

Review

Not peer-reviewed version

---

# Machine Learning for Enhancing Metaheuristics in Global Optimization: A Comprehensive Review

---

[Antonio Bolufé-Röhler](#) \* and [Dania Tamayo-Vera](#)

Posted Date: 31 July 2025

doi: 10.20944/preprints202507.2611.v1

Keywords: Metaheuristic Optimization; Machine Learning; Global Optimization; Learnheuristics; Surrogate Modeling; Reinforcement Learning; Algorithm Configuration; Representation Learning; Adaptive Parameter Control; Evolutionary Computation



Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a Creative Commons CC BY 4.0 license, which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

*Review*

# Machine Learning for Enhancing Metaheuristics in Global Optimization: A Comprehensive Review

Antonio Bolufé-Röhler <sup>\*,†</sup>  and Dania Tamayo-Vera <sup>†</sup> 

University of Prince Edward Island

\* Correspondence: aboluferohler@upe.ca; Tel.: +1-416-859-4814

† 550 University Ave, Charlottetown, PE C1A 4P3, Canada.

## Abstract

The integration of machine learning (ML) with metaheuristic optimization has emerged as one of the most promising frontiers in artificial intelligence and global search. Metaheuristics offer flexibility and effectiveness in solving complex optimization problems where gradients are unavailable or unreliable, but often struggle with premature convergence, parameter sensitivity, and poor scalability. ML techniques—especially supervised, unsupervised, reinforcement, and meta-learning—provide powerful tools to address these limitations through adaptive, data-driven, and intelligent search strategies. This review presents a comprehensive survey of ML-enhanced metaheuristics for global optimization. We introduce a functional taxonomy that categorizes integration strategies based on their role in the optimization process, from operator control and surrogate modeling to landscape learning and learned optimizers. We critically analyze representative techniques, identify emerging trends, and highlight key challenges and future directions. The paper aims to serve as a structured and accessible resource for advancing the design of intelligent, learning-enabled optimization algorithms.

**Keywords:** metaheuristic optimization; machine learning; global optimization; learnheuristics; surrogate modeling; reinforcement learning; algorithm configuration; representation learning; adaptive parameter control; evolutionary computation

## 1. Introduction

Global optimization plays a central role in scientific discovery, engineering design, and decision-making under uncertainty [1]. It involves the task of finding the best possible solution in a non-finite, often non-convex search space where multiple local optima may exist and gradient information is typically unavailable or unreliable [2]. Due to their flexibility and generality, metaheuristic algorithms, such as Genetic Algorithms (GA), Particle Swarm Optimization (PSO), Differential Evolution (DE), Evolution Strategies (ES), Simulated Annealing (SA) and Covariance Matrix Adaptation Evolution Strategy (CMA-ES), have become indispensable tools for tackling these challenges [3]. Metaheuristics are particularly well suited for black-box optimization and multi-modal landscapes, and have demonstrated success across a wide range of applications [4,5].

Despite their practical utility, metaheuristics face several limitations. Their performance often depends heavily on parameter settings [6], and they may struggle to maintain a good balance between exploration and exploitation [7,8]. Furthermore, they tend to be sample-inefficient, especially in highly dimensional or computationally expensive scenarios [9,10]. These challenges have prompted growing interest in data-driven and adaptive approaches that can augment or guide heuristic search [11,12].

In parallel, machine learning (ML) has emerged as a powerful paradigm for pattern recognition, function approximation, and decision-making [13,14]. Its ability to learn from data and generalize across instances offers new opportunities to enhance the design, control, and adaptability of metaheuristic algorithms [15]. Recent research has shown that ML techniques can be employed to learn effective operators [16], model objective functions [17], analyze fitness landscapes [18], and even

design new optimization strategies altogether [19]. This convergence of learning and search has led to the emergence of a new class of methods, often referred to as “learning-enhanced” or “intelligent” metaheuristics [12,20].

This review aims to provide a comprehensive and structured overview of the integration of machine learning with metaheuristic optimization, with a particular focus on global optimization problems. We begin by presenting foundational concepts from both metaheuristics and machine learning, emphasizing the aspects most relevant to their integration. We then propose a taxonomy that categorizes the various ways in which ML can support or enhance metaheuristic search. Following that, we examine different ML techniques—including supervised learning, unsupervised learning, reinforcement learning, and meta-learning—and discuss how each contributes to the advancement of global optimization strategies. Throughout the paper, we highlight representative examples, identify emerging trends, and analyze critical challenges in the field. Finally, we outline key open questions and suggest promising directions for future research.

By synthesizing developments across these rapidly evolving domains, this review seeks to serve both as a resource for researchers entering the field and as a roadmap for advancing the design of intelligent optimization algorithms.

## 2. Background

### 2.1. Metaheuristics for Global Optimization

Global optimization refers to the task of locating the best possible solution in complex search spaces where objective functions may be non-convex, non-differentiable, noisy, or expensive to evaluate [1,2]. In such contexts, traditional exact methods are often impractical, necessitating the use of approximate algorithms that can efficiently explore the space and converge toward high-quality solutions. Metaheuristics are a broad class of optimization algorithms inspired by natural and physical processes, including evolution (e.g., Genetic Algorithms), social behavior (e.g., Particle Swarm Optimization), or thermodynamic analogies (e.g., Simulated Annealing) [3,5]. Their success lies in their general-purpose nature, simplicity of implementation, and ability to operate in black-box settings.

Central to all metaheuristics is the interplay between **exploration** and **exploitation**. Exploration refers to the global search behavior that prevents premature convergence by probing new regions of the solution space. Exploitation, by contrast, focuses on intensifying the search in promising areas already identified [7]. A well-designed algorithm must balance these forces to avoid stagnation in local optima while making steady progress toward convergence [21].

**Convergence** denotes the process by which a population or trajectory narrows around optimal (or suboptimal) regions over time. However, rapid convergence can be detrimental if it sacrifices solution quality or diversity too early in the search [6,22]. **Diversity maintenance** is thus a critical concept, ensuring that candidate solutions remain sufficiently varied to escape local basins and discover new optima. Another major consideration is **scalability**. While many metaheuristics perform well on low-to moderate-dimensional problems, their efficiency and effectiveness often degrade significantly as the dimensionality of the problem increases—a phenomenon sometimes referred to as the “curse of dimensionality” [9,10].

Despite their practical appeal, the performance of metaheuristics can be highly problem-dependent, requiring significant expertise to fine-tune algorithmic behavior for new domains. This sensitivity stems from their reliance on fixed heuristics and hand-crafted rules, which may fail to generalize across problem instances or adapt to changing landscape characteristics [6]. Moreover, their stochastic nature introduces variability in results, often necessitating multiple runs to obtain statistically meaningful conclusions [23].

Another challenge arises in the design of search operators—such as crossover, mutation, and perturbation schemes—which typically encode prior assumptions about the search space. When these assumptions are misaligned with the problem structure, the search may become inefficient or misguided [24]. Similarly, population-based methods can suffer from selection pressure dynamics:

excessive pressure may lead to loss of diversity, while insufficient pressure can delay convergence [25]. Striking the right balance is difficult without mechanisms that adapt to the evolving state of the search.

Metaheuristics are also inherently **agnostic to problem structure**. Unlike gradient-based methods that exploit analytic properties (e.g., Lipschitz continuity or smoothness), metaheuristics do not leverage domain-specific information unless it is manually incorporated. This generality is both a strength and a weakness—enabling black-box optimization, but potentially ignoring exploitable regularities in the problem landscape [26].

To address these issues, recent research has turned to machine learning as a means to imbue metaheuristics with learning capabilities. Instead of relying solely on manually designed rules and fixed control schemes, ML methods allow metaheuristics to learn from data—whether from prior optimization runs, ongoing search behavior, or collections of problem instances. This enables a shift from static to adaptive and even predictive search strategies. For example, regression models or neural networks can learn mappings from candidate solutions to objective values, serving as surrogates for expensive evaluations [27]. Clustering and dimensionality reduction techniques can uncover latent structure in the search space, facilitating more efficient navigation [28]. Reinforcement learning can be used to dynamically select operators or control hyperparameters in response to feedback from the search trajectory [29,30].

## 2.2. Relevant Machine Learning Foundations

The integration of machine learning with metaheuristic optimization relies on several foundational paradigms, each offering distinct capabilities for learning from data and improving search behavior. This section outlines the conceptual tools that underpin the enhancement of metaheuristics, with emphasis on the roles these tools play in guiding, modeling, and adapting the optimization process.

**Supervised learning** provides the framework for learning mappings from inputs to outputs based on labeled data. In the context of optimization, this paradigm is often used to predict objective values, model search landscapes, or classify candidate solutions as promising or unpromising based on historical evaluations [11,17]. Regression models such as Gaussian processes, neural networks, and ensemble methods are commonly employed in surrogate modeling [27,31], while classification approaches can be used to construct decision boundaries or feasibility predictors [20]. Supervised learning thus serves as a critical component in fitness approximation, model-based algorithm selection, and landscape-aware control.

**Unsupervised learning** focuses on uncovering latent structure in data without the need for labels. In metaheuristics, clustering algorithms are used to group similar solutions for adaptive population management, niching, or archive pruning [29,32], thereby promoting diversity and reducing redundancy. Dimensionality reduction techniques, such as principal component analysis (PCA), autoencoders, and other manifold learning methods, can aid scalability in some high-dimensional optimization problems by simplifying the search landscape. These tools facilitate latent space modeling and allow for exploration in reduced representations, which can significantly improve scalability in high-dimensional optimization problems [28,33].

**Reinforcement learning** (RL) formalizes decision-making under uncertainty by learning policies that map states to actions in a sequential manner, driven by reward signals [13]. Within metaheuristics, RL is increasingly applied to control operator selection, parameter tuning, and phase-switching strategies [29,30,34]. Unlike supervised methods, RL does not require pre-collected labels but learns from interaction with the environment, making it particularly suitable for adaptive online control during optimization. Techniques such as Q-learning, policy gradient methods, and deep reinforcement learning (DRL) are employed to manage exploration–exploitation trade-offs dynamically and autonomously [35].

