [21]; the Iman–Davenport correction offers a more sensitive omnibus test under its chi-square approximation when rank distributions permit [90].

Two practices undermine this machinery in practice, and their statistical mechanics deserve precision. Within-study tuning, granting the proposed algorithm more tuning effort than its baselines, does not break the null hypothesis of the significance test itself; rather, it changes the estimand from "algorithm" to "algorithm plus its tuning package," so the resulting significance statements quantify tuning asymmetry as much as algorithmic merit [14], [79]. Selective reporting of benchmark subsets, in turn, exploits the multiplicity that Holm correction is designed to control, since excluded functions carry no correction penalty [19]. A third, quieter weakness is statistical power: With the run counts and function counts typical of the surveyed studies, only large effects reach significance, and "no significant difference" is routinely over-read as equivalence. We therefore adopt the position that claims of algorithmic advantage should be accompanied by an analysis of how conclusions vary with budget, tuning effort, and benchmark composition [22], a form of robustness checking that remains rare in the surveyed literature. Sensitivity-analysis methods in the Sobol' tradition support exactly this kind of conclusion-stability audit when inputs (budgets, tuning levels, subset choices) can be varied systematically [91], though their cost grows quickly with the number of factors and their application to protocol variables is a methodological analogy rather than an established practice.

5.3 | Cross-Study Synthesis

Table 4 consolidates the comparative evidence synthesized from the surveyed studies, organized by benchmark family and algorithm family. Because the underlying studies differ in platforms, budgets, and metrics, direct numeric aggregation would be misleading; entries therefore record the direction and consistency of reported findings, together with an explicit assessment of the evidence base behind each row. The coding rule is stated here once: A cell receives "++" only when multiple independent studies with symmetric tuning report consistent advantage; "+" when directionally favorable evidence exists but is thinner; "o" when findings are mixed; "–" when the balance of evidence is unfavorable. Row-level evidence depth is characterized as broad (many studies across groups), moderate (several studies, overlapping groups), or sparse (few studies, heterogeneous protocols), and it is exactly the sparsity pattern in the lower rows that constitutes one of this survey's findings.

Table 4. Synthesis of reported comparative evidence by benchmark family; entry semantics defined in text.

Benchmark Family	Evolution-Based	Swarm-Based	Physics-Based	Human/Social	Evidence Depth
CEC numerical suites [77], [78]	++ (DE variants, CMA-ES) [36], [44–48]	o (PSO strong on separable/unimodal, weaker on rotated multimodal) [50], [79]	o (schedule-sensitive) [66], [67]	o (limited evidence) [76]	Broad [22], [79]
Continuous SNN parameter tuning	+ (CMA-ES on anisotropic landscapes) [36]	+ under short budgets (PSO, GWO) [50], [53]	o [66]	o [76]	Sparse
Topology / Architecture search	+ (GA, NEAT lineage) [13], [43]	o (discretization overhead) [10]	o (schedule-sensitive) [66]	o [76]	Moderate (neuroevolution line) [13]
Mapping and placement	o (feasibility-rule pairing) [37]	+ (ACO constructive bias) [51]	o [66]	o [76]	Sparse
Multi-objective energy–accuracy fronts	+ (established reference frameworks) [34], [35]	o [10]	o [66]	o [76]	Methodologically broad; domain-sparse

Note: Entries summarize the direction and consistency of findings reported in the surveyed studies under their respective protocols; they are not outcomes of a single controlled experiment, and the budget and tuning asymmetries documented in Section 5.2 qualify all entries. Entries in the CEC row rest on broad, replicated evidence [22], [79]; entries in the four domain rows record the documented transfer of family-level strengths to the domain's structural features, hypothesized fit awaiting the controlled comparisons whose scarcity this survey documents, rather than replicated controlled studies. The sparse evidence depth of the domain rows is itself a survey finding (Section 5.4).

Three patterns recur with sufficient consistency to state as findings, each qualified by its evidence base: 1) on the controlled numerical suites, DE variants and CMA-ES consistently occupy the leading ranks, with PSO strong on separable and unimodal functions but weaker on rotated multimodal ones [22], [79], 2) on neuromorphic-specific workloads, no family dominates across all objective combinations: Reported advantages of swarm-based methods concentrate on continuous parameter tuning under tight budgets, whereas evolution-based methods dominate when topology search or multi-objective trade-offs are involved [13], [34], [35], but this pattern rests on a substantially thinner corpus than the first, and the direction of individual studies is conditioned by their platforms and protocols, and 3) reported rankings are strongly conditioned on evaluation budget: methods with aggressive early convergence (PSO, GWO) lead under short budgets, while methods with sustained diversification (DE variants, hybrid schemes) overtake them as budgets grow [80], a pattern that budget-resolved measurement frameworks make directly visible [22] and that constitutes the strongest argument for reporting performance as a function of budget rather than at a single terminal budget. These patterns jointly support our central empirical claim: algorithmic progress over the surveyed period has been real but incremental, with the largest attributable gains coming from operator-level innovations, parameter adaptation, external archives, success-history learning [36], [45–48], rather than from new metaphors [14], [15].

5.4 | Qualitative Observations and Threats to Validity

Four qualitative observations, drawn repeatedly across the surveyed studies, qualify every quantitative claim: 1) hardware-awareness changes rankings: Incorporating mapping and communication costs into the objective can reverse the ordering produced by simulation-only fitness, because placement-induced congestion penalizes configurations that appear optimal in isolation (Section 6), 2) noise tolerance differentiates families: Methods whose selection pressure concentrates rapidly on single candidates are more vulnerable to evaluation noise than methods maintaining explicit diversity or restart mechanisms [80], 3) budget-conditioned superiority is the norm rather than the exception [16], [17]; published rankings should therefore always be read jointly with their budgets, and 4) the prevalence of within-study tuning means that headline improvements over baselines frequently shrink or vanish when baselines receive equal tuning effort [14], [79].

