Towards Neuromorphic Evolutionary Computation: A Position Paper

Jorge M. Cruz-Duarte and El-Ghazali Talbi
University of Lille, CNRS, Inria, Centrale Lille, UMR 9189 CRIStAL, F-59000 Lille, France.
E-Mails: {jorge.cruz-duarte, el-ghazali.talbi}@univ-lille.fr

***Abstract*—Neuromorphic Computing (NC) and Evolutionary Computation (EC) are mature, yet largely fragmented. This position paper argues that their convergence is timely and technically plausible. We define Neuromorphic Evolutionary Computation (NEC) as a paradigm that bridges NC and EC. We review the current landscape and its limitations, with an emphasis on the prevalence of hybrid execution and incomplete energy-latency reporting. We then present a probable Cortico-Basal-Thalamic Loop (CBTL)-inspired conceptual architecture for operator selection and outline what must be implemented on-chip to achieve an NEC approach, including operator embodiment, associative memory, and state estimation. We summarise open challenges spanning numerical precision, scalability, learning rules, benchmarking, theory, and tooling, including the need for convergence analysis framed in terms of the best-so-far process. We close with a phased roadmap and concrete reporting requirements towards a reproducible NEC ecosystem.**

***Index Terms*—Neuromorphic Evolutionary Computation, Event-driven Optimisation, Neuromorphic Computing, Spiking Neural Networks, Operator Selection, Energy-Latency Benchmarking, Best-so-far Convergence**

I. Introduction

ARTIFICIAL intelligence has expanded its reach across scientific, industrial, and domestic domains. It is part of contemporary human life. However, this progress has intensified the energy demands associated with large-scale computation. Training contemporary foundation models requires extensive electrical resources, and deploying neural networks on edge platforms imposes strict constraints on battery-powered systems. This situation reflects a structural limitation rather than an operational inconvenience, as data centres already account for a significant share of global electricity consumption and are projected to continue growing without a decisive architectural change. The limitation originates from the von Neumann bottleneck, in which computation and memory remain physically separated, and energy is dominated by data movement rather than arithmetic operations. Iterative black box optimisation methods, including the population-based processes central to Evolutionary Computation (EC), are susceptible to this constraint [1,2]. Fitness evaluation, state manipulation, and population updates require repeated memory accesses and synchronisation, which amplify energy dissipation and limit scalability on conventional hardware.

EC remains central to Black-Box Optimisation (BBO), although its deployment on conventional hardware exposes constraints that extend beyond algorithmic design. EC provides effective search behaviour in domains where gradients are inaccessible, or objectives are irregular [2,3]. Still, their theoretical support remains limited, their performance deteriorates in high-dimensional spaces, and their execution incurs synchronisation costs associated with processors optimised for dense linear algebra.

The structural mismatch between EAs and conventional hardware motivates the study of alternative computational substrates. Neuromorphic Computing (NC) processors such as Intel Loihi [4], IBM TrueNorth [5], SpiNNaker [6], BrainScaleS [7], SynSense Xylo [8], and BrainChip Akida [9] integrate memory and computation, communicate via sparse asynchronous spikes, and operate in an event-driven regime rather than through dense synchronous updates [10]. These architectures achieve substantial energy efficiency. For instance, Loihi 2 operates within a power envelope of a few hundred milliwatts for networks of millions of neurons, far below the requirements of CPU-based simulation [10]. This efficiency derives from reduced data movement and the exploitation of sparse, asynchronous neural activity.

Neuromorphic hardware has matured to the point at which stable software support and commercial availability are established. Its applications now extend beyond neural modelling to embedded inference [11,12]. However, optimisation remains almost entirely unexplored within this class of processors [13].

These developments motivate our central thesis. The convergence of EC with neuromorphic hardware constitutes a strategic opportunity for both domains. NC substrates offer sub-watt operation, intrinsic parallelism through distributed spiking ensembles, and online adaptability driven by synaptic plasticity, enabling real-time optimisation within the strict energy budgets of embedded platforms. Conversely, EC provides NC systems with a rigorous collection of black box benchmarks, concrete application scenarios beyond inference tasks, and a set of theoretical challenges concerning convergence, approximation capacity, and computational complexity under event-driven dynamics. Therefore, Neuromorphic Evolutionary Computation (NEC) should be treated as a first-class topic within the EC community. This is timely in the era of large-scale and expensive real-world optimisation. It is also relevant to sustainability, energy efficiency, and the environmental impact of EC methods and applications. Moreover, edge and embedded devices are a key target. They impose tight energy budgets and strict latency constraints. NEC fits

NEVO is part of the NeurOptim research initiative: https://neuroptim.org.

this setting. It supports event-driven, local, and asynchronous decision making on-device. NEC may also intersect with quantum computing. Hybrid workflows are plausible. A quantum backend could support specific subroutines, while spike-driven dynamics provide coordination and control.