**Meta-learning**, or “learning to learn”, extends the adaptability of ML by training models to generalize across distributions of tasks. In the context of global optimization, meta-learning enables optimizers to transfer knowledge between problem instances, automatically configure algorithm parameters, or even synthesize new search strategies [20,36,37]. Bi-level optimization and recur-



rent architectures such as Long-Short-Term Memory neural networks (LSTMs) or Transformers are often used to encode meta-learning processes [38]. This paradigm supports the development of self-improving, problem-agnostic optimizers that can adapt to new tasks with minimal data, improving sample efficiency and generalization.

Beyond these core paradigms, several cross-cutting methodologies play a foundational role in ML-enhanced metaheuristics. Among them, **surrogate modeling** and **representation learning** stand out for their versatility and impact. These techniques are not tied to a single learning paradigm but are applied across supervised, unsupervised, and reinforcement learning contexts. Both serve as enablers of data-efficient optimization and structure-aware search, and are integral to many of the most effective hybrid algorithms [17,27].

Integrating surrogate models effectively requires not only accurate predictions but also mechanisms for uncertainty quantification and model management throughout the optimization process [39,40]. Meanwhile, representation learning focuses on constructing compact, structured, or task-adaptive encodings of candidate solutions. In many real-world problems, raw decision variables may be redundant or ill-conditioned; learning more meaningful feature spaces can facilitate better sampling, more robust constraint handling, and improved generalization across instances [28,41].

Finally, optimization problems frequently arise in **non-stationary environments**, where the underlying objective function, constraints, or data distributions may shift over time. These changes may be gradual, abrupt, cyclical, or even adversarial in nature, and they are common in real-world scenarios such as adaptive control systems, dynamic supply chains, financial modeling, and user-centered recommender systems. In such settings, optimization algorithms must not only identify high-quality solutions but also continually adapt their strategies as the problem landscape evolves. Standard metaheuristics are typically designed under the assumption of stationarity, which limits their ability to respond effectively to concept drift or structural changes in the optimization task. To overcome this, **continual learning** and **online learning** methods have been introduced to endow the optimizer with a memory of past experiences and mechanisms for adaptation. Continual learning aims to incrementally acquire knowledge over time while avoiding catastrophic forgetting, often through mechanisms such as elastic weight consolidation, memory replay, or dynamic network expansion [42]. Online learning, in contrast, allows the model to update its parameters immediately based on new data, enabling real-time responsiveness and improved tracking of environmental changes.

Within ML-enhanced metaheuristics, these principles have been operationalized in several ways. For instance, adaptive parameter control mechanisms that use reinforcement learning can adjust mutation rates or population sizes in response to observed performance shifts during the search [34,43]. Similarly, clustering-based controllers or memory-enhanced surrogate models can detect transitions in the landscape and selectively update strategies or retrain approximators [44,45]. By integrating continual and online learning paradigms, metaheuristics become more than static optimizers; they evolve into adaptive agents capable of sustained performance in dynamic, uncertain, and information-rich domains. Together, these machine learning foundations form the theoretical and computational basis for enhancing metaheuristic optimization. They not only increase the adaptivity and intelligence of traditional methods but also open the door to new algorithmic paradigms where learning is tightly integrated with search.

### 3. Taxonomy of Integration Approaches

The integration of machine learning with metaheuristic optimization has led to a range of hybrid strategies aimed at improving search performance, adaptability, and scalability. These approaches differ based on the role that learning plays within the metaheuristic loop. In this section, we propose a taxonomy based on the functional role of ML in the optimization process.

#### 3.1. Learning or Adapting Search Operators

In traditional metaheuristics, the behavior of search operators, such as mutation, crossover, or neighborhood perturbation, is generally fixed or controlled through static heuristics [4]. Although

effective in many scenarios, such rigid designs fail to respond to the dynamic nature of the optimization process, where different phases of the search or characteristics of the objective function might benefit from different operator behaviors. To overcome these limitations, researchers have increasingly turned to machine learning techniques to enable metaheuristics to learn or adapt search operators dynamically during the optimization process.

One prominent direction involves the use of reinforcement learning to manage a portfolio of search operators. The metaheuristic is modeled as an agent that interacts with the fitness landscape, and operators are selected as actions based on their past success. For example, Ming et al. [30] proposed a deep reinforcement learning assisted operator selection framework for constrained multi-objective optimization. By modeling the population's convergence, diversity, and feasibility as states, and using a deep Q-network to estimate the value of operator actions, the framework dynamically selects operators that maximize population improvement. This approach was embedded into several constrained multi-objective evolutionary algorithms and achieved superior performance across benchmark problems compared to traditional fixed or random operator strategies.

Similarly, Durgut et al. [29] introduced an adaptive operator selection mechanism for an artificial bee colony (ABC) algorithm, leveraging Q-learning combined with clustering techniques. Their system maps problem states to the most effective operators in a pool, learning optimal sequences of operator applications over time. Applied to the Set Union Knapsack Problem, this RL-enhanced ABC algorithm significantly outperformed state-of-the-art methods, demonstrating the promise of adaptive learning mechanisms for discrete and binary optimization problems. While most work focuses on selecting between a set of predefined operators, other approaches adapt the internal behavior of individual operators themselves. A notable example is the work of Bolufé-Röhler and Luke [46], who enhanced an Estimation of Distribution Algorithm (EDA) using a machine-learned controller for its sampling behavior. Specifically, they extended the *Threshold Convergence* mechanism—originally designed to modulate step sizes during sampling—with a trained classifier that determines when to tighten or relax the sampling threshold based on features from the optimization state. This results in a learned, data-driven adaptation of the operator's exploration–exploitation trade-off, and showed consistent improvements over both fixed-threshold and conventional EDA implementations on benchmark suites from the Congress on Evolutionary Computation (CEC).

Although this paper focuses on global optimization, it is worth noting that similar ideas have been successfully explored in combinatorial contexts. For example, Karimi-Mamaghan et al. [32] proposed a Q-learning-based operator selection strategy for permutation flowshop scheduling problems. More recently, Johnn et al. [47] developed a Graph Reinforcement Learning for Operator Selection (GRLOS) framework, integrating deep reinforcement learning with graph neural networks to guide operator selection within Adaptive Large Neighborhood Search (ALNS) applied to vehicle routing problems. While these approaches target discrete problem spaces, they exemplify how ML techniques can enhance adaptive decision-making across diverse metaheuristic landscapes, providing inspiration for future global optimization research.

These adaptive strategies offer several advantages. First, they enhance performance by learning which operators (or operator configurations) are most effective in different stages of the search or types of problems, thus guiding the algorithm more intelligently. Second, they add flexibility and robustness, as adaptive operator selection can be applied across a wide range of metaheuristic frameworks and domains. Third, they promote automation by reducing or eliminating the need for manual tuning of operator probabilities or schedules [48].

However, implementing these approaches also introduces challenges. ML-based adaptation adds computational overhead, both in terms of model inference during optimization and in the training or data collection phase. Credit assignment can be difficult, as it is not always clear which operator contributed to success in stochastic search processes. Furthermore, ensuring that learned strategies generalize well across problem instances is an open research question, especially in black-box optimization scenarios [48].

### 3.2. Surrogate Modeling for Fitness Approximation

One of the most widely studied strategies for enhancing metaheuristic optimization, particularly for computationally expensive global optimization problems, is the use of surrogate models, also known as meta-models or approximate models. Surrogates are trained to approximate the true objective or constraint functions, allowing the optimizer to explore the search space efficiently while reducing the number of expensive exact evaluations. They are especially useful in domains such as engineering design, multi-objective optimization, and black-box global optimization, where each function evaluation may require significant simulation or experimental resources.

In the context of evolutionary computation, surrogate models are typically integrated either as substitutes for direct fitness evaluation or as guiding mechanisms to focus exploration. A foundational contribution in this area is the framework by Jin et al. [27], which introduced evolution control strategies to manage the interplay between surrogate predictions and real evaluations. Their work demonstrated that without appropriate control, approximate models can mislead the optimization process, particularly when approximation errors introduce false optima. The proposed solution combines individual-based and generation-based control mechanisms, ensuring correct convergence while significantly reducing computational effort.

Lim et al. [49] extended this idea by generalizing surrogate-assisted evolutionary computation, proposing memetic algorithms that run parallel local searches supported by ensemble surrogates. This approach leverages both the strengths and the imperfections of surrogates, turning the “curse of uncertainty” into a “blessing of uncertainty” by smoothing rugged landscapes to aid global exploration.

Recent advancements further illustrate the versatility and performance of surrogate-assisted techniques. Yu et al. [39] proposed a two-stage dominance-based surrogate-assisted evolutionary algorithm (TSDEA) tailored for high-dimensional, expensive multi-objective optimization problems. Their method combines radial basis function (RBF) models with an angle-penalized distance mechanism to maintain both convergence and diversity, outperforming several state-of-the-art competitors on large-scale benchmarks.

Another notable contribution is the adaptive switching framework by Chung et al. [50], which alternates between global and local search phases using a weighted maximin distance metric for exploration and multi-start gradient optimization for exploitation. This approach, tested on both synthetic benchmarks and real-world engineering problems, demonstrated superior robustness and efficiency compared to traditional surrogate-assisted schemes.

Machine learning models have also been integrated into surrogate-based global optimization frameworks. Bertsimas and Margaritis [51] developed a machine learning-enhanced mixed-integer optimization (MIO) framework that embeds surrogates such as decision trees, gradient boosted trees, neural networks, and support vector machines directly into MIO formulations. Their adaptive sampling and robust optimization enhancements improve solution feasibility and optimality, showing competitive performance against commercial solvers like BARON on benchmark sets.

A novel metaheuristic approach, Landscape-Sketch-Step (LSS), was introduced by Monteiro and Sau [52], combining reinforcement learning and stochastic search without explicitly constructing surrogate models. Instead, the method builds a dynamic state-value function to guide exploration, demonstrating promising performance on rugged low-dimensional landscapes.

Collectively, these studies highlight the growing sophistication of surrogate-assisted metaheuristics, encompassing diverse strategies such as ensemble modeling, adaptive control, local-global hybridization, and direct ML-model embedding. For a broader synthesis of the field, including applications to dynamic, constrained, and multi-modal optimization, we refer readers to the survey by Jin [17], which outlines key developments, challenges, and future directions in surrogate-assisted evolutionary computation.