A candid account of the threats to the validity of this synthesis is also required, using the four standard categories of the validity typology [92] supplemented here by publication bias as a domain-specific fifth threat. Publication and selection bias arise because the surveyed literature over-represents successful applications: Algorithms that failed on neuromorphic workloads are seldom reported, biasing family-level assessments toward optimism. Construct threats arise from metric heterogeneity, energy measured under different power regimes, accuracy on different tasks, so our cross-study aggregation relies on directional consistency rather than magnitude. Internal validity threats arise from the confounding of algorithmic merit with implementation quality; a well-engineered DE implementation can outperform a poorly engineered implementation of a nominally stronger method, and published comparisons rarely disentangle the two [79]. Statistical conclusion validity threats arise from the power and multiplicity problems of Section 5.2. Finally, external validity threats arise from the rapid evolution of neuromorphic platforms: conclusions calibrated on TrueNorth- and Loihi-class contracts [3], [5] may transfer imperfectly to analog and memristive substrates whose variability profiles differ qualitatively [2], [29], [31]. We flag these threats so that readers can weigh our findings accordingly, and because acknowledging them is a precondition of the reproducibility agenda advocated in Section 7.

6 | Hardware-Aware Optimization and Energy-Efficiency Analysis

6.1 | Evaluation Cost and Surrogate Strategies

Hardware-in-the-loop evaluation, deploying each candidate configuration on Loihi, SpiNNaker, or a mixed-signal testbed to measure energy and latency directly, is the gold standard for ecological validity but is prohibitively expensive for population-based methods requiring thousands of evaluations [5], [6]. Simulation-based evaluation is cheaper but imperfectly correlated with hardware outcomes, particularly for communication-dominated workloads where simulator abstractions elide congestion and routing effects [6]. The standard mitigation is surrogate-assisted optimization: Learn a regression model of the objective from evaluated configurations, use it to filter or rank candidates, and evaluate in hardware or high-fidelity simulation only the most promising subset, a paradigm surveyed systematically in the data-driven evolutionary optimization literature [93]. Bayesian optimization occupies the adjacent niche, targeting the same expensive-black-box regime with sequential posterior inference and acquisition functions [41]; its sequential character suits very tight budgets, whereas population-based surrogate methods amortize model updates across a candidate pool and couple more naturally with the diversity requirements of multimodal neuromorphic landscapes [36], [93].

The economics of this trade-off deserve explicit accounting because they are often left implicit. Let the cost ratio between a hardware evaluation and a simulator evaluation be q ; a population method with population size P and generation count G consumes $P \cdot G$ evaluations, of which only a fraction can realistically be hardware-in-the-loop even for moderate P and G . Three budget architectures consequently coexist in the surveyed literature: Full simulation with hardware validation of the final configuration (cheapest, weakest evidence); tiered screening in which simulation pre-filters candidates for hardware evaluation (the dominant surrogate pattern); and full hardware-in-the-loop search (rare, confined to small configuration spaces or academic clusters, as in the accelerated-evaluation regime that analog platforms enable [29], [31]). Reporting the chosen architecture, the effective budget, and the simulator–hardware correlation coefficient would materially improve cross-study interpretation, yet all three are reported only sporadically; an omission the roadmap of Section 7 addresses. The principal open problems in surrogate practice are model staleness as search moves away from sampled regions and systematic bias toward regions the surrogate has seen [93].

6.2 | Mapping and Placement Constraints

The mapping stage, assigning populations to cores, allocating synaptic memory, and routing spike traffic, accounts for much of the discrete structure of the optimization problem. TrueNorth's shared synapse model imposes hard per-core synaptic memory caps [3]; Loihi partitions per-core SRAM among synaptic storage, neuron state, and routing, so that memory pressure in one category is relieved only by sacrificing another [5]; SpiNNaker's 1024-entry multicast routing tables bind the number of distinct multicast patterns a population may emit [6]. Violations render configurations infeasible regardless of nominal fitness, so mapping is exactly where the constraint-handling portfolio of Section 3.2 is exercised: Feasibility rules dominate in population-based schemes [37], while repair-style greedy reassignment is common in constructive approaches [51]. Reported experience converges on a practical lesson: Jointly optimizing topology and mapping outperforms the decoupled alternative of optimizing topology first and mapping afterward, because the decoupled approach forfeits interaction effects, compact topologies that fit locality constraints, that dominate energy and latency on real substrates [3], [5]. The evidential status of this lesson deserves candor: It is supported by converging practitioner experience across the platform literature rather than by a single controlled comparison, and confirming it under the statistical protocol of Section 5.2 would itself be a valuable contribution.

6.3 | Energy Metrics and Multi-Objective Trade-Offs

Energy-efficiency analysis in this domain is inseparable from multi-objective formulation, because the criteria of interest genuinely compete: reducing spike count typically costs accuracy; replicating synapses to reduce latency raises area and static power; aggressive pruning improves energy and memory but degrades noise margins [2], [8]. The surveyed studies therefore converge on Pareto-based treatment, with NSGA-II [34] and MOEA/D [35] as the dominant engines, on reporting trade-off fronts rather than scalar optima, and on quality indicators such as the hypervolume for front assessment [94]. Decomposition approaches scale more gracefully to many objectives and permit preference articulation through weight vectors [35]; dominance-based approaches require fewer assumptions and remain the default for two- and three-objective instances [34].

Methodological care is required on three metric conventions, whose absence undermines the paper's own anti-conflation stance when overlooked: 1) synaptic operations should be reported as a count per episode or as a rate per second, never conflated: The two differ by the workload's duration and duty cycle, and cross-study comparisons that mix them are meaningless, 2) energy per inference must state its amortization convention: dynamic-only figures ignore leakage, which dominates on sparse workloads precisely where neuromorphic advantages are largest, while inclusive figures require a stated utilization assumption [3], [5], and 3) power measurement methodology matters: Loihi results, for instance, derive from board-level instrumentation of supply rails rather than from on-chip energy counters, with consequences for attribution across chip and board components [4], [5]. Finally, energy comparisons across coding schemes are meaningful only at matched task accuracy: The TrueNorth line's constrained-training results [4] exemplify the iso-accuracy convention, without which reported orders-of-magnitude savings invite apples-to-oranges readings.