This position paper makes three contributions. First, NEC is defined and distinguished from related work. Second, a CBTL-inspired reference design for operator selection is outlined. Third, the open challenges are organised, and a phased roadmap is proposed, with explicit evaluation criteria. These contributions require consistent reporting. For this reason, NEC studies should follow simple design principles.

- Event-driven first: use sparse, asynchronous updates. Avoid global synchronisation.
- On-device state: encode the search state in spikes. Minimise host-side control.
- Measured trade-offs: report solution quality, energy, and latency together.
- Composable operators: use interfaces that support composition and ablation.
- Reproducibility: release code, seeds, and profiling scripts.

The remainder of the paper reviews the current landscape, introduces a reference architecture for operator selection, summarises open challenges, and outlines a roadmap towards fully neuromorphic optimisation.

II. Current Landscape and Limitations

Research at the intersection of NC and optimisation remains fragmented. Existing contributions show that spike-driven dynamics can support basic search behaviour. However, the literature remains difficult to compare. TABLE I synthetises the main strands and their recurring limitations. The current landscape and limitations are discussed below.

TABLE I
Landscape of neuromorphic optimisation work. It summarises representative strands and the recurring limitations.

Strand	**Typical limitation (examples)**
Problem-specific solvers	Efficient, but tightly coupled to a problem class or device primitive. Extending to new objectives often requires redesign [14]–[17].
Neuroevolution of SNNs	Search runs on CPUs. Neuromorphic hardware is used for inference or evaluation. Energy gains occur after deployment [18]–[20].
Spiking swarm models	Often simulation-only. Hardware constraints and energy are rarely measured. Portability across platforms is unclear [21,22].
Hybrid frameworks	Control may be spiking, but key steps remain conventional. Candidate generation or fitness evaluation often stays off-chip [13,23]–[25].

A first group of studies embeds optimisation directly in physical or device-specific neuromorphic substrates. Examples include ferroelectric memristor networks for the Travelling Salesperson Problem [14], organic-memristor Max-Cut solvers [15], and QUBO-oriented implementations based on RRAM-based spiking swarms [16] or Simulated Annealing on Loihi 2 [17]. These systems are tightly coupled to the characteristics of the target problem or hardware. They achieve efficiency but offer limited adaptability, since extending them to continuous or multi-objective settings typically requires redesigning the neural architecture rather than adjusting parameters. In parallel, recent neuromorphic vision and control work highlights event-driven sensing and algorithm-hardware co-design for energy-efficient edge intelligence, but still focuses on perception rather than black-box optimisation [12].

A second line of work concerns the evolutionary design of Spiking Neural Networks (SNNs) rather than optimisation executed within neuromorphic substrates. Approaches such as the Evolutionary Optimisation for NC Systems framework, along with surveys of evolutionary SNNs and biologically plausible learning rules, demonstrate that Evolutionary Algorithms (EAs) can effectively configure SNNs [18]–[20]. However, the optimisation process itself takes place on conventional processors, and neuromorphic hardware is used only for inference. Energy gains arise after evolution, not during search. Therefore, these approaches fall outside the scope of neuromorphic optimisation considered in this work.

Swarm-inspired models provide a complementary perspective. Spiking adaptations of swarm mechanisms have been proposed for both continuous and combinatorial settings [21], and oscillator-based variants of Particle Swarm Optimisation (PSO) offer formal convergence guarantees [22]. Yet these systems are typically evaluated in software simulations and do not exploit the architectural advantages of neuromorphic hardware, namely event-driven execution, distributed memory, and on-chip plasticity. As a result, they shed light on how swarm behaviour maps to spiking dynamics but do not address the practical requirements of NC-implemented optimisers.