### 3.3. Adaptive Parameter Control

One of the most influential lines of research in integrating machine learning with metaheuristics is the development of **Adaptive parameter control** strategies. These approaches aim to dynamically adjust critical parameters such as mutation rates, crossover probabilities, or cooling schedules—based on feedback from the search process itself, rather than relying on fixed or manually tuned values. This dynamic adjustment can improve search efficiency, robustness, and adaptability, especially in high-dimensional or multimodal global optimization problems.

Early pioneering work in this area includes Eiben et al. [43], who proposed using reinforcement learning to guide parameter adjustments online in evolutionary algorithms. Their approach modeled the parameter control process as a sequential decision-making problem, where an RL agent learns to adjust settings like mutation rates or population sizes to maximize search progress. This work laid foundational concepts for online learning control mechanisms.

Recent advances have significantly expanded these ideas. Tessari and Iacca [34] presented a general framework for reinforcement learning-based adaptive metaheuristics, demonstrating how various RL models, such as Q-learning, can be embedded within metaheuristics to adapt operators and parameters in real time. Similarly, Reijnen et al. [53] applied deep reinforcement learning to control key parameters of differential evolution in multi-objective optimization. Their DRL-APC-DE framework learned adaptive strategies for setting the scaling factor and crossover probability, outperforming static DE configurations on standard benchmark problems and showing promising generalization to unseen tasks.

In a broader review, Karafotias et al. [6] highlighted the benefits of adaptive control compared to static tuning: (1) the ability to tailor parameter values to different search phases (e.g., exploration early, exploitation later), (2) resilience to dynamic or non-stationary fitness landscapes, and (3) reduced manual effort, since adaptive mechanisms can self-regulate without extensive pre-tuning. However, they also note challenges such as balancing adaptation overhead and avoiding overfitting to specific problem instances.

Complementing these adaptive methods are automatic offline tuning approaches such as irace [54], which uses iterated racing techniques to identify optimal static configurations before runtime. While highly effective, such off-line tuning does not replace the need for online adaptability in dynamic environments.

More recently, Tatsis and Ioannidis [44] proposed an online cluster-based parameter control method, where machine learning techniques cluster similar search states and apply tailored parameter updates for each cluster, enhancing adaptability across heterogeneous search landscapes. This data-centric approach aligns with the trends observed by Bolufé-Röhler and Han [45], who advocate the use of ML to extract and exploit patterns in search dynamics to guide parameter decisions.

Comprehensive surveys such as Huang et al. [55] and Talbi [12] have systematically reviewed both offline (parameter tuning) and online (parameter control) strategies, emphasizing that while offline tuning is often simpler to implement, adaptive online approaches hold greater promise for generalization and robustness across problem domains.

Although considerable progress has been made, there remain open challenges. These include designing generalizable controllers that transfer across problem types, ensuring sample efficiency when learning during optimization, and integrating uncertainty modeling to account for noisy or partial feedback. Future research may explore hybrid systems combining offline-learned priors with online adaptive controllers, as well as leveraging advances in meta-reinforcement learning for cross-task generalization.

### 3.4. Offline Algorithm Configuration and Selection

While most **Learning-enhanced metaheuristics** focus on adapting search behavior *during* a single run, a complementary body of work applies machine learning techniques *before* the optimization process begins. Specifically, these methods aim to either (1) automatically configure a metaheuristic



by finding a static set of parameter values that perform well on average across multiple instances or (2) automatically select the best-performing algorithm or configuration for each individual problem instance. These approaches target cross-instance generalization, focusing on building reusable knowledge that can guide future runs.

One prominent direction in this area involves **offline configuration** techniques. Early approaches such as F-Race [56] treated parameter tuning as a generate–evaluate process, using statistical racing to quickly eliminate poor configurations. More advanced configurators combine racing with model-based search strategies. For example, ParamILS [57] performs iterated local search in the parameter space, while SMAC [58] and BOHB [59] use surrogate models, such as random forests or Bayesian optimization, to guide the exploration of promising configurations. The irace package [54] adopts adaptive sampling and sequential testing to focus evaluations on promising regions of the parameter space. These offline configurators have become standard tools for producing well-tuned static configurations for popular metaheuristics such as CMA-ES, DE, PSO, and various hybrid algorithms.

When no single configuration or algorithm consistently outperforms others across all problem instances, researchers turn to **per-instance selection and portfolio** strategies. A foundational example is SATzilla [60], which extracts cheap-to-compute instance features, trains regression models to predict the expected performance of each solver, and selects the one with the lowest predicted cost. Other systems such as ISAC [61] and Hydra [62] apply instance clustering or iterative greedy selection to build complementary solver portfolios. In continuous optimization, Exploratory Landscape Analysis (ELA) features [63] have been used to guide portfolio selection among CMA-ES, DE, and PSO variants. AutoFolio [64] combines SMAC-based configuration with SATzilla-style selectors, winning multiple international algorithm-selection competitions.

These offline approaches offer several advantages. First, they deliver strong *out-of-the-box* performance on unseen problem instances by leveraging patterns learned from prior experience. Second, they yield reproducible, fixed configurations or deterministic selection rules, simplifying deployment in industrial settings. Third, they can complement the online adaptive mechanisms described in previous sections, as offline-selected or offline-configured solvers can themselves include embedded online learning modules.

In summary, data-driven offline configuration provides strong plug-and-play solvers that can later be embedded within the online learning schemes discussed in Sections 3.1–3.3. Although training such models is compute intensive and hinges on good instance features, they furnish reproducible baselines and lighten deployment effort. Recent hybrid frameworks, such as BOHB’s combination of Bayesian surrogate modeling with multi-fidelity resource allocation [59], illustrate how offline and online principles can be blended. Reinforcing the view that static and adaptive strategies are complementary rather than competing.

### 3.5. Learning Landscape Characteristics

Understanding the structural characteristics of optimization landscapes—such as modality, ruggedness, neutrality, or constraint violation—is essential for designing effective global optimization strategies. Recent advances in machine learning and landscape analysis have opened powerful new pathways for characterizing these landscapes and dynamically adapting metaheuristic behavior.

A foundational pillar of this area is **exploratory landscape analysis**, first formalized by Mersmann et al. [65]. Subsequent extensions, such as the information-content features of Muñoz et al. [66] and the automated algorithm-selection framework of Kerschke and Trautmann [63], broadened ELA’s scope and strengthened its practical utility. ELA extracts a broad set of quantitative landscape features (e.g., dispersion, skewness, ruggedness, local-optima density) from continuous optimization problems, which can then feed ML models for tasks such as automated algorithm selection, performance prediction, or adaptive control. One key strength of ELA is its ability to leverage small samples to yield interpretable, human-readable insights.

Complementing this, Malan and Engelbrecht [67] introduced entropy-based measures to quantify ruggedness in continuous landscapes, offering a theoretically grounded approach to assessing

landscape difficulty. Malan's subsequent survey [68] provides a comprehensive synthesis of landscape analysis advances, including new landscape types (such as multiobjective, violation, or dynamic landscapes) and diverse applications spanning from algorithm explanation to automated configuration.

Building on these foundations, newer research has explored deep learning and latent representations for landscape characterization. Seiler et al. [69] evaluated deep learning-based, feature-free methods for characterizing continuous landscapes, highlighting the potential of convolutional and recurrent architectures to bypass handcrafted feature sets. These models promise scalability to complex, high-dimensional spaces but raise challenges around interpretability and computational overhead.

In terms of constraint handling, Malan [70] proposed a *landscape-aware switching* mechanism for differential evolution, in which on-line landscape features decide when and how to apply different constraint-handling techniques. This dynamic approach outperformed its individual constituent techniques on a diverse test set.

A particularly innovative development comes from Karp et al. [71], who proposed the Landscape-Aware Growing (LAG) strategy for model scaling in deep learning. Unlike traditional loss-preserving growth heuristics, LAG uses early training dynamics after model expansion to predict final performance, demonstrating that landscape signals shortly after initialization (rather than at initialization itself) provide the most reliable indicators for selecting efficient scaling strategies. Although developed in the context of deep neural networks, this insight carries intriguing implications for optimization more broadly, especially in settings requiring adaptive, stage-wise strategies.

Comparatively, classical ELA approaches excel in interpretability and integration with existing metaheuristics but can struggle in high-dimensional or noisy settings. Deep learning-based methods offer superior scalability and adaptability but at the cost of explainability and often requiring significant data. Latent space optimization, as explored by Tripp et al. [33], enables metaheuristics to search in compressed spaces, reducing dimensionality and potentially smoothing rugged landscapes, but still faces open questions about preserving global optima during projection. Finally, landscape-aware strategies like LAG challenge conventional assumptions, shifting the focus from static structural metrics to dynamic, emergent patterns observed during early optimization phases.

Taken together, these works underscore the growing role of ML in extracting, learning, and exploiting landscape knowledge to inform global optimization. They mark a shift away from static, expert-designed heuristics toward data-driven, adaptive strategies that can dynamically tailor search behaviors to the underlying problem structure.

### 3.6. Meta-Learning and Self-Improving Optimizers

The ambition to learn not just parameters of an algorithm but the algorithm itself—often referred to as *learning to optimize* or meta-learning—has deep roots in machine learning. One of the early breakthroughs was the LSTM-based optimizer by Andrychowicz et al. [36], which demonstrated how update rules could be parameterized and trained via backpropagation through the optimization process. These early works, focused on supervised learning settings, formalized core meta-learning principles such as bilevel optimization (with inner and outer loops), generalization across task distributions, and the replacement of hand-crafted update rules with learned modules [72].

Although this paradigm originated in gradient-based learning tasks, its influence has extended to global optimization, where gradients are unavailable and optimization relies on population-based metaheuristics like CMA-ES, DE, or ES. In this context, meta-learning techniques have been adapted to learn control policies, dynamic update rules, or even entire algorithmic parameterizations, often yielding significant gains in performance and adaptability.

A notable example is the Learned Evolution Strategy (LES) proposed by Lange et al. [38], which uses a small self-attention network trained offline to learn recombination weights and step-size control policies. By training across a distribution of test functions, the LES generalizes well to new problems and longer optimization horizons, matching or surpassing canonical ES on the MuJoCo tasks they tested.