6.4 | Robustness and Noise Tolerance

Robustness enters the analysis at two levels. At the device level, analog and memristive substrates exhibit cycle-to-cycle and device-to-device variability, so a configuration's realized fitness is a random variable [2], [31]; at the system level, stochastic spike generation and scheduling nondeterminism add evaluation noise even on digital platforms [5]. Optimization methods respond along a spectrum: Explicit averaging over repeated evaluations (costly), fitness inheritance from similar evaluated candidates, and robust formulations optimizing a worst-case or quantile statistic instead of the mean. Variance-based sensitivity analysis in the Sobol' tradition [91] serves a screening role here, decomposing output variance among decision variables to identify which dominate and thus admit dimensionality reduction, at a cost of roughly $N(n+2)$ model evaluations for n variables and N base samples, which confines routine use to simulator-based fitness; the indices are also sensitive to output noise, further limiting their use on hardware-measured objectives. Certifying that a specific optimum remains stable under perturbation is a related but distinct robustness question requiring explicit perturbation testing rather than variance decomposition alone. The surveyed evidence on robustness-oriented optimization remains thin: Suggestive reports indicate that configurations optimized under worst-case or quantile statistics outperform nominal optima as deployment duration and variability exposure grow, but the study base is too sparse to support quantitative claims. We flag this as an open area.

7 | Research Roadmap

The gaps identified in Sections 5 and 6 crystallize into five interlocking challenges, summarized in *Table 5* with their state of the art, principal barriers, and recommended actions; the subsections that follow add metrics and milestones that the table cannot carry.

Table 5. Research roadmap: Five priority challenges for metaheuristic neuromorphic architecture optimization.

#	Challenge	State of the Art	Principal Barrier	Recommended Action
1	Surrogate-assisted optimization [36], [93]	Mature frameworks in data-driven EC [93]; ad hoc use in domain studies	Model staleness; unquantified bias toward sampled regions	Treat surrogate selection as protocol; report calibration diagnostics; combine with trust-region or archive mechanisms
2	AOS and parameter control [81-83]	Mature in DE lineage and AOS frameworks [45-48]	Untested on neuromorphic objectives; adaptation overhead	Port adaptation machinery; evaluate transferability across instances, not only final fitness
3	Standardized neuromorphic benchmarks [77], [78], [90]	CEC/COCO for numerical layer [22]; NeuroBench emerging [90]; platform stacks [30]	No community contract for platform models and budgets	Tiered suite: numerical foundations plus domain workloads with published platform contracts and budget disclosure
4	Reproducibility and fair comparison [21], [22], [90]	Statistical protocol available and widely cited [90]	Within-study tuning; selective reporting; unreported budgets	Pre-registered protocols; release configurations, seeds, raw data; Wilcoxon-Friedman-CD reporting with budget-resolved curves
5	Algorithm-hardware co-design [7], [37], [38]	Decoupled topology-then-mapping workflows	Interaction effects forfeited; constraint handling fixed a priori	Mixed-variable formulation with mapping as first-class decisions; platform-specific cost models; constraint-handling choice as co-design decision

Challenge 1 (Surrogate-assisted optimization for expensive evaluations). The gap between hardware-in-the-loop fidelity and simulation-based affordability can only be bridged by principled surrogate modeling, yet the surveyed studies employ ad hoc regressors with little attention to staleness, bias, or uncertainty calibration. The data-driven evolutionary optimization literature offers ready machinery, instance selection, model management, and trust-region evolution, whose porting to neuromorphic objectives is straightforward in principle [93]. Success metrics: Simulator-hardware correlation reported as a matter of course; surrogate calibration diagnostics alongside optimization results; and demonstrated equal-quality solutions at a stated fraction (a reasonable initial target: One tenth) of the hardware evaluation count. A meaningful milestone would be a public benchmark of (configuration, measured fitness) pairs for a fixed platform and workload, enabling offline surrogate comparison, an asset the community currently lacks.

Challenge 2 (AOS and parameter control). The budget-conditioned superiority documented in Section 5.3 implies that static operator and parameter choices are structurally mismatched to neuromorphic landscapes, whose character changes across regions of the search space and phases of the search. The adaptation machinery of the DE lineage [45-48], the AOS framework [81], and the parameter-control taxonomy [82], [83] port naturally; the open question is whether adaptation decisions learned on one platform or workload transfer to others, since the value of adaptation lies precisely in transferability. Success metrics: Cross-instance robustness (rank stability across workload subsets) rather than single-instance fitness; adaptation overhead reported as a fraction of evaluation budget.

Challenge 3 (Standardized neuromorphic benchmark suites). No community standard yet fixes the workloads, platform contracts, budget scales, and reporting format that an architecture-optimization comparison must include, leaving every comparison confined to individual studies. Building blocks exist: The numerical layer is well served by CEC suites [77], [78] and budget-resolved measurement [22]; the workload layer has consolidated datasets, N-MNIST [87], DVS128 Gesture [88], and the Heidelberg spiking datasets [89]; and NeuroBench [95] together with the Loihi 2 evaluation stack [30] signals community appetite. Recommended actions: Define a tiered suite spanning numerical foundations and domain workloads with published platform contracts (resource limits, cost models) for TrueNorth-, Loihi-, and SpiNNaker-class substrates; and mandate budget and tuning disclosure as a condition of benchmark compliance.