Probabilistic optimisation has also been explored through neuromorphic interfaces. LavaBO and its asynchronous variant demonstrate that NC platforms can coordinate Bayesian optimisation pipelines [23,24]. However, the core components, including Gaussian process models and acquisition optimisation, remain CPU-based. The SNNs act mainly as scheduling and communication layers. The approach is practical but still hybrid, leaving open whether the optimisation logic itself can be realised within spike-driven computation.

More recently, Talbi's Nheuristics framework [13] has offered a principled terminology for neuromorphic heuristic search, identifying neuron models, encoding schemes, learning rules, and communication structures as a shared design space. Building on this, the NeurOptimiser framework [25] has illustrated how several Neuromorphic Heuristic Units can be coordinated through decentralised spike-triggered mechanisms. These efforts represent meaningful steps toward general neuromorphic optimisation. Nevertheless, existing prototypes retain hybrid execution: operator selection is neuromorphic, whereas candidate generation and fitness evaluation remain conventional. They reveal feasibility, but also the open challenges of scalability, competitiveness, and validation on physical hardware.

Across these strands, several structural limitations recur. Problem-specific neuromorphic solvers lack generality, whereas more flexible frameworks remain hybrid and therefore cannot realise the full benefits of event-driven execution. Existing models often emulate familiar EC approaches rather than inquiring about the search behaviours that naturally arise from spiking activity. Benchmarking is incomplete because very few studies report energy, latency, and solution quality together. Finally, most neuromorphic software ecosystems remain oriented toward computational neuroscience and machine learning, offering little direct support for expressing operators, populations, or search states in optimisation terms.

These limitations motivate the present work. The goal is a coherent NEC paradigm with clear evaluation protocols. This paper consolidates the current landscape and its limitations, including problem-specific solvers, hybrid frameworks, and neuroevolution pipelines. A CBTL-inspired reference architecture is then used to frame what must move on-chip, including operator embodiment, state estimation, and selection. Finally, a roadmap towards fully neuromorphic implementations is outlined, and the key challenges are summarised, including precision, scalability, learning rules, benchmarking, and deployment. The aim is not to outperform CPU-based EAs. The aim is to define what fully neuromorphic optimisation requires and to support reproducible progress.

III. Towards Fully Neuromorphic Optimisation

NEC assumes that coordination and operator selection can be expressed as spike-driven dynamics. Fig. 1 provides a CBTL-inspired architecture for this role as a conceptual reference.

A fully NEC optimiser would still require candidate generation, memory management, and objective evaluation (or suitable surrogates) to be realised within neural ensembles. The main obstacles are grouped into four categories: operator embodiment, memory architecture, representation and learning, hardware deployment, and adaptive and embedded systems.

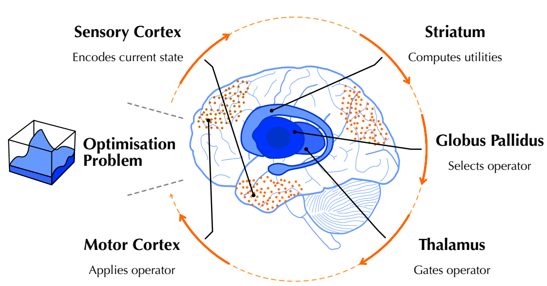

Fig. 1. Reference design for Neuromorphic Evolutionary Computation (NEC). A Cortico-Basal-Thalamic Loop (CBTL)-inspired spike-driven loop selects and reinforces search operators based on an observed optimisation state (conceptual schematic).

A. Operator Embodiment

In many existing NEC prototypes, operators are implemented as procedural routines rather than as neural processes. A fully neuromorphic optimiser requires each operator to be embodied as a spiking circuit that maps encoded memory states to new candidate solutions. The central challenge is to design circuits that approximate exploration and exploitation while preserving numerical reliability. Heavy-tailed perturbations for Lévy flights, pairwise differences for differential mutation, velocity-based updates for particle swarm, and rotational contractions for spiral search all admit spiking approximations. Nevertheless, each requires tailored mechanisms, such as stochastic spike generation [26], synaptic superposition [27], recurrent reservoirs [28], or approximate trigonometric projections [29]. The broader objective is to identify operator formulations that naturally arise from spiking dynamics and can be implemented with tractable neuron and synapse budgets.