In parallel, several works have embedded reinforcement learning agents within existing meta-heuristics to enable online control. For instance, Shala et al. [73] use guided policy search to control the step-size in CMA-ES, outperforming standard Cumulative Step-Size Adaptation. Similarly, recent work by Yang et al. [74] and Zhao et al. [75] applies Q-learning to adapt mutation and exploration strategies in multi-population DE, leading to robust improvements across CEC benchmark suites. Tessari et al. [34] go a step further by training a policy to switch among multiple metaheuristics (CMA-ES, PSO, DE) based on online feedback—illustrating how policy learning can function as a meta-level controller that adapts to changing landscape conditions.

Rather than learning how to control an existing algorithm, some approaches evolve the structure of the algorithm itself. For example, Chen et al. [76] introduce MetaDE, which treats the entire DE pipeline (mutation, crossover, and population structure) as a search space and evolves improved variants via GPU-accelerated differential evolution. Guo et al. [41] take a similar approach with LCC-CMAES, where a neural controller is trained to schedule cooperative-coevolution decompositions, enabling optimization in extremely high-dimensional spaces (up to 10,000 variables).

An earlier and simpler form of meta-learning is to predict when to switch from exploration to exploitation in hybrid algorithms. This transition decision has also been modeled as a supervised classification task in hybrid metaheuristics [77], using run-time features as inputs. Once trained, the classifier generalizes across different functions and dimensionalities, making it an effective low-cost control strategy.

Recent surveys have begun mapping this emerging design space. Ma et al. [37] frame meta-learning in black-box optimization (Meta-BBO) as a unifying paradigm connecting online control, algorithm selection, and hyper-heuristics. Szenasi et al. [78] catalog ML-enhanced local search hybrids, and Nomura et al. [79] show that even individual parameters (like the covariance learning rate in CMA-ES) can benefit from meta-learned tuning based on signal-to-noise ratios.

While progress is promising, several open challenges remain. These include improving sample efficiency (can meta-learned optimizers generalize from a small set of expensive problems?), expanding theoretical understanding (which classes of problems admit universal learned optimizers?), and enhancing interpretability (can learned strategies be distilled into new design principles?). Notably, work on reverse-engineering LES has shown that learned strategies can resemble concise, human-readable heuristics, suggesting that meta-learning may also serve as a tool for algorithm discovery.

In summary, meta-learning is gradually reshaping the landscape of global optimization. Whether through offline-learned update rules, RL-driven controllers, meta-evolved pipelines, or supervised control triggers, these approaches all aim to automate the design of self-improving metaheuristics, extending the legacy of handcrafted algorithms with data-driven intelligence.

3.7. Summary Table

Table 1 provides a comparative overview of the main categories in our taxonomy.

Table 1. Bird’s-eye summary of ML integration patterns reviewed in Section 3.

Category	What Is Learned?	When/Where Used?	Typical ML Techniques	Canonical Metaheuristics	Key Gains / Main Caveat
Learning / Adapting Operators	Operator choice or parameters	Every generation (online)	Bandit RL, Q-learning, policy gradient	GA, DE, PSO, ABC [29,30,46]	+ Phase-sensitive search / - Credit-assignment noise
Surrogate Modeling	Cheap fitness proxy	During evaluation (in-loop)	GPs, RBFs, DNN ensembles	ES, CMA-ES, BO hybrids [27,39,50]	+ Drastically fewer expensive calls / - Model-bias risk
Adaptive Parameter Control	Mutation rate, step-size, etc.	Periodic or event-triggered (online)	TD-RL, regression, clustering	SA, DE, CMA-ES, PSO [34,43,53]	+ Self-tuning across phases / - Overhead on tiny budgets
Offline Configuration & Selection	Static parameter set or solver portfolio	Before run (offline)	SMAC, BOHB, decision forests	CMA-ES, VNS, SAT/VRP portfolios [57,59,60,64]	+ Plug-and-play performance / - Needs large training set
Landscape Learning	Modality, ruggedness, constraint descriptors	Pre-run probes or intermittent sampling	ELA features, CNNs, autoencoders	Problem-agnostic (guides choice/control) [65,68,69]	+ Instance-aware solver matching / - High-D or noisy landscapes tricky
Meta-Learning Learned Optimizers	Update rule or control policy itself	Trained offline, applied online across tasks	LSTM optimizers, attention nets, meta-RL	Learned ES, meta-CMA-ES, DE variants [36–38]	+ Can discover new heuristics / - Costly training, generalization limits

## 4. Machine Learning Techniques Applied

### 4.1. Supervised Learning

Supervised learning techniques have been extensively applied within metaheuristics to build predictive models that guide search, particularly when dealing with expensive objective functions or the need for selective evaluation. Their most notable impact is in surrogate-assisted optimization, fitness landscape modeling, and solution preselection. One prominent class of applications is **surrogate-assisted evolutionary algorithms** (SAEAs), where supervised regression models replace or augment exact objective evaluations. For example, Jin et al. [27] demonstrated a framework where radial basis function networks are used to approximate the fitness landscape, reducing the number of exact evaluations by an order of magnitude. More recent implementations rely on Gaussian processes, which offer both predictions and uncertainty estimates, and are widely used in Bayesian optimization pipelines. A concrete example is provided by Guo et al. [31], who applied dropout neural networks to create uncertainty-aware surrogates in high-dimensional multi-objective problems. Their method used Monte Carlo dropout to quantify predictive uncertainty, allowing the algorithm to avoid over-exploitation of potentially misleading predictions. This led to substantial gains in sample efficiency and solution diversity in engineering design benchmarks.

Supervised models have also been integrated into **online model management strategies**. In the work by Yu et al. [39], two surrogate models (a global and a local one) were adaptively selected based on performance, helping the optimizer switch between exploration and exploitation modes. This dynamic selection enabled better responsiveness to landscape heterogeneity, which is often difficult to address with static heuristics alone.

In another line of work, **classification-based preselection** has been used to discard poor-quality or infeasible solutions before costly evaluation. Calvet et al. [20] embedded supervised classifiers into a hybrid routing optimization framework, where decision trees were trained to identify low-potential moves based on historical performance. This allowed the metaheuristic to focus computational effort on more promising regions without manually encoding decision rules. Supervised learning has also been leveraged to enhance **selection and replacement strategies** in population-based algorithms, particularly in scenarios involving many-objective or noisy optimization. Instead of relying solely on raw fitness evaluations or Pareto dominance—which can be unreliable in the presence of noise or computational cost—surrogate models are trained to approximate solution quality and guide the selection process. For instance, Han et al. [80] proposed a surrogate-assisted evolutionary algorithm for many-objective optimization in the refining process. Their approach employed ensemble learning to predict solution quality and prioritize candidates, thereby reducing the number of expensive evaluations while preserving convergence pressure. By integrating prediction-based preselection, the algorithm effectively navigated complex search spaces with reduced reliance on direct fitness comparisons.

These examples illustrate that supervised models are not merely plug-ins for fitness approximation—they can play active roles in guiding the core evolutionary operators, improving adaptivity, and mitigating stagnation. Their success depends not only on prediction accuracy but also on how uncertainty, model updating, and computational overhead are handled within the optimization loop. For instance, supervised classification has also been employed to regulate exploration pressure in distribution-based metaheuristics, where a learned controller adjusts sampling parameters based on observed optimization dynamics [46].

### 4.2. Unsupervised Learning

Unsupervised learning techniques are commonly integrated into metaheuristics to extract structural information from evaluated solutions, allowing the optimizer to adapt its behavior without explicit labels. These methods are particularly valuable for preserving diversity, identifying latent structure, and enhancing scalability in high-dimensional spaces. One of the most impactful uses of unsupervised learning in metaheuristics is the integration of **clustering techniques** to enhance popula-



tion diversity, control mating strategies, or define adaptive niches. These methods extract structural information from the population to steer selection, variation, and environmental pressure more intelligently. Wang and Zhang [81] proposed a K-means clustering-based offspring generation mechanism for evolutionary multi-objective optimization. By partitioning solutions in the objective space, their method adapted mating pool construction based on local search behaviors, achieving improvements in convergence and diversity. Similarly, Zhang et al. [82] introduced a fuzzy c-means clustering-based mating restriction strategy that limited crossover to within-cluster individuals, thereby preserving local convergence properties while maintaining global diversity.

Affinity propagation has also been applied for automated niching. Wang et al. [83] developed a differential evolution variant that uses contour prediction and affinity propagation clustering (APC) to discover and exploit multimodal niches. Their approach outperformed traditional niching methods by dynamically adapting the number and shape of clusters based on evolving solution distributions. Alternative formulations based on immune-inspired metaheuristics have also incorporated clustering as a diversity mechanism. Tsang and Lau [84] designed a multi-objective immune algorithm where clusters were used to modulate clonal selection pressure and manage competing subpopulations along the Pareto front. Across these frameworks, clustering provides a powerful unsupervised signal for structuring variation, preserving diversity, and enabling region-specific search—all of which are crucial in solving multimodal, many-objective, or noisy optimization problems.

**Dimensionality reduction** techniques are used to simplify visualization, analysis, or search in complex, high-dimensional landscapes. Methods such as PCA and t-distributed stochastic neighbor embedding (t-SNE) have been incorporated into visual steering tools that help human-in-the-loop optimizers interpret search dynamics. In more autonomous settings, low-dimensional embeddings can also be used to bias variation operators toward meaningful subspaces. For instance, Lim et al. [85] introduced a PCA-based mutation operator in genetic algorithms applied to IIR filter design. Their method perturbed candidate solutions along principal directions of variance, which increased genetic diversity and improved convergence compared to uniform or non-uniform mutations.

Another powerful application involves **latent space modeling via deep unsupervised learning**. Autoencoders and variational autoencoders (VAEs) are increasingly used to encode candidate solutions into compressed representations, enabling exploration and variation in a structured latent space. Tripp et al. [33] proposed an architecture where candidate solutions are encoded and then perturbed in the latent space before decoding back to the solution space. This allowed the search to respect underlying manifold structure and operate more effectively in high-dimensional domains with correlated or redundant variables.

These unsupervised techniques are particularly beneficial in settings where search landscapes are complex, noisy, or poorly understood. By extracting structure from the evolving population, they provide a foundation for adaptive sampling, diversity control, and dimensionality-aware search, thereby improving both robustness and efficiency.