Challenge 4 (Reproducibility and fair comparison). Within-study tuning, selective reporting, and unreported budgets collectively undermine the statistical machinery of Section 5.2 [14], [79]. Recommended actions: Pre-register comparison protocols, including per-method tuning budgets; release configurations, seeds, and raw measurements with publications; adopt the Wilcoxon-Friedman-critical-difference protocol

[18], [20], [21], [90] as a reporting default, supplemented by budget-resolved performance curves in the COCO manner [22]; and state statistical power alongside significance. Open-source toolchains, Lava [30], snnTorch [7], and their peers, lower the marginal cost of compliance and should be leveraged as reproducibility vehicles.

Challenge 5 (Algorithm–hardware co-design). The prevailing workflow optimizes a network abstraction and then maps it onto hardware, forfeiting interaction effects that dominate energy and latency (Section 6.2). Recommended actions: formulate co-design problems in the mixed-variable vocabulary of Section 2.4, with mapping variables as first-class decision variables; exploit platform-specific cost models where available [7]; and treat constraint-handling choice (Section 3.2) as a co-design decision rather than a fixed implementation detail [37], [38], [42]. A measurable milestone: Published instances where joint optimization demonstrates statistically significant improvement over the best decoupled pipeline under the protocol of Section 5.2, the controlled experiment that Section 6.2 notes is currently missing.

These challenges are interlocking in a precise sense: Standardized benchmarks (*Challenge 3*) enable the fair comparisons that reproducibility practices (*Challenge 4*) certify, which in turn generate the evidence base on which surrogate methods (*Challenge 1*) and adaptive schemes (*Challenge 2*) can be validated, while co-design (*Challenge 5*) supplies the problem formulations that give all four their applied significance. Progress on any subset is therefore amplified by progress on the remainder.

8 | Conclusion

This survey has synthesized the literature on metaheuristic optimization for neuromorphic computing architectures across its methodological, hardware, and application layers, organizing 95 sources into a problem-oriented taxonomy built on an explicit mathematical formulation of the configuration problem and a platform-by-platform account of the resource contracts that shape it. Three conclusions summarize the evidence: 1) the largest attributable performance gains trace to operator-level innovations, parameter adaptation, external archives, and success-history learning, concentrated in the evolution-based lineage [36], [45–48], whereas the metaphor-driven stream has added volume more than validated capability [14], [15], 2) the structural properties that make the problem hard, multimodality, mixed discrete–continuous variables, expensive noisy evaluations, and platform-indexed coupled constraints, are precisely the properties that metaheuristics tolerate and that classical methods do not, which explains and justifies the field's methodological orientation [10], [11]; the simultaneous maturation of differentiable architecture search and Bayesian optimization sharpens, rather than weakens, the case for principled metaheuristics in the regimes where discreteness and population diversity dominate [39], [41], 3) the evidentiary foundation of the field is limited by protocol heterogeneity, unreported tuning, sparse intersection studies, and the absence of standardized neuromorphic benchmarks; until these deficiencies are remedied, cross-study rankings should be interpreted as conditional statements about specific budget and platform regimes rather than as general truths [21], [22], [79].

The roadmap of Section 7 charts the path from the present state to a cumulative science of neuromorphic architecture optimization: Surrogate-assisted methods to reconcile fidelity with affordability, AOS to match the time-varying structure of the landscape, standardized benchmarks and reproducibility safeguards to make comparisons meaningful, and algorithm–hardware co-design to realize the energy-efficiency promise that motivates the enterprise. What the evidence ultimately supports is a redirection of effort: Away from the proposal of further algorithmic metaphors and toward the methodological infrastructure, benchmarks, protocols, surrogate machinery, and co-designed formulations, on which the next decade of genuine progress in energy per inference, synaptic operations, and spike count will have to be built [2], [3], [5–8].

Authors' Contributions

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

Funding

Not applicable.

Data Availability

No new data were created or analyzed in this study. All sources synthesized in this survey are published works cited in the reference list; the bibliographic corpus underlying *Fig. 1* is fully specified by that list.

Conflict of Interest

The authors declare no conflict of interest.

Consent for Publication

The author confirms consent for the publication of this work.

Ethics Approval and Consent to Participate

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

References

- [1] Maass, W. (1997). Networks of spiking neurons: The third generation of neural network models. *Neural Networks*, 10(9), 1659–1671. [https://doi.org/10.1016/S0893-6080\(97\)00011-7](https://doi.org/10.1016/S0893-6080(97)00011-7)
- [2] Roy, K., Jaiswal, A., & Panda, P. (2019). Towards spike-based machine intelligence with neuromorphic computing. *Nature*, 575(7784), 607–617. <https://doi.org/10.1038/s41586-019-1677-2>
- [3] Merolla, P. A., Arthur, J. V., Alvarez-Icaza, R., Cassidy, A. S., Sawada, J., Akopyan, F., ... & Modha, D. S. (2014). A million spiking-neuron integrated circuit with a scalable communication network and interface. *Science*, 345(6197), 668–673. <https://doi.org/10.1126/science.1254642>
- [4] Esser, S. K., Appuswamy, R., Merolla, P., Arthur, J. V., & Modha, D. S. (2015). Backpropagation for energy-efficient neuromorphic computing. *Advances in Neural Information Processing Systems*, 28. <https://doi.org/10.1073/pnas.1604850113>
- [5] Davies, M., Srinivasa, N., Lin, T. H., China, G., Cao, Y., Choday, S. H., ... & Wang, H. (2018). Loihi: A neuromorphic manycore processor with on-chip learning. *IEEE Micro*, 38(1), 82–99. <https://doi.org/10.1109/MM.2018.112130359>
- [6] Furber, S. (2016). Large-scale neuromorphic computing systems. *Journal of Neural Engineering*, 13(5), 51001. <https://doi.org/10.1088/1741-2560/13/5/051001>
- [7] Eshraghian, J. K., Ward, M., Neftci, E. O., Wang, X., Lenz, G., Dwivedi, G., ... & Lu, W. D. (2023). Training spiking neural networks using lessons from deep learning. *Proceedings of the IEEE*, 111(9), 1016–1054. <https://doi.org/10.1109/JPROC.2023.3308088>
- [8] Kosta, A. K., Roy, K., Sengupta, A., Chakraborty, I., Panda, P., Rathi, N., & Ankit, A. (2021). Exploring neuromorphic computing based on spiking neural networks: Algorithms to hardware. *ACM Computing Surveys*, 55(12), 1–49. <https://doi.org/10.1145/3571155>
- [9] Boussaïd, I., Lepagnot, J., & Siarry, P. (2013). A survey on optimization metaheuristics. *Information Sciences*, 237, 82–117. <https://doi.org/10.1016/j.ins.2013.02.041>
- [10] Mirjalili, S., Dong, J. S., & Lewis, A. (2020). Nature-inspired optimizers. *Studies in Computational Intelligence*, 811, 7–20. <https://doi.org/10.1007/978-3-030-12127-3>
- [11] Osman, I. H., & Laporte, G. (1996). Metaheuristics: A bibliography. *Annals of Operations Research*, 63(5), 511–623. <https://doi.org/10.1007/BF02125421>
- [12] Davis, L. (1991). *Handbook of genetic algorithms*. Van Nostrand Reinhold. <https://www.semanticscholar.org/paper/Handbook-Of-Genetic-Algorithms-Davis/54acdb67ca083326c34eabdeb59bfdc01c748df0>