B. Memory Architecture

In many current systems, solution memory is maintained in conventional RAM. In contrast, a fully neuromorphic optimiser requires an associative structure that stores, retrieves, and updates solution vectors through spike-based operations. Recurrent attractor networks [30] and vector symbolic architectures [31] provide viable substrates, since both support distributed representations and incremental updates through synaptic plasticity [32]. In these settings, stored solutions correspond to stable activity patterns, and retrieval arises from the network settling toward its nearest attractor. A central difficulty is fitness-based replacement, which requires competition circuits capable of comparing many stored items and selecting the least informative one. This is straightforward in conventional code but costly in spiking systems. Scalable selection, therefore, demands structured Winner-Takes-All (WTA) mechanisms [33] to avoid prohibitive neuron counts.

C. Representation and Learning

In many current systems, search state variables are computed outside the spiking substrate. A neuromorphic implementation requires these statistics to be produced within neural ensembles. Diversity can be approximated by circuits that estimate second-order statistics, supported by evidence that spiking populations can encode variance through homeostatic plasticity and divisive normalisation [34]. Improvement rate and convergence follow from temporal integration and fitness comparison, which can be realised through LIF dynamics and inhibitory competition. Beyond handcrafted features, learnt state representations offer a promising direction. Spike-based dimensionality reduction and reward-modulated plasticity can train encoder networks to extract task-relevant features directly from stored solutions and fitness values [32].

D. Hardware Deployment

The main benefit of neuromorphic optimisation is reduced energy consumption. Loihi 2, TrueNorth, SpiNNaker, and Akida operate at power levels several orders of magnitude below those of CPU-based simulations for comparable spiking workloads [4]–[6,35]. Deploying reference designs on Loihi 2 would enable direct energy measurements and clarify hardware limits on operator complexity, memory size, and decision latency. Event-driven computation offers further gains by updating only active neurons, yielding practical speedups when neural activity is sparse. Extending the evaluation to SpiNNaker and BrainScaleS, which offer distinct communication and timing properties [6], would help determine the portability of spike-driven optimisation across NC platforms.

E. Adaptive and Embedded Systems

A fully NEC optimiser offers capabilities that extend beyond energy efficiency. The Spike-Timing-Dependent Plasticity (STDP) enables the system to track time-varying objectives through learning rules without external control. This is particularly relevant in embedded settings where resources and communication are limited. An NEC optimiser on a drone or sensor platform, e.g., could adjust its behaviour in real time while operating within a sub-watt power budget. Edge-oriented neuromorphic processors such as BrainChip Akida and SynSense Xylo further extend this setting to ultra-low-power, on-device inference at milliwatt or sub-milliwatt levels, representing concrete deployment targets for embedded NEC implementations [8,9]. The event-driven nature of SNNs also supports parallel and asynchronous search, since multiple operator ensembles can explore different regions of the search space and exchange information through sparse spike-based signals, consistent with computational models of basal ganglia action selection [36]. These properties enable integration of NEC with other spike-based learning approaches, including reinforcement learning and sensorimotor control, and point toward a unified NC agent that adapts continuously within a single computational substrate.

IV. Challenges and Open Questions

Whilst the vision outlined above is technically plausible, substantial challenges remain. We group them into theoretical, algorithmic, hardware, and deployment issues.

A. Numerical Precision and Stability

Spiking representations offer limited effective precision, typically in the range of eight to twelve bits [29], which affects the reliability of fitness comparisons, diversity estimates, and operator weight updates. Therefore, a central question is the numerical accuracy required for consistent convergence. Possible directions include adaptive precision schemes that increase representational resolution near convergence, or the selective use of conventional processors for high-precision operations. The appropriate balance between biological plausibility, energy efficiency, and numerical stability remains unresolved. In particular, whether adaptive precision can be applied without distorting the spike-timing correlations on which STDP depends is itself an open sub-problem.