#### 4.3. Reinforcement Learning

Reinforcement learning provides a powerful framework for adaptive control in metaheuristic optimization. By modeling the metaheuristic as an agent interacting with its search environment, RL allows the algorithm to dynamically select operators or adjust parameters based on feedback from the search process. Early demonstrations, such as Eiben et al.'s Q-learning approach to online parameter control, showed that evolutionary algorithms could benefit from real-time adaptation without relying on fixed schedules [86]. Recent studies have significantly advanced this idea. For instance, Durgut et al. applied a simple RL-based strategy to learn which variation operator to apply at each iteration, leading to consistent gains over static heuristics [29]. Similarly, Ming et al. developed a deep Q-network that selects among crossover and mutation strategies based on features like population diversity and convergence, demonstrating superior performance in constrained multi-objective settings [30].

Beyond operator selection, RL has been used for dynamic parameter tuning. Reijnen et al. used a deep RL agent to adjust differential evolution parameters online, improving performance on problems

with shifting landscapes [53]. Von Eschwege and Engelbrecht integrated a Soft Actor-Critic (SAC) agent into particle swarm optimization, enabling the algorithm to continuously adjust its learning coefficients based on swarm behavior [87].

RL can also coordinate between different metaheuristics. Tessari and Iacca trained a policy-gradient agent to switch among algorithms like DE, CMA-ES, and PSO, choosing the most appropriate one for each search phase [34]. Guo et al. applied a similar strategy to switch among DE variants [88]. These approaches transform the metaheuristic into a self-configuring strategy selector, guided by learning rather than static design. RL has also been employed to control restart timing and reinitialization strategies in hybrid metaheuristics. For example, a deep Q-network was used to implement a smart restart mechanism that dynamically switches between exploration- and exploitation-focused configurations, yielding significant performance gains on CEC benchmark problems without requiring prior knowledge of the problem structure [35].

Among the strengths of RL in metaheuristics are adaptive control, online learning, and flexibility across diverse problems. RL-based methods can continuously refine their strategies during the run, without requiring offline training. This makes them well-suited for dynamic or non-stationary optimization. Multi-objective settings can also benefit from custom reward designs that balance competing objectives [30]. However, challenges remain. RL is often sample-inefficient, requiring many evaluations to learn effective policies. State and reward design is non-trivial and problem-dependent. Poorly shaped rewards can mislead learning, and high-dimensional or partially observable states can overwhelm simple RL algorithms. Deep RL approaches mitigate these issues but introduce complexity in tuning and stability. Moreover, RL agents may overfit to specific problem types or learn brittle behaviors if not properly generalized. Nevertheless, reinforcement learning continues to evolve as a key enabler of intelligent, adaptive metaheuristics. Its ability to guide search based on learned experience opens new possibilities for robust and flexible global optimization.

#### 4.4. Representation Learning

**Representation learning** aims to discover compact, meaningful embeddings of candidate solutions or search spaces that improve the effectiveness of metaheuristic optimization. Instead of operating in raw, high-dimensional decision spaces, metaheuristics can benefit from latent spaces that capture important structure or constraints, making the search smoother and more efficient. Autoencoders and variational autoencoders (VAEs) are commonly used to construct such representations. Tripp et al. showed that searching in the latent space of a VAE trained on candidate solutions can drastically reduce the number of evaluations needed for convergence [33]. Latent variables encode high-level patterns in the data, allowing metaheuristics to perform variation along semantically meaningful directions. Representation learning has proven particularly useful in constrained optimization. Bentley et al. introduced the COIL framework, which trains a VAE on feasible solutions to learn a feasibility-biased latent space. Metaheuristic search conducted in this space naturally produces valid solutions, alleviating the burden of constraint handling [89].

In addition to handling constraints, representations can support transfer learning. By training embeddings on multiple related tasks, metaheuristics can generalize across problem instances. Wang et al. demonstrated that a co-surrogate model using a transfer-learned latent space can accelerate multi-objective optimization [90]. Principal component analysis (PCA) and other linear embeddings have also been used to bias variation operators. Lim et al. proposed a PCA-based mutation operator that perturbs individuals along principal directions, improving exploration [91]. More complex models such as GANs have been employed to generate realistic candidate solutions in design optimization. The advantages of representation learning include dimensionality reduction, constraint embedding, and knowledge reuse. Representations can smooth rugged search landscapes and steer the search toward feasible or promising regions. They also allow for amortized inference: once trained, the encoder and decoder can be quickly applied during optimization. Yet, representation learning introduces new challenges. Poor training data can bias the latent space away from optimal regions. Training deep models adds computational overhead, and determining the right dimensionality for the latent space

is often non-trivial. Interpretability is another concern, especially when using black-box neural embeddings. Additionally, these methods require sufficient training samples, which may be unavailable in sparse-data settings. Overall, representation learning offers a powerful toolkit for structuring the search space in metaheuristic optimization. When properly integrated, it enables algorithms to operate more intelligently and efficiently by leveraging patterns learned from the problem domain.

#### 4.5. Meta-Learning and Learnheuristics

Meta-learning enables optimizers to improve based on past experiences, shifting from problem-specific tuning to generalizable learning across tasks. This approach captures the idea of "learning to learn," allowing metaheuristics to adapt faster and more effectively to new optimization problems. A foundational concept is optimizer transfer: using performance data from previous runs to inform decisions on new problems. Early work by Calvet et al. introduced learnheuristics, where machine learning is used to predict heuristic behavior in response to changing inputs [20]. This laid the groundwork for adaptive optimizers that evolve based on accumulated experience. Modern approaches go further by training neural models to perform optimization. Andrychowicz et al. trained LSTM networks to act as optimizers via meta-learning, showing that learned update rules can outperform hand-crafted ones on new tasks [36]. Chen et al. surveyed learning-to-optimize frameworks, highlighting how recurrent and transformer models can generalize across optimization tasks [72].

**Few-shot learning** techniques enable rapid adaptation with minimal data. Meta-learners can identify the closest known problems to a new instance using initial evaluations, allowing fast algorithm selection or configuration. Guo et al. demonstrated deep RL-based dynamic algorithm selection, an approach that can be extended to meta-learning by training controllers on performance data [88]. Self-configuring optimizers represent another trend. Shala et al. trained a CMA-ES variant with a learned step-size adaptation rule, improving convergence on unseen problems [73]. Such continual learning approaches allow the optimizer to refine its internal behavior across runs.

LSTMs and transformers are particularly effective for modeling optimization trajectories. Lange et al. introduced the Evolution Transformer, which learned to imitate an evolution strategy and generalized to new tasks via imitation learning [38]. Supervised meta-controllers can also be trained using historical data. These models map problem features to algorithm configurations or operator choices, reducing the need for manual tuning [92]. Durgut et al. proposed reusing RL policies across problems, demonstrating that knowledge transfer can reduce learning overhead [93]. The strengths of meta-learning include rapid adaptation, robust generalization, and reduced need for manual configuration. Meta-learned optimizers can identify patterns across tasks, recall effective strategies, and configure themselves with minimal input. They are particularly valuable in dynamic or repetitive optimization scenarios. However, meta-learning also demands extensive training data, increasing upfront cost. Complex models may lack interpretability, and negative transfer can degrade performance on dissimilar tasks. Implementing meta-learning frameworks requires expertise and may involve significant development effort. Despite these challenges, meta-learning holds promise for creating autonomous optimizers that improve with experience. As training data becomes more accessible and architectures more efficient, meta-learning is poised to transform how metaheuristics adapt to complex, evolving problem landscapes.

## 5. Emerging Trends

The integration of machine learning into continuous metaheuristic optimization has progressed from basic surrogate assistance to a rich and growing ecosystem of data-driven, learning-enhanced strategies. Recent developments highlight six particularly promising and interconnected trends, signaling a fundamental shift in how optimization algorithms are designed, adapted, and deployed.

### 5.1. Learned Optimizers via Sequence Models

A new class of approaches seeks to learn the optimization algorithm itself, replacing hand-crafted update rules with trained neural architectures. This trend is exemplified by **in-context evolutionary**

**optimization**, where large Transformer models ingest sequences of fitness and search data to propose updates in an autoregressive fashion. Notably, the *Evolution Transformer* is trained via behavioral cloning to mimic the distributional update of canonical evolution strategies and can generalize across tasks and horizons [94]. In parallel, the *MetaBBO* framework learns attention-based update rules for evolution strategies through a bi-level meta-optimization loop [38].

More recently, *RIBBO* reframes black-box optimization as a sequence modeling task. By augmenting optimization traces with regret tokens and using hindsight relabeling, it enables Transformers to learn robust sampling strategies offline, outperforming their teacher optimizers on BBOB, HPO, and control tasks [95]. Together, these works suggest that large-scale, attention-based architectures can act as general-purpose optimizers, trained once and reused across problem families.

### 5.2. Reinforcement Learning for Online Control

Whereas sequence models aim to learn the full optimizer offline, reinforcement learning is increasingly used to adapt critical components *during* search. For instance, Soft Actor-Critic agents have been used to control PSO velocity parameters on the fly, enabling robust, hyperparameter-free performance [87]. In a similar vein, online adjustment of the CMA-ES learning rate based on a signal-to-noise heuristic allows for more efficient adaptation without population inflation [79].

RL can also orchestrate hybrid algorithms. In [35], a deep Q-network is trained to manage smart restarts between exploration- and exploitation-focused configurations in a CMA-ES–UES hybrid. At a more applied level, RL has been successfully embedded in Crow Search algorithms for real-time energy loss optimization in smart grid systems, improving voltage stability and reducing distribution loss [96].

These examples show how RL bridges the gap between reactive search and proactive decision-making, especially when paired with well-defined state representations and reward signals.

### 5.3. Surrogate Modeling: Deep, Explainable, and Cost-Aware

Surrogate models remain a core tool in metaheuristic optimization for reducing expensive evaluation costs. Recent advances have not only improved their scalability and accuracy but also extended their functionality into domains like explainability, budget-awareness, and transfer learning. Deep learning surrogates, including dropout neural networks, now serve dual roles as high-capacity predictors and uncertainty estimators. These models outperform traditional Gaussian processes in high-dimensional and many-objective problems, where standard kernels become infeasible [31]. Their robustness and scalability have enabled broader deployment in real-world applications, such as engineering design and bioinformatics, where evaluating each candidate solution is computationally intensive.