- [13] Stanley, K. O., & Miikkulainen, R. (2002). Evolving neural networks through augmenting topologies. *Evolutionary Computation*, 10(2), 99–127. <https://doi.org/10.1162/106365602320169811>
- [14] Sörensen, K. (2015). Metaheuristics—The metaphor exposed. *International Transactions in Operational Research*, 22(1), 3–18. <https://doi.org/10.1111/itor.12001>
- [15] Campelo, F., & Aranha, C. (2023). Lessons from the evolutionary computation bestiary. *Artificial Life*, 29(4), 421–432. https://doi.org/10.1162/artl_a_00402
- [16] Wolpert, D. H., & Macready, W. G. (1997). No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1(1), 67–82. <https://doi.org/10.1109/4235.585893>
- [17] Droste, S., Jansen, T., & Wegener, I. (2002). Optimization with randomized search heuristics—The (A) NFL theorem, realistic scenarios, and difficult functions. *Theoretical Computer Science*, 287(1), 131–144. [https://doi.org/10.1016/S0304-3975\(02\)00094-4](https://doi.org/10.1016/S0304-3975(02)00094-4)
- [18] Wilcoxon, F. (1945). Individual comparisons by ranking methods. *Biometrics Bulletin*, 1(6), 80–83. <https://doi.org/10.2307/3001968>
- [19] Holm, S. (1979). A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, 6(2), 65–70. <https://www.jstor.org/stable/4615733>
- [20] Friedman, M. (1937). The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of the American Statistical Association*, 32(200), 675–701. <https://www.tandfonline.com/doi/pdf/10.1080/01621459.1937.10503522>
- [21] Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7(Jan), 1–30. <https://www.jmlr.org/papers/volume7/demsar06a/demsar06a.pdf>
- [22] Hansen, N., Auger, A., Ros, R., Mersmann, O., Tušar, T., & Brockhoff, D. (2021). COCO: A platform for comparing continuous optimizers in a black-box setting. *Optimization Methods and Software*, 36(1), 114–144. <https://doi.org/10.1080/10556788.2020.1808977>
- [23] Bohte, S. M., Kok, J. N., & La Poutre, H. (2002). Error-backpropagation in temporally encoded networks of spiking neurons. *Neurocomputing*, 48(1–4), 17–37. [https://doi.org/10.1016/S0925-2312\(01\)00658-0](https://doi.org/10.1016/S0925-2312(01)00658-0)
- [24] Rueckauer, B., Lungu, I. A., Hu, Y., Pfeiffer, M., & Liu, S. C. (2017). Conversion of continuous-valued deep networks to efficient event-driven networks for image classification. *Frontiers in Neuroscience*, 11, 682. <https://doi.org/10.3389/fnins.2017.00682>
- [25] Bi, G., & Poo, M. (1998). Synaptic modifications in cultured hippocampal neurons: Dependence on spike timing, synaptic strength, and postsynaptic cell type. *The Journal of Neuroscience*, 18(24), 10464–10472. <https://doi.org/10.1523/JNEUROSCI>
- [26] Song, S., Miller, K. D., & Abbott, L. F. (2000). Competitive Hebbian learning through spike-timing-dependent synaptic plasticity. *Nature Neuroscience*, 3(9), 919–926. <https://doi.org/10.1038/78829>
- [27] Diehl, P. U., & Cook, M. (2015). Unsupervised learning of digit recognition using spike-timing-dependent plasticity. *Frontiers in Computational Neuroscience*, 9, 99. <https://doi.org/10.3389/fncom.2015.00099>
- [28] Qiao, N., Mostafa, H., Corradi, F., Osswald, M., Stefanini, F., Sumislawska, D., & Indiveri, G. (2015). A reconfigurable on-line learning spiking neuromorphic processor comprising 256 neurons and 128K synapses. *Frontiers in Neuroscience*, 9, 141. <https://doi.org/10.3389/fnins.2015.00141>
- [29] Pehle, C., Billaudelle, S., Cramer, B., Kaiser, J., Schreiber, K., Stradmann, Y., ... & Schemmel, J. (2022). The BrainScaleS-2 accelerated neuromorphic system with hybrid plasticity. *Frontiers in Neuroscience*, 16, 795876. <https://doi.org/10.3389/fnins.2022.795876>
- [30] Davies, M., Wild, A., Orchard, G., Sandamirskaya, Y., Guerra, G. A. F., Joshi, P., ... & Risbud, S. R. (2021). Advancing neuromorphic computing with Loihi: A survey of results and outlook. *Proceedings of the IEEE*, 109(5), 911–934. <https://doi.org/10.1109/JPROC.2021.2507580>
- [31] Schemmel, J., Brüderle, D., Grübl, A., Hock, M., Meier, K., & Millner, S. (2010). A wafer-scale neuromorphic hardware system for large-scale neural modeling. *2010 IEEE International Symposium on Circuits and Systems (ISCAS)* (pp. 1947–1950). IEEE. <https://doi.org/10.1109/TBCAS.2010.2053773>
- [32] Benjamin, B. V., Gao, P., McQuinn, E., Choudhary, S., Chandrasekaran, A. R., Bussat, J. M., ... & Boahen, K. (2014). Neurogrid: A mixed-analog-digital multichip system for large-scale neural simulations. *Proceedings of the IEEE*, 102(5), 699–716. <https://doi.org/10.1109/JPROC.2014.2313565>