B. Scalability

High-dimensional problems pose a significant challenge, as performance degrades rapidly with increasing dimensionality D, and the number of neurons required for population-coded representations scales linearly with D. Thus, storing M solutions with N_{ens} neurons per dimension already demands $\mathcal{O}(M \cdot D \cdot N_{\text{ens}})$ neurons, which becomes substantial for realistic values of D and limits the resources available for operators and control circuits. Scaling to very high dimensions would require multi-chip deployments, accompanied by associated latency and energy costs. Dimensionality reduction offers a possible mitigation through learned low-dimensional representations, structural decompositions, or random projections. Still, each entails trade-offs in accuracy, data requirements, or prior knowledge. Designing reduction schemes that are both effective and compatible with neuromorphic substrates remains an open question. Compatibility with STDP imposes an additional constraint: any encoding must remain stable enough for eligibility traces to accumulate meaningful correlations across evaluation windows.

C. Operator Design

Existing neuromorphic optimisation studies cover only a small portion of EC mechanisms, and many modern operators, such as those in CMA-ES or SHADE, rely on adaptation rules and covariance estimates that are difficult to realise in spiking circuits [37,38]. This raises the question of which classes of operators are compatible with spike-based computation. Operators that depend on global synchronisation or precise matrix operations may be impractical, whereas those based on local interactions and stochastic sampling are more naturally expressed in distributed spiking dynamics. The issue of composability also remains open, since combining operators in SNNs requires that the output activity of one operator can serve as a stable input to another. A principled framework for composing neuromorphic operators, comparable to algebraic descriptions of classical EC techniques [39], would enable more flexible algorithm design.

D. Learning Mechanisms

In many existing approaches, the reinforcement rule is not expressed within the spiking substrate. A fully neuromorphic approach requires plasticity mechanisms such as STDP, reward-modulated STDP, or related three-factor rules [40]. These mechanisms are sensitive to short spike timing correlations. In contrast, operator selection evolves over much longer evaluation windows, and bridging this temporal gap requires eligibility traces or other forms of temporal abstraction, which incur additional circuit cost [41]. Utility mappings are often defined heuristically. An important question is whether such mappings can be learned across tasks. Meta-learning approaches that adapt learning rules from prior experience offer a promising direction, though their translation to SNNs remains largely unexplored.

E. Benchmarking

Evaluating NEC optimisers raises methodological challenges that extend beyond standard measures of solution quality and convergence [42]. Metrics such as energy use, wall clock time, neuron count, synaptic operations, and inter-chip communication are essential for characterising neuromorphic performance but are difficult to obtain without hardware access or detailed simulation. Existing simulators do not provide reliable energy estimates or remain accessible only to a limited number of groups. Fair comparison with conventional optimisers is also non-trivial, since NC-based methods may trade speed for energy efficiency. Thus, a consistent evaluation requires multi-objective benchmarking that reports performance, energy, and latency jointly and compares algorithms via Pareto-based analyses, rather than relying on single metrics. Recent benchmarking efforts such as NeuroBench propose standardised, multi-metric evaluation of NC algorithms and deployed systems, including optimisation scenarios such as QUBO [43]. Still, existing suites remain oriented toward perception and continual learning, and do not yet cover the quality, energy, and latency trade-offs central to NEC [12].

More concretely, evaluation should reflect NEC's objective and should not rely solely on solution quality. A minimal reporting checklist includes: (i) Quality, i.e, best-so-far value versus evaluations with confidence intervals; (ii) Energy (or Power), i.e., Joules per run and per improvement step when measurable; (iii) Latency, i.e., time to decision and time per evaluation cycle; (iv) Resources, i.e., neuron or synapse counts, spike rates, and communication volume; and (v) Reproducibility, i.e., released code, seeds, and profiling scripts. These requirements are compatible with emerging suites such as NeuroBench, which already report correctness, synaptic operations, sparsity, and system-level energy and latency [43].

F. Theory

Theoretical understanding of spike-driven optimisation remains limited. Convergence guarantees available for classical EC, such as Markov chain analyses of Simulated Annealing [44], have no clear counterparts for event-driven optimisers. In particular, the dynamics cannot always be framed as a regular or conventional EA with well-defined generations, populations, and update operators. A more appropriate object of analysis may be the stochastic process induced by the best-so-far value, i.e., the monotone sequence of incumbent improvements produced by asynchronous spiking interactions. Establishing conditions under which this best-so-far process converges (in probability or almost surely), relating its rate to resources such as neuron count, spike rates, and communication delays, and characterising the approximation capacity of spiking circuits with respect to EC primitives are all open problems. It is also unclear how the "No Free Lunch" results extend to NEC settings, and how to express the computational complexity of spike-driven operators in terms of synaptic operations rather than floating-point computations [45]. An extensive study is required to develop proof techniques and assumptions that faithfully reflect neuromorphic execution.