Explainability has emerged as a surprising but powerful complement to surrogate modeling. Instead of relying solely on surrogate predictions to rank or sample solutions, some frameworks now use interpretable features of the surrogate to inform variation. For instance, in the EXO-SAEA framework [97], SHAP values derived from the surrogate model guide crossover and mutation by identifying influential input variables. This approach not only improves convergence but also embeds problem-specific knowledge directly into the search dynamics. By integrating model explanations into the generation of new solutions, the optimization process becomes more strategic and data-aware.

Surrogate modeling is also adapting to the reality of asymmetric evaluation costs. In many multi-objective problems, not all objectives are equally expensive to compute. Recent approaches use co-surrogates to learn the relationships between low-cost and high-cost objectives, allowing the optimization to defer expensive evaluations until they are most informative [90]. This enables more efficient discovery of the Pareto set and opens the door to cost-aware strategies that dynamically allocate computational resources across objectives.

A recent survey consolidates these trends into a general design blueprint for surrogate-assisted evolutionary algorithms [40]. The review organizes the space into four core components—sampling, modeling, control, and integration—and highlights open challenges such as handling uncertainty



under dynamic conditions and managing multiple models over time. This framework provides both a conceptual roadmap and a practical checklist for future developments. Together, these innovations suggest that surrogate modeling is evolving from a passive evaluation shortcut into a fully integrated layer of intelligence within metaheuristics. Deep models expand the reach, explainability improves decision-making, and budget-awareness brings real-world practicality. As optimization problems grow more complex, surrogate-assisted strategies will likely play an even more central role in adaptive, scalable, and interpretable search.

#### 5.4. Generative Diffusion Models for Offline Optimization

Generative diffusion models have recently emerged as a promising new paradigm for offline black-box optimization. These models learn to sample from high-fitness regions of the solution space by iteratively denoising noise vectors toward the data distribution, offering a powerful generative framework that can operate without explicit surrogate models or gradients.

Li et al. [98] introduced a reward-directed diffusion model that employs classifier-free guidance to steer samples toward high-reward regions. By incorporating reward signals directly into the denoising process and deriving formal sub-optimality bounds, their approach bridges the gap between probabilistic modeling and optimization. This makes it particularly effective for problems with noisy or limited logged data, where traditional optimization algorithms may struggle due to sparse supervision or costly evaluations.

Complementing this, Krishnamoorthy et al. [99] proposed DDOM (Diffusion-based Decision Optimizer for Modeling), which uses a reweighted loss function during training to emphasize high-reward samples. At inference time, DDOM applies a guidance mechanism that nudges the generative process toward extrapolated regions of the fitness landscape—potentially sampling beyond known optima. Their results show that DDOM consistently outperforms GAN-based and surrogate-assisted approaches on benchmark problems from the Design-Bench suite, demonstrating the method's sample efficiency and flexibility.

One of the key strengths of diffusion models lies in their ability to capture complex, multimodal distributions over solutions. Unlike GANs, which can suffer from mode collapse, or surrogates, which are limited to pointwise approximation, diffusion models naturally learn a full generative process that preserves diversity and structure in the solution space. This makes them especially well-suited for domains like materials discovery, protein design, or neural architecture search, where optimal solutions are sparse and varied.

By shifting the optimization paradigm from explicit modeling to implicit generative reasoning, diffusion models provide a scalable, surrogate-free framework for offline optimization. As these models become more expressive and controllable, they offer new avenues for integrating machine learning with global search—particularly in settings where data is fixed, structured, or expensive to obtain.

#### 5.5. Landscape-Aware and Data-Centric Search

While traditional metaheuristics often rely on heuristic restarts, niching, or random diversification, recent methods are shifting toward explicit modeling of the fitness landscape to guide search more intelligently. Instead of treating the search space as opaque, these algorithms infer structural properties such as modality, basin boundaries, and optima distribution to inform decisions during optimization.

The LADE algorithm exemplifies this trend by actively maintaining a memory of detected peaks and employing peak-distinction heuristics to distinguish between global and local optima [100]. When the search stagnates or risks premature convergence, LADE guides reinitialization toward underexplored or diverse regions of the landscape. This structured feedback loop improves both convergence and population diversity, making LADE particularly effective in multimodal and deceptive environments where traditional heuristics may falter.

Complementing landscape-aware design are data-centric techniques that adaptively adjust algorithm behavior based on features extracted during the optimization run. These approaches aim to

replace static control mechanisms with learned or reactive policies. Simple supervised models have been shown to improve the adaptation of threshold control in run time [46]. This data-driven strategy led to better performance than fixed schedules, suggesting that even lightweight learning components can yield substantial gains when integrated into the control loop.

Together, these approaches mark a shift from generic, one-size-fits-all heuristics toward instance-aware, feedback-driven optimization. Landscape-aware algorithms gain insight into the global structure of the search space, while data-centric models offer flexible, real-time control based on evolving signals. As metaheuristics increasingly blend these two perspectives, they become more capable of adapting to the nuances of diverse and challenging optimization problems.

### 5.6. Meta-Black-Box Optimization and Automated Algorithm Design

A unifying trend across many recent advances is the emergence of *Meta-Black-Box Optimization* (Meta-BBO): the idea of not only solving optimization problems but also optimizing the optimizers themselves. This paradigm elevates the design of algorithms into an outer-loop black-box problem—treating algorithm configuration, selection, and even construction as tasks that can be solved by meta-level search.

In a comprehensive survey, Ma et al. [37] classify Meta-BBO tasks into three categories: algorithm configuration, where parameters are tuned for specific tasks; algorithm selection, where the best-performing algorithm is chosen from a portfolio; and algorithm generation, where new optimizers are synthesized altogether. The review catalogs a wide spectrum of approaches, ranging from reinforcement learning and supervised regression to neuroevolution and transformer-based policy learning. It also highlights an emerging role for large-scale pre-trained models and meta-learning pipelines that can generalize across problem families—enabling the creation of self-adaptive, cross-domain optimizers that improve with experience.

This shift reflects a deeper convergence between optimization, AutoML, and program synthesis. As optimization increasingly adopts learning-based tools, the role of the algorithm designer evolves. Rather than manually crafting heuristics, researchers now act as curators of training data, architects of learning objectives, and designers of policy spaces. This reorientation opens up new forms of abstraction and automation in algorithm development—accelerating innovation while reducing reliance on domain-specific expertise.

Ultimately, Meta-BBO recasts the challenge of algorithm design as a learning problem. By embedding optimization knowledge into models that themselves evolve, this approach promises a future where optimizers are not just engineered, but trained—adapting automatically to the structure, dynamics, and complexity of the tasks they are deployed on.

### 5.7. Summary and Outlook

Taken together, these trends signal a transition from static, manually tuned heuristics to adaptive, learning-driven optimizers that respond to the search landscape, performance feedback, and computational constraints. Deep models are now viable both as surrogate predictors and as full optimizers. Reinforcement learning is enabling intelligent online adaptation. Surrogate models are being made interpretable and cost-aware. And general-purpose meta-optimization frameworks are emerging to tie everything together.

As this ecosystem matures, new challenges arise: maintaining generalization across tasks, balancing transparency with learning capacity, and reducing the computational overhead of complex models. Yet the trajectory is clear—machine learning is no longer an add-on to metaheuristics; it is becoming the foundation of next-generation global optimization.

## 6. Open Challenges

Despite the rapid progress and promising results in integrating machine learning with metaheuristics for global optimization, several critical challenges remain unresolved. These challenges reflect both the unique difficulties of global optimization (e.g., high dimensionality, sample inefficiency,

non-stationarity) and the emerging complexities introduced by learning-based enhancements (e.g., generalization, interpretability, data scarcity). In this section, we highlight five key open challenges that, if addressed, could significantly improve the reliability, scalability, and usability of ML-enhanced metaheuristics in practice.

### 6.1. Generalization Across Problem Instances

A central challenge in ML-enhanced global optimization is the ability of learned strategies—whether update rules, control policies, or configuration selectors—to generalize beyond their training distribution. Unlike classical metaheuristics, which rely on manually designed and generally applicable rules, ML-based methods often depend on data from previous runs or representative tasks to guide their learning process. As a result, there is a risk that learned optimizers overfit to specific benchmark families, dimensions, or search space geometries.

This issue is particularly acute in meta-learning and learned optimizer approaches. For example, Lange et al. [101] propose a learned evolution strategy (LES) using a self-attention-based architecture trained via meta-black-box optimization. While LES shows promising generalization to new tasks, dimensions, and population sizes, its success relies heavily on careful meta-training across diverse representative functions. The authors acknowledge that performance can degrade if test problems fall too far outside the meta-training distribution.

Guo et al. [88] tackle a similar challenge from the angle of dynamic algorithm selection using deep reinforcement learning. Their proposed RL-DAS framework learns to switch among differential evolution variants in real time, based on observed landscape and algorithmic features. Although RL-DAS outperforms any single DE variant and exhibits zero-shot generalization across CEC benchmark functions, the authors note that generalization is still constrained by the feature design and diversity of the training instances.

Bolufé-Röhler et al. [77] explores this problem in the context of hybrid metaheuristics, modeling the transition point between exploration and exploitation phases as a supervised learning problem. Even with careful feature engineering, the learned control model risks misclassification when applied to unseen functions or new dimensionalities, limiting the robustness of such adaptive hybridizations.

In short, the generalization of ML-enhanced metaheuristics remains a delicate balancing act between expressivity and overfitting. The tension is particularly pronounced in global optimization, where problems often differ widely in scale, modality, and ruggedness. Even when powerful representation learning methods are used—as in the deep landscape characterization framework by Seiler et al. [69]—generalization can suffer in high-dimensional or noisy settings due to model overfitting or lack of interpretability. Future work is needed to design learning strategies that incorporate robustness, transferability, and domain adaptation mechanisms, potentially by drawing inspiration from meta-reinforcement learning or few-shot learning paradigms.

### 6.2. Sample Efficiency and Computational Overhead

In global optimization problems where objective evaluations are costly or time-consuming, reducing the number of evaluations without sacrificing solution quality is a primary concern. Machine learning-enhanced metaheuristics have tackled this challenge through a variety of strategies, including surrogate modeling, reinforcement learning, and adaptive parameter control. However, these improvements often come with their own computational overhead, raising new questions about trade-offs between evaluation cost, learning cost, and optimization performance.