- [33] Pei, J., Deng, L., Song, S., Zhao, M., Zhang, Y., Wu, S., Wang, G., Zou, Z., Wu, Z., He, W., ... & Shi, L. (2019). Towards artificial general intelligence with hybrid Tianjic chip architecture. *Nature*, 572(7767), 106–111. <https://doi.org/10.1038/s41586-019-1424-8>
- [34] Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2), 182–197. <https://doi.org/10.1109/4235.996017>
- [35] Zhang, Q., & Li, H. (2007). MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on Evolutionary Computation*, 11(6), 712–731. <https://doi.org/10.1109/TEVC.2007.892759>
- [36] Hansen, N., & Ostermeier, A. (2001). Completely derandomized self-adaptation in evolution strategies. *Evolutionary Computation*, 9(2), 159–195. <https://doi.org/10.1162/106365601750190398>
- [37] Deb, K. (2000). An efficient constraint handling method for genetic algorithms. *Computer Methods in Applied Mechanics and Engineering*, 186(2–4), 311–338. [https://doi.org/10.1016/S0045-7825\(99\)00389-8](https://doi.org/10.1016/S0045-7825(99)00389-8)
- [38] Coello, C. A. C. (2002). Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: A survey of the state of the art. *Computer Methods in Applied Mechanics and Engineering*, 191(11–12), 1245–1287. [https://doi.org/10.1016/S0045-7825\(01\)00323-1](https://doi.org/10.1016/S0045-7825(01)00323-1)
- [39] Liu, H., Simonyan, K., & Yang, Y. (2018). *Darts: Differentiable architecture search*. <https://doi.org/10.48550/arXiv.1806.09055>
- [40] Tan, M., Chen, B., Pang, R., Vasudevan, V., Sandler, M., Howard, A., & Le, Q. V. (2019). Mnasnet: Platform-aware neural architecture search for mobile. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 2815–2823). IEEE. <https://doi.org/10.1109/CVPR.2019.00293>
- [41] Snoek, J., Larochelle, H., & Adams, R. (2012). Practical bayesian optimization of machine learning algorithms. *Advances in Neural Information Processing Systems*, 25. https://proceedings.neurips.cc/paper_files/paper/2012/hash/05311655a15b75fab86956663e1819cd-Abstract.html
- [42] Mezura-Montes, E., & Coello, C. A. C. (2011). Constraint-handling in nature-inspired numerical optimization: Past, present and future. *Swarm and Evolutionary Computation*, 1(4), 173–194. <https://doi.org/10.1016/j.swevo.2011.10.001>
- [43] Holland, J. H. (1975). *Adaptation in natural and artificial systems: An introductory analysis with applications to biology*. University of Michigan Press. <https://doi.org/10.7551/mitpress/1090.001.0001>
- [44] Storn, R., & Price, K. (1997). Differential evolution-A simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 11(4), 341–359. <https://doi.org/10.1023/A:1008202821328>
- [45] Qin, A. K., Huang, V. L., & Suganthan, P. N. (2008). Differential evolution algorithm with strategy adaptation for global numerical optimization. *IEEE Transactions on Evolutionary Computation*, 13(2), 398–417. <https://doi.org/10.1109/TEVC.2008.927706>
- [46] Zhang, J., & Sanderson, A. C. (2009). JADE: Adaptive differential evolution with optional external archive. *IEEE Transactions on Evolutionary Computation*, 13(5), 945–958. <https://doi.org/10.1109/TEVC.2009.2014613>
- [47] Tanabe, R., & Fukunaga, A. (2013). Success-history based parameter adaptation for differential evolution. *2013 IEEE Congress on Evolutionary Computation* (pp. 71–78). IEEE. <https://doi.org/10.1109/CEC.2013.6557555>
- [48] Tanabe, R., & Fukunaga, A. S. (2014). Improving the search performance of SHADE using linear population size reduction. *2014 IEEE Congress on Evolutionary Computation (CEC)* (pp. 1658–1665). IEEE. <https://doi.org/10.1109/CEC.2014.6900380>
- [49] Simon, D. (2008). Biogeography-based optimization. *IEEE Transactions on Evolutionary Computation*, 12(6), 702–713. <https://doi.org/10.1109/TEVC.2008.919004>
- [50] Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. *Proceedings of ICNN'95-International Conference on Neural Networks* (Vol. 4, pp. 1942–1948). IEEE. <https://doi.org/10.1109/ICNN.1995.488968>
- [51] Dorigo, M., Maniezzo, V., & Coloni, A. (1996). Ant system: Optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics—Part B: Cybernetics*, 26(1), 29–41. <https://doi.org/10.1109/3477.484436>