G. Tooling and Accessibility

Existing neuromorphic frameworks, such as Nengo, Brian, BindsNET, NEST, and the SpiNNaker toolchain, are primarily designed for computational neuroscience and machine learning, offering limited support for optimisation-specific operations, including fitness-based replacement, tournament selection, and operator composition. A dedicated neuromorphic optimisation library would lower the entry barrier for EC practitioners, e.g., comparable to Paradiseo [46]. Such a library should provide spiking implementations of standard operators, composable components for population and state management, hardware agnostic backends for Loihi, SpiNNaker, and CPU simulation, and utilities for energy and performance profiling. Educational resources and worked examples are also needed to bridge the gap between EC and neuromorphic engineering and to support wider adoption.

V. Roadmap

Advancing towards fully NEC requires shared benchmarks, open-source implementations, and cross-disciplinary venues bridging evolutionary computation and neuromorphic engineering. Hence, we organise the roadmap into three indicative time horizons, subject to hardware availability, community engagement, and the challenges discussed in this work.

- Short Term (1-2 years): Expand the spiking operator set; establish benchmarking protocols that report quality, energy, and latency jointly; prototype associative memories for solution storage; and deploy reference loops (e.g., CBTL) on NC hardware for direct measurements.
- Mid Term (2-4 years): Develop fully spiking perturbation (e.g., differential, swarm, genetic) operators; learn state representations with spike-based encoders; scale across multiple chips with efficient communication; and quantify energy-performance trade-offs across problem classes.
- Long Term (> 4 years): Design NEC combining temporal coding, inhibition, and traditional learning; support continual optimisation under non-stationary objectives via on-chip plasticity; ground spike-driven search theory in information-theoretic bounds; and validate end-to-end embedded deployments.

VI. Call to Action

Advancing NEC requires shared practices across communities to lower barriers to entry and facilitate meaningful comparisons of results. We propose three concrete steps toward reproducible and comparable NEC research.

1) Release reference implementations of coordination and operator selection circuits.
2) Report solution quality, energy consumption, and latency together when benchmarking.
3) Provide access to hardware or validated energy models to ensure fair comparison.

Acknowledgments

Grammarly and Claude Sonnet 4.6 were used solely for language polishing under human supervision. The authors remain fully responsible for the scientific content of this manuscript.

REFERENCES

[1] A. Eiben and J. Smith, "Introduction to evolutionary computing," *Assembly Automation*, vol. 24, no. 3, pp. 324–324, 2004.

[2] Z.-H. Zhan, L. Shi, K. C. Tan, and J. Zhang, "A survey on evolutionary computation for complex continuous optimization," *Artificial Intelligence Review*, vol. 55, no. 1, pp. 59–110, 2022.

[3] J. Liu, R. Sarker, S. Elsayed *et al.*, "Large-scale evolutionary optimization: A review and comparative study," *Swarm and Evolutionary Computation*, vol. 85, p. 101466, 2024.

[4] M. Davies, A. Wild, G. Orchard *et al.*, "Advancing Neuromorphic Computing With Loihi: A Survey of Results and Outlook," *Proceedings of the IEEE*, vol. 109, no. 5, pp. 911–934, 2021.

[5] F. Akopyan, J. Sawada, A. Cassidy *et al.*, "Truenorth: Design and tool flow of a 65 mW 1 million neuron programmable neurosynaptic chip," *IEEE transactions on computer-aided design of integrated circuits and systems*, vol. 34, no. 10, pp. 1537–1557, 2015.

[6] S. B. Furber, F. Galluppi, S. Temple, and L. A. Plana, "The spinnaker project," *Proceedings of the IEEE*, vol. 102, no. 5, pp. 652–665, 2014.

[7] C. Pehle, S. Billaudelle, B. Cramer *et al.*, "The brainscales-2 accelerated neuromorphic system with hybrid plasticity," *Frontiers in Neuroscience*, vol. 16, p. 795876, 2022.

[8] H. Bos and D. Muir, "Sub-mW Neuromorphic SNN audio processing applications with Rockpool and Xylo," 2022. [Online]. Available: https://arxiv.org/abs/2208.12991

[9] E. Bråtman and L. Dow, "Neuromorphic Medical Image Analysis at the Edge: On-Edge training with the Akida Brainchip," 2023.