Jin [17] provides one of the foundational reviews on surrogate-assisted evolutionary computation, describing how surrogate models such as Gaussian processes, radial basis functions, and support vector machines can be integrated into metaheuristics to approximate objective functions and reduce the evaluation burden. The paper also discusses the importance of model management strategies in controlling approximation error and in preventing misleading search behavior. Notably, Jin identified persistent challenges related to scalability in high-dimensional problems, efficient sampling strategies, and the computational cost of maintaining accurate surrogate models.

To address some of these limitations, Yu et al. [39] propose a Two-Stage Dominance-Based Surrogate-Assisted Evolutionary Algorithm (TSDEA) for expensive, high-dimensional multi-objective optimization. Their approach uses radial basis function surrogates to pre-rank candidates and applies an angle-penalized distance metric to balance convergence and diversity. Only the most promising solutions are passed to the exact evaluation phase, significantly reducing the number of function calls. This method exemplifies a careful balance between surrogate precision and computational feasibility in large-scale settings.

Monteiro and Sau [52] offer an alternative to conventional surrogate modeling with their Landscape-Sketch-Step (LSS) algorithm, which employs reinforcement learning to build a state-value approximation of the optimization landscape. Rather than explicitly modeling the objective function, LSS uses interpolated reward signals from past evaluations to guide stochastic search, yielding a lightweight and scalable search mechanism. This surrogate-free strategy emphasizes sample efficiency and demonstrates robust performance on rugged low-dimensional functions, suggesting that full model construction is not always necessary for informed decision-making.

A different perspective is provided by Reijnen et al. [53], who apply deep reinforcement learning to adapt key parameters in Differential Evolution during multi-objective optimization. Their DRL-APC-DE framework learns to adjust scaling factors and crossover probabilities based on optimization state features, thereby improving efficiency without relying on surrogate modeling. While this method enhances performance and adaptability, it also introduces additional computational overhead from policy training and inference, highlighting the ongoing tension between learning sophistication and runtime budget.

In summary, while ML techniques have introduced powerful tools to reduce evaluation cost in global optimization, they also raise new challenges in terms of computational efficiency and model management. Future work may benefit from hybrid strategies that combine surrogate-assisted methods with lightweight, policy-driven control, and from designing metaheuristics that can dynamically balance learning effort with available computational resources.

### 6.3. Interpretability and Trust

As machine learning components become more tightly coupled with metaheuristic-based global optimizers, interpretability and user trust are emerging as first-order design constraints. High-capacity learners, such as deep neural networks, can capture rich regularities in black-box landscapes, yet their opaque reasoning makes it difficult to validate, debug, or gain insight, an issue that becomes acute in safety-critical domains such as engineering design, medical modeling, or scientific discovery.

Seiler et al. [69] systematically evaluated two feature-free deep architectures—a lightweight convolutional network (ShuffleNet V2) and a point-cloud transformer—for characterising single-objective continuous fitness landscapes. Across the BBOB test suite, the deep models delivered performance comparable to (and occasionally slightly below) that of handcrafted exploratory-landscape-analysis features, demonstrating that end-to-end representation learning can match manual descriptors without problem-specific feature design. At the same time, the latent embeddings produced by these networks lack a straightforward semantic interpretation, underscoring an inherent trade-off between representational flexibility and transparency.

To make surrogate decisions more explainable, Li et al. [102] embed SHAP (SHapley Additive exPlanations) analysis within a surrogate-assisted evolutionary algorithm. SHAP values quantify each design variable's contribution to the surrogate's prediction, and this information is fed back into variation operators so that crossover and mutation focus on the most influential dimensions. The result is a method that not only accelerates convergence but also provides actionable, instance-specific explanations of *why* certain variables are being emphasised—thereby increasing practitioner trust.

A complementary perspective adopts simple and transparent models—such as decision trees, support vector machines,  $k$ -nearest neighbours, and Naïve Bayes—to detect phase transitions between exploration and exploitation in adaptive hybrid metaheuristics [77]. Because these classifiers are



readily visualised and audited, their control rules can be inspected, debugged, and ported across problem domains with minimal effort.

Finally, the classic survey of Jin [17] cautions that low-fidelity surrogates may introduce spurious optima or bias the search trajectory if their uncertainty is not properly managed. Although written in the pre-deep-learning era, this warning remains pertinent: delegating critical decisions to black-box learners without safeguards can erode reliability and erode user confidence.

In sum, achieving state-of-the-art optimisation performance with machine-learning-enhanced metaheuristics must go hand in hand with mechanisms that render their behaviour intelligible. Promising directions include hybrid frameworks that couple interpretable surrogate models with post-hoc explanation tools, and modular algorithm designs that expose internal learning signals for selective introspection.

#### 6.4. Benchmarking and Reproducibility

Reliable benchmarking and reproducibility remain central concerns in the evaluation of ML-enhanced metaheuristics for global optimization. However, the increasing algorithmic complexity introduced by learning components—such as surrogate models, adaptive policies, and meta-level controllers—compounds long-standing issues related to evaluation standards, comparability, and result validity.

Bartz-Beielstein et al. [23] provide a comprehensive discussion of benchmarking principles and common pitfalls in optimization research. They emphasize that despite progress in platforms like COCO and BBOB, many studies in evolutionary computation still lack standardized reporting of experimental protocols, fail to distinguish clearly between stochastic variability and structural algorithmic changes, and offer limited information about initialization, termination, or performance measures. The paper argues that benchmarking is often treated as a routine post-processing step rather than an integral methodological component, which can weaken the reliability and generalizability of conclusions.

Kerschke and Trautmann [63] echo these concerns in the context of automated algorithm selection for continuous black-box optimization. Their study demonstrates that subtle differences in instance selection, feature computation (e.g., through Exploratory Landscape Analysis), and function preprocessing can substantially affect performance comparisons. They note that reproducibility is especially challenging when ML-based selectors are involved, since learned models can be sensitive to training noise and benchmark composition.

López-Ibáñez et al. [54] address reproducibility from a complementary perspective: the configuration of optimization algorithms themselves. The irace framework applies a racing-based approach to compare parameter configurations on sets of instances under controlled budgets. The authors discuss the importance of training/testing separation, careful seed control, and repeated evaluation to prevent overfitting and ensure fair comparison. While their work focuses on automatic configuration, the broader implication is that rigorous evaluation practices are essential even before benchmarking comparisons are made.

Lastly, Ma et al. [37] draw attention to emerging issues in meta-black-box optimization (MetaBBO), where learned optimizers are applied across tasks rather than single problem instances. They highlight the difficulty of benchmarking such methods consistently, since performance depends not only on the test functions used but also on the distribution of training tasks and the nature of the meta-level adaptation. The paper points out that few current benchmarks support this level of variability, which limits our ability to evaluate generalization in learned metaheuristics.

Collectively, these works indicate that reproducibility in ML-enhanced global optimization requires more than access to code or data. It demands clear descriptions of algorithmic components and evaluation conditions, consistency across studies, and the use of robust experimental design. These considerations are especially important when comparing hybrid or learned optimizers whose behavior may be data-dependent, history-sensitive, or influenced by random seeds in subtle ways.

## 7. Future Directions

As the intersection of machine learning and metaheuristics continues to evolve, the next breakthroughs will likely emerge from creative recombination, architectural innovation, and a rethinking of foundational assumptions. This section outlines several forward-looking directions that build on the trends and techniques reviewed in Sections 3 and 5, offering high-reward avenues for shaping the next generation of optimization systems.

### 7.1. Neuro-Symbolic Hybridization: Learning with Logic and Structure

While much of the recent progress in ML-enhanced optimization has been driven by deep learning (see Section 3.6), symbolic approaches offer complementary strengths in structure, interpretability, and inductive bias. The integration of neural models with symbolic reasoning—through mechanisms such as logic programming, symbolic regression, or grammar-guided search—could produce metaheuristics capable of abstract reasoning and explainable decision-making. Embedding algebraic constraints or programmatic priors into learned optimizers may yield search strategies that generalize better and offer more transparent control mechanisms.

### 7.2. Foundation Models and Multi-Modal Optimization

Recent work has explored learned optimizers using Transformers and sequence models (Section 5.1), but the broader class of foundation models remains underutilized. These large, pretrained models (e.g., GPT, CLIP) open up possibilities for multi-modal optimization, where textual descriptions, diagrams, or simulation logs guide the search process. Integrating foundation models into metaheuristics could enable zero-shot or few-shot optimization in real-world design tasks, with the model suggesting variation strategies or shaping fitness landscapes based on contextual understanding.

### 7.3. Co-Evolution in Agent-Based Optimization

Building on ideas from adaptive control and dynamic search policies (see Sections 3.1 and 3.3), future work may adopt agent-based models in which populations of optimizers evolve in parallel with candidate solutions. Inspired by ecological dynamics, this paradigm enables emergent search behaviors through competition, cooperation, or specialization. Such systems could support continual learning, dynamic role assignment, and online strategy discovery—offering an alternative to hand-designed or pretrained policies.

### 7.4. Green Optimization and Energy-Aware Learning

The computational cost of intelligent optimizers—especially those using deep models or meta-learning (Section 3.6)—is becoming a practical constraint. Future research will need to consider energy efficiency as a first-class objective. Techniques such as anytime optimization, sparse models, and thermodynamically motivated algorithm design could give rise to *green metaheuristics*: methods that adapt their complexity to budget constraints. This is particularly relevant for embedded systems, edge devices, or climate-conscious computation.

### 7.5. Open Science and Reproducibility as Infrastructure

Challenges in reproducibility and benchmarking (Section 6.4) are increasingly limiting the pace of scientific progress in ML-enhanced metaheuristics. Moving forward, the field would benefit from infrastructure similar to OpenML or Hugging Face, but tailored to optimization—hosting optimizers, benchmark suites, training traces, and diagnostic tools. Such resources would enable standardized comparisons, facilitate reuse, and promote the transition from fragmented experimentation to cumulative, inspectable science.