- [52] Karaboga, D., & Basturk, B. (2007). A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm. *Journal of Global Optimization*, 39(3), 459–471. <https://doi.org/10.1007/s10898-007-9149-x>
- [53] Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey wolf optimizer. *Advances in Engineering Software*, 69, 46–61. <https://doi.org/10.1016/j.advengsoft.2013.12.007>
- [54] Mirjalili, S., & Lewis, A. (2016). The whale optimization algorithm. *Advances in Engineering Software*, 95, 51–67. <https://doi.org/10.1016/j.advengsoft.2016.01.008>
- [55] Mirjalili, S. (2015). The ant lion optimizer. *Advances in Engineering Software*, 83, 80–98. <https://doi.org/10.1016/j.advengsoft.2015.01.010>
- [56] Mirjalili, S. (2015). Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm. *Knowledge-Based Systems*, 89, 228–249. <https://doi.org/10.1016/j.knsys.2015.07.006>
- [57] Mirjalili, S. (2016). SCA: A sine cosine algorithm for solving optimization problems. *Knowledge-Based Systems*, 96, 120–133. <https://doi.org/10.1016/j.knsys.2015.12.022>
- [58] Mirjalili, S. (2016). Dragonfly algorithm: A new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems. *Neural Computing and Applications*, 27(4), 1053–1073. <https://doi.org/10.1007/s00521-015-1920-1>
- [59] Mirjalili, S., Gandomi, A. H., Mirjalili, S. Z., Saremi, S., Faris, H., & Mirjalili, S. M. (2017). Salp swarm algorithm: A bio-inspired optimizer for engineering design problems. *Advances in Engineering Software*, 114, 163–191. <https://doi.org/10.1016/j.advengsoft.2017.07.002>
- [60] Saremi, S., Mirjalili, S., & Lewis, A. (2017). Grasshopper optimisation algorithm: Theory and application. *Advances in Engineering Software*, 105, 30–47. <https://doi.org/10.1016/j.advengsoft.2017.01.004>
- [61] Heidari, A. A., Mirjalili, S., Faris, H., Aljarah, I., Mafarja, M., & Chen, H. (2019). Harris Hawks optimization: Algorithm and applications. *Future Generation Computer Systems*, 97, 849–872. <https://doi.org/10.1016/j.future.2019.02.028>
- [62] Passino, K. M. (2002). Biomimicry of bacterial foraging for distributed optimization and control. *IEEE Control Systems Magazine*, 22(3), 52–67. <https://doi.org/10.1109/MCS.2002.1004010>
- [63] Yang, X. S., & Deb, S. (2009). Cuckoo search via Lévy flights. *2009 World Congress on Nature & Biologically Inspired Computing (NaBIC)* (pp. 210–214). IEEE. <https://doi.org/10.1109/NABIC.2009.5393690>
- [64] Abualigah, L., Yousri, D., Abd Elaziz, M., Ewees, A. A., Al-Qaness, M. A. A., & Gandomi, A. H. (2021). Aquila optimizer: A novel meta-heuristic optimization algorithm. *Computers & Industrial Engineering*, 157, 107250. <https://doi.org/10.1016/j.cie.2021.107250>
- [65] Alimoradi, M., Azgomi, H., & Asghari, A. (2022). Trees social relations optimization algorithm: A new swarm-based metaheuristic technique to solve continuous and discrete optimization problems. *Mathematics and Computers in Simulation*, 194, 629–664. <https://doi.org/10.1016/j.matcom.2021.12.010>
- [66] Kirkpatrick, S., Gelatt, C. D., Jr., & Vecchi, M. P. (1983). Optimization by simulated annealing. *Science*, 220(4598), 671–680. <https://doi.org/10.1126/science.220.4598.671>
- [67] Rashedi, E., Nezamabadi-Pour, H., & Saryazdi, S. (2009). GSA: A gravitational search algorithm. *Information Sciences*, 179(13), 2232–2248. <https://doi.org/10.1016/j.ins.2009.03.004>
- [68] Yang, X. S. (2009). Firefly algorithms for multimodal optimization. *International Symposium on Stochastic Algorithms* (pp. 169–178). Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-04944-6_14
- [69] Yang, X. S. (2010). A new metaheuristic bat-inspired algorithm. In *Nature Inspired Cooperative Strategies for Optimization (NICSO)* (pp. 65–74). Springer. https://doi.org/10.1007/978-3-642-12538-6_6
- [70] Yang, X. S. (2012). Flower pollination algorithm for global optimization. *International Conference on Unconventional Computing and Natural Computation* (pp. 240–249). Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-32894-7_27
- [71] Eskandar, H., Sadollah, A., Bahreininejad, A., & Hamdi, M. (2012). Water cycle algorithm-A novel metaheuristic optimization method for solving constrained engineering optimization problems. *Computers & Structures*, 110, 151–166. <https://doi.org/10.1016/j.compstruc.2012.07.010>