[10] P. K. Enuganti, B. Sen Bhattacharya, T. Serrano Gotarredona, and O. Rhodes, "Neuromorphic computing and applications: a topical review," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 15, no. 2, p. e70014, 2025.

[11] D. Kudithipudi, C. Schuman, C. M. Vineyard *et al.*, "Neuromorphic computing at scale," *Nature*, vol. 637, no. 8047, p. 801–812, 2025.

[12] S. S. Chowdhury, D. Sharma, A. Kosta, and K. Roy, "Neuromorphic computing for robotic vision: algorithms to hardware advances," *Communications Engineering*, vol. 4, no. 1, p. 152, 2025.

[13] E. Talbi, "Neuromorphic-based metaheuristics: A new generation of low power, low latency and small footprint optimization algorithms," 2025. [Online]. Available: https://arxiv.org/abs/2505.16362

[14] Z. Lin and Z. Fan, "A ferroelectric memristor-based transient chaotic neural network for solving combinatorial optimization problems," *Symmetry*, vol. 15, no. 1, p. 59, 2022.

[15] H. Kim, M. Kim, A. Lee *et al.*, "Organic memristor-based flexible neural networks with bio-realistic synaptic plasticity for complex combinatorial optimization," *Advanced Science*, vol. 10, no. 19, p. 2300659, 2023.

[16] A. S. Lele, M. Chang, S. D. Spetalnick *et al.*, "Neuromorphic swarm on RRAM compute-in-memory processor for solving QUBO problem," *2023 60th ACM/IEEE Design Automation Conference (DAC)*, p. 6, 2023.

[17] A. Pierro, P. Stratmann, G. A. F. Guerra *et al.*, "Solving QUBO on the Loihi 2 Neuromorphic Processor," *arXiv*, 2024.

[18] C. D. Schuman, J. P. Mitchell, R. M. Patton *et al.*, "Evolutionary optimization for neuromorphic systems," in *Proceedings of the 2020 Annual Neuro-Inspired Computational Elements (NICE) Workshop*, 2020.

[19] C. D. Schuman, S. R. Young, B. P. Maldonado *et al.*, "Real-time evolution and deployment of neuromorphic computing at the edge," in *2021 12th International Green and Sustainable Computing Conference (IGSC)*. IEEE, 2021, pp. 1–8.

[20] S. Shen, R. Zhang, C. Wang *et al.*, "Evolutionary spiking neural networks: A survey," *Journal of Membrane Computing*, vol. 6, no. 4, pp. 335–346, 2024.

[21] Y. Fang, Z. Wang, J. Gomez, S. Datta, A. I. Khan, and A. Raychowdhury, "A swarm optimization solver based on ferroelectric spiking neural networks," *Frontiers in Neuroscience*, vol. 13, p. 855, 2019.

[22] T. Sasaki and H. Nakano, "Swarm intelligence algorithm based on spiking neural-oscillator networks, coupling interactions and search performances," *Nonlinear Theory and Its Applications, IEICE*, vol. 14, no. 2, p. 267–291, 2023.

[23] S. Snyder, S. R. Risbud, and M. Parsa, "Neuromorphic bayesian optimization in Lava," in *Proc. International Conference on Neuromorphic Systems (ICONS)*, 2023.

[24] S. Snyder, S. Risbud, and M. Parsa, "Asynchronous neuromorphic optimization in Lava," in *Proc. Great Lakes Symposium on VLSI*, 2024, p. 776–778.

[25] J. M. Cruz-Duarte and E.-G. Talbi, "Neuroptimisation: The spiking way to evolve," *arXiv*, 2025. [Online]. Available: https://arxiv.org/abs/2507.08320

[26] E. O. Neftci, H. Mostafa, and F. Zenke, "Surrogate gradient learning in spiking neural networks: Bringing the power of gradient-based optimization to spiking neural networks," *IEEE Signal Processing Magazine*, vol. 36, no. 6, pp. 51–63, 2019.

[27] C. Eliasmith and C. Anderson, *Neural engineering: Computation, representation, and dynamics in neurobiological systems*. MIT press, 2003.

[28] P. Wijesinghe, G. Srinivasan, P. Panda, and K. Roy, "Analysis of liquid ensembles for enhancing the performance and accuracy of liquid state machines," *Frontiers in neuroscience*, vol. 13, p. 504, 2019.

[29] M. Boerlin, C. K. Machens, and S. Denève, "Predictive coding of dynamical variables in balanced spiking networks," *PLoS computational biology*, vol. 9, no. 11, p. e1003258, 2013.

[30] D. Krotov and J. J. Hopfield, "Dense associative memory for pattern recognition," *Advances in neural information processing systems*, vol. 29, 2016.

[31] A. R. Voelker and C. Eliasmith, "Improving spiking dynamical networks: Accurate delays, higher-order synapses, and time cells," *Neural computation*, vol. 30, no. 3, pp. 569–609, 2018.

[32] B. Nessler, M. Pfeiffer, L. Buesing, and W. Maass, "Bayesian computation emerges in generic cortical microcircuits through spike-timing-dependent plasticity," *PLOS Computational Biology*, vol. 9, no. 4, pp. 1–30, 04 2013. [Online]. Available: https://doi.org/10.1371/journal.pcbi.1003037

[33] U. Rutishauser, R. J. Douglas, and J.-J. Slotine, "Collective stability of networks of winner-take-all circuits," *Neural computation*, vol. 23, no. 3, pp. 735–773, 2011.

[34] T. Toyoizumi, M. Kaneko, M. P. Stryker, and K. D. Miller, "Modeling the dynamic interaction of hebbian and homeostatic plasticity," *Neuron*, vol. 84, no. 2, pp. 497–510, 2014.

[35] B. Azadi, B. Anzengruber-Tanase, G. Sopidis, M. Haslgrübler, and A. Ferscha, "Towards an energy-efficient and sustainable iiot using embedded neuromorphic ai," in *Proc. of the 15th International Conference on the Internet of Things (IOT)*, 2025, p. 237–244.

[36] K. Gurney, T. J. Prescott, and P. Redgrave, "A computational model of action selection in the basal ganglia. i. a new functional anatomy," *Biological cybernetics*, vol. 84, no. 6, pp. 401–410, 2001.

[37] N. Hansen, "The cma evolution strategy: A tutorial," *arXiv preprint arXiv:1604.00772*, 2016.

[38] R. Tanabe and A. S. Fukunaga, "Improving the search performance of shade using linear population size reduction," in *2014 IEEE congress on evolutionary computation (CEC)*. IEEE, 2014, pp. 1658–1665.

[39] J. M. Cruz-Duarte, J. C. Ortiz-Bayliss, I. Amaya *et al.*, "Towards a generalised metaheuristic model for continuous optimisation problems," *Mathematics*, vol. 8, no. 11, p. 2046, 2020.

[40] N. Frémaux and W. Gerstner, "Neuromodulated spike-timing-dependent plasticity, and theory of three-factor learning rules," *Frontiers in neural circuits*, vol. 9, p. 85, 2016.

[41] W. Gerstner, M. Lehmann, V. Liakoni, D. Corneil, and J. Brea, "Eligibility traces and plasticity on behavioral time scales: experimental support of neohebbian three-factor learning rules," *Frontiers in neural circuits*, vol. 12, p. 53, 2018.

[42] N. Hansen, A. Auger, R. Ros *et al.*, "COCO: A platform for comparing continuous optimizers in a black-box setting," *Optimization Methods and Software*, vol. 36, pp. 114–144, 2021.

[43] J. Yik *et al.*, "Neurobench: A framework for benchmarking neuromorphic computing algorithms and systems," 2025. [Online]. Available: https://arxiv.org/abs/2304.04640

[44] D. Delahaye, S. Chaimatanan, and M. Mongeau, "Simulated annealing: From basics to applications," in *Handbook of metaheuristics*. Springer, 2018, pp. 1–35.

[45] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 67–82, 2002.

[46] J. Dreo, A. Liefooghe, S. Verel, M. Schoenauer, J. J. Merelo, A. Quemy, B. Bouvier, and J. Gmys, "Paradiseo: from a modular framework for evolutionary computation to the automated design of metaheuristics: 22 years of paradiseo," in *Proc. of the Genetic and Evolutionary Computation Conference Companion*, 2021, p. 1522–1530.