- [72] Faramarzi, A., Heidarinejad, M., Stephens, B., & Mirjalili, S. (2020). Equilibrium optimizer: A novel optimization algorithm. *Knowledge-Based Systems*, 191, 105190. <https://doi.org/10.1016/j.knsys.2019.105190>
- [73] Mirjalili, S., Mirjalili, S. M., & Hatamlou, A. (2016). Multi-verse optimizer: A nature-inspired algorithm for global optimization. *Neural Computing and Applications*, 27(2), 495–513. <https://doi.org/10.1007/s00521-015-1870-7>
- [74] Al-Madi, N., Faris, H., & Mirjalili, S. (2019). Binary multi-verse optimization algorithm for global optimization and discrete problems. *International Journal of Machine Learning and Cybernetics*, 10(12), 3445–3465. <https://doi.org/10.1007/s13042-019-00931-8>
- [75] Daliri, A., Asghari, A., Azgomi, H., & Alimoradi, M. (2022). The water optimization algorithm: A novel metaheuristic for solving optimization problems. *Applied Intelligence*, 52(15), 17990–18029. <https://doi.org/10.1007/s10489-022-03397-4>
- [76] Rao, R. V., Savsani, V. J., & Vakharia, D. P. (2011). Teaching-learning-based optimization: A novel method for constrained mechanical design optimization problems. *Computer-Aided Design*, 43(3), 303–315. <https://doi.org/10.1016/j.cad.2010.12.015>
- [77] Liang, J. J., Qu, B. Y., & Suganthan, P. N. (2013). Problem definitions and evaluation criteria for the CEC 2014 special session and competition on single objective real-parameter numerical optimization. *Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou China and Technical Report, Nanyang Technological University, Singapore*, 635(2), 2014. [https://bee22.com/manual/tf_images/Liang CEC2014.pdf](https://bee22.com/manual/tf_images/Liang%20CEC2014.pdf)
- [78] Kumar, A., Price, K. V., Mohamed, A. W., Hadi, A. A., & Suganthan, P. N. (2021). *Problem definitions and evaluation criteria for the CEC 2022 special session and competition on single objective bound constrained numerical optimization*. Nanyang Technological University. file:///C:/Users/Admin/Downloads/CEC2022 TR (1).pdf
- [79] Werth, B., Pitzer, E., Ostermayer, G., & Affenzeller, M. (2018). Quasi-bistability of walk-based landscape measures in stochastic fitness landscapes. *Proceedings of the Genetic and Evolutionary Computation Conference* (pp. 1087-1094). Association for Computing Machinery (ACM). <https://doi.org/10.1145/3205455.3205471>
- [80] Črepinšek, M., Liu, S. H., & Mernik, M. (2013). Exploration and exploitation in evolutionary algorithms: A survey. *ACM Computing Surveys (CSUR)*, 45(3), 1–33. <https://doi.org/10.1145/2480741.2480752>
- [81] Pham, H. N. A., Shr, A., Chen, P. P., & Liu, A. (2008). Scheduling for dedicated machine constraint using integer programming. *2008 20th IEEE International Conference on Tools with Artificial Intelligence* (Vol. 1, pp. 499-506). IEEE. <https://doi.org/10.1109/ICTAI.2008.85>
- [82] Eiben, Á. E., Hinterding, R., & Michalewicz, Z. (1999). Parameter control in evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, 3(2), 124–141. <https://doi.org/10.1109/4235.771166>
- [83] Karafotias, G., Hoogendoorn, M., & Eiben, Á. E. (2014). Parameter control in evolutionary algorithms: Trends and challenges. *IEEE Transactions on Evolutionary Computation*, 19(2), 167–187. <https://doi.org/10.1109/TEVC.2014.2308294>
- [84] Rice, J. R. (1976). The algorithm selection problem. In *Advances in computers* (Vol. 15, pp. 65–118). Elsevier. [https://doi.org/10.1016/S0065-2458\(08\)60520-3](https://doi.org/10.1016/S0065-2458(08)60520-3)
- [85] Muñoz, M. A., Sun, Y., Kirley, M., & Halgamuge, S. K. (2015). Algorithm selection for black-box continuous optimization problems: A survey on methods and challenges. *Information Sciences*, 317, 224–245. <https://doi.org/10.1016/j.ins.2015.05.010>
- [86] Kerschke, P., Hoos, H. H., Neumann, F., & Trautmann, H. (2019). Automated algorithm selection: Survey and perspectives. *Evolutionary Computation*, 27(1), 3–45. https://doi.org/10.1162/evco_a_00242
- [87] Orchard, G., Jayawant, A., Cohen, G. K., & Thakor, N. (2015). Converting static image datasets to spiking neuromorphic datasets using saccades. *Frontiers in Neuroscience*, 9, 437. <https://doi.org/10.3389/fnins.2015.00437>
- [88] Amir, A., Taba, B., Berg, D., Melano, T., McKinstry, J., Di Nolfo, C., ... & Modha, D. (2017). A low power, fully event-based gesture recognition system. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 7388-7397). IEEE. <https://doi.org/10.1109/CVPR.2017.781>
- [89] Cramer, B., Stradmann, Y., Schemmel, J., & Zenke, F. (2020). The heidelberg spiking data sets for the systematic evaluation of spiking neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 33(7), 2744–2757. <https://doi.org/10.1109/TNNLS.2020.3044364>

- [90] Derrac, J., García, S., Molina, D., & Herrera, F. (2011). A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm and Evolutionary Computation*, 1(1), 3-18. <https://doi.org/10.1016/j.swevo.2011.02.002>
- [91] Sobol, I. M. (2001). Global sensitivity indices for nonlinear mathematical models and their Monte Carlo estimates. *Mathematics and Computers in Simulation*, 55(1-3), 271-280. [https://doi.org/10.1016/S0378-4754\(00\)00270-6](https://doi.org/10.1016/S0378-4754(00)00270-6)
- [92] Anderson-Cook, C. M. (2005). Experimental and quasi-experimental designs for generalized causal inference. *Journal of the American Statistical Association*, 100(470), 708. <https://doi.org/10.1198/jasa.2005.s22>
- [93] Jin, Y., Wang, H., Chugh, T., Guo, D., & Miettinen, K. (2018). Data-driven evolutionary optimization: An overview and case studies. *IEEE Transactions on Evolutionary Computation*, 23(3), 442-458. <https://doi.org/10.1109/TEVC.2018.2869001>
- [94] Zitzler, E., & Thiele, L. (1999). Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach. *IEEE Transactions on Evolutionary Computation*, 3(4), 257-271. <https://doi.org/10.1109/4235.797969>
- [95] Yik, J., Van den Berghe, K., Den Blanken, D., Bouhadjar, Y., Fabre, M., Hueber, P., ... & Reddi, V. J. (2025). The neurobench framework for benchmarking neuromorphic computing algorithms and systems. *Nature Communications*, 16(1), 1545. <https://doi.org/10.1038/s41467-025-56739-4>