## 8. Conclusion

This review has explored the rapidly evolving intersection between machine learning and metaheuristic optimization—a convergence that is reshaping how complex, black-box, and resource-

constrained problems are approached. Motivated by the limitations of traditional metaheuristics in dynamic, high-dimensional, and data-scarce settings, researchers are increasingly incorporating learning mechanisms at every level of the optimization process.

We proposed a taxonomy that categorizes machine learning contributions according to their functional role within metaheuristics—ranging from operator selection and parameter control to surrogate modeling, landscape learning, and meta-learning. This framework helps clarify how learning enhances adaptability, scalability, and efficiency across diverse optimization scenarios.

Throughout the paper, we reviewed a wide spectrum of techniques: surrogate models that guide search with predictive accuracy and interpretability; reinforcement learning agents that control variation and hybridize strategies; representation learning techniques that uncover useful structure in decision spaces; and meta-learned optimizers that transfer knowledge across tasks. Emerging paradigms such as diffusion models, foundation models, and neuro-symbolic hybrids signal an expansion into new modes of optimization that blend reasoning, generation, and adaptation.

Beyond performance improvements, these developments challenge long-standing boundaries: between algorithm and model, between offline configuration and online control, and between static heuristics and self-improving systems. As ML-enhanced metaheuristics become more autonomous and context-aware, they open the door to more general and intelligent optimization frameworks.

Realizing this vision will require greater collaboration across communities, along with a sustained focus on benchmarking, reproducibility, and theoretical foundations. With shared infrastructure and principled abstractions, the field can evolve from a collection of isolated heuristics into a cumulative science of adaptive optimization.

In the long run, the most impactful optimizers may not be those with the best hand-crafted rules—but those that learn how to learn, and adapt how to search.

**Funding:** This research received no external funding.

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

## References

1. Floudas, C.A. *Deterministic global optimization: theory, methods and applications*; Vol. 37, Springer Science & Business Media, 2013.
2. Larson, J.; Menickelly, M.; Wild, S.M. Derivative-free optimization methods. *Acta Numerica* **2019**, *28*, 287–404.
3. Gendreau, M.; Potvin, J.Y.; et al. *Handbook of metaheuristics*; Vol. 2, Springer, 2010.
4. Talbi, E.G. *Metaheuristics: from design to implementation*; John Wiley & Sons, 2009.
5. Yang, X.S. Metaheuristic optimization: Nature-inspired algorithms and applications. In *Artificial intelligence, evolutionary computing and metaheuristics: In the footsteps of alan turing*; Springer, 2013; pp. 405–420.
6. Karafotias, G.; Hoogendoorn, M.; Eiben, Á.E. Parameter control in evolutionary algorithms: Trends and challenges. *IEEE Transactions on Evolutionary Computation* **2014**, *19*, 167–187.
7. Blum, C.; Roli, A. Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM computing surveys (CSUR)* **2003**, *35*, 268–308.
8. Chen, S.; Bolufé-Röhler, A.; Montgomery, J.; Tamayo-Vera, D.; Hendtlass, T. Measuring the effects of increasing dimensionality on fitness-based selection and failed exploration. In *Proceedings of the 2022 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2022, pp. 1–8.
9. Chen, S.; Montgomery, J.; Bolufé-Röhler, A. Measuring the curse of dimensionality and its effects on particle swarm optimization and differential evolution. *Applied Intelligence* **2015**, *42*, 514–526.
10. Sergeyev, Y.D.; Kvasov, D.; Mukhametzhayev, M. On the efficiency of nature-inspired metaheuristics in expensive global optimization with limited budget. *Scientific reports* **2018**, *8*, 453.
11. Jin, Y.; Wang, H.; Chugh, T.; Guo, D.; Miettinen, K. Data-driven evolutionary optimization: An overview and case studies. *IEEE Transactions on Evolutionary Computation* **2018**, *23*, 442–458.
12. Talbi, E.G. Machine learning into metaheuristics: A survey and taxonomy. *ACM Computing Surveys (CSUR)* **2021**, *54*, 1–32.
13. Jordan, M.I.; Mitchell, T.M. Machine learning: Trends, perspectives, and prospects. *Science* **2015**, *349*, 255–260.
14. Bengio, Y.; Goodfellow, I.; Courville, A.; et al. *Deep learning*; Vol. 1, MIT press Cambridge, MA, USA, 2017.

15. Karimi-Mamaghan, M.; Mohammadi, M.; Meyer, P.; Karimi-Mamaghan, A.M.; Talbi, E.G. Machine learning at the service of meta-heuristics for solving combinatorial optimization problems: A state-of-the-art. *European Journal of Operational Research* **2022**, *296*, 393–422.
16. Tian, Y.; Li, X.; Ma, H.; Zhang, X.; Tan, K.C.; Jin, Y. Deep reinforcement learning based adaptive operator selection for evolutionary multi-objective optimization. *IEEE Transactions on Emerging Topics in Computational Intelligence* **2022**, *7*, 1051–1064.
17. Jin, Y. Surrogate-assisted evolutionary computation: Recent advances and future challenges. *Swarm and Evolutionary Computation* **2011**, *1*, 61–70.
18. Battiti, R.; Campigotto, P. Reactive search optimization: Learning while optimizing. an experiment in interactive multi-objective optimization. In Proceedings of the Proceedings of MIC. Citeseer, 2009.
19. Elsken, T.; Metzen, J.H.; Hutter, F. Neural architecture search: A survey. *Journal of Machine Learning Research* **2019**, *20*, 1–21.
20. Calvet, L.; Armas, J.d.; Masip, D.; Juan, A.A. Learnheuristics: hybridizing metaheuristics with machine learning for optimization with dynamic inputs. *Open Mathematics* **2017**, *15*, 261–280.
21. Chen, S.; Bolufé-Röhler, A.; Montgomery, J.; Hendtlass, T. An analysis on the effect of selection on exploration in particle swarm optimization and differential evolution. In Proceedings of the 2019 IEEE congress on evolutionary computation (CEC). IEEE, 2019, pp. 3037–3044.
22. Chen, S.; Montgomery, J.; Bolufé-Röhler, A.; Gonzalez-Fernandez, Y. A review of threshold convergence. *GECONTEC: revista Internacional de Gestión del Conocimiento y la Tecnología* **2015**, *3*.
23. Bartz-Beielstein, T.; Doerr, C.; Berg, D.v.d.; Bossek, J.; Chandrasekaran, S.; Eftimov, T.; Fischbach, A.; Kerschke, P.; La Cava, W.; Lopez-Ibanez, M.; et al. Benchmarking in optimization: Best practice and open issues. *arXiv preprint arXiv:2007.03488* **2020**.
24. Eiben, A.E.; Smith, J.E. *Introduction to evolutionary computing*; Springer, 2015.
25. Bäck, T.; Fogel, D.B.; Michalewicz, Z. Handbook of evolutionary computation. *Release* **1997**, *97*, B1.
26. Boussaïd, I.; Lepagnot, J.; Siarry, P. A survey on optimization metaheuristics. *Information sciences* **2013**, *237*, 82–117.
27. Jin, Y.; Olhofer, M.; Sendhoff, B. A framework for evolutionary optimization with approximate fitness functions. *IEEE Transactions on evolutionary computation* **2002**, *6*, 481–494.
28. Seiler, M.V.; Prager, R.P.; Kerschke, P.; Trautmann, H. A collection of deep learning-based feature-free approaches for characterizing single-objective continuous fitness landscapes. In Proceedings of the Proceedings of the Genetic and Evolutionary Computation Conference, 2022, pp. 657–665.
29. Durgut, R.; Aydin, M.E.; Atli, I. Adaptive operator selection with reinforcement learning. *Information Sciences* **2021**, *581*, 773–790.
30. Ming, F.; Gong, W.; Wang, L.; Jin, Y. Constrained multi-objective optimization with deep reinforcement learning assisted operator selection. *IEEE/CAA Journal of Automatica Sinica* **2024**, *11*, 919–931.
31. Guo, D.; Wang, X.; Gao, K.; Jin, Y.; Ding, J.; Chai, T. Evolutionary optimization of high-dimensional multiobjective and many-objective expensive problems assisted by a dropout neural network. *IEEE transactions on systems, man, and cybernetics: systems* **2021**, *52*, 2084–2097.
32. Karimi-Mamaghan, M.; Mohammadi, M.; Pasdeloup, B.; Meyer, P. Learning to select operators in meta-heuristics: An integration of Q-learning into the iterated greedy algorithm for the permutation flowshop scheduling problem. *European Journal of Operational Research* **2023**, *304*, 1296–1330.
33. Tripp, A.; Daxberger, E.; Hernández-Lobato, J.M. Sample-efficient optimization in the latent space of deep generative models via weighted retraining. *Advances in Neural Information Processing Systems* **2020**, *33*, 11259–11272.
34. Tessari, M.; Iacca, G. Reinforcement learning based adaptive metaheuristics. In Proceedings of the Proceedings of the Genetic and Evolutionary Computation Conference Companion, 2022, pp. 1854–1861.
35. Bolufé-Röhler, A.; Xu, B. Deep Reinforcement Learning for Smart Restarts in Exploration-Only Exploitation-Only Hybrid Metaheuristics. In Proceedings of the Metaheuristics International Conference. Springer, 2024, pp. 19–34.
36. Andrychowicz, M.; Denil, M.; Gomez, S.; Hoffman, M.W.; Pfau, D.; Schaul, T.; Shillingford, B.; De Freitas, N. Learning to learn by gradient descent by gradient descent. *Advances in neural information processing systems* **2016**, *29*.
37. Ma, Z.; Guo, H.; Gong, Y.J.; Zhang, J.; Tan, K.C. Toward automated algorithm design: A survey and practical guide to meta-black-box-optimization. *IEEE Transactions on Evolutionary Computation* **2025**.



38. Lange, R.; Schaul, T.; Chen, Y.; Zahavy, T.; Dalibard, V.; Lu, C.; Singh, S.; Flennerhag, S. Discovering evolution strategies via meta-black-box optimization. In Proceedings of the Proceedings of the Companion Conference on Genetic and Evolutionary Computation, 2023, pp. 29–30.
39. Yu, M.; Wang, Z.; Dai, R.; Chen, Z.; Ye, Q.; Wang, W. A two-stage dominance-based surrogate-assisted evolution algorithm for high-dimensional expensive multi-objective optimization. *Scientific Reports* **2023**, *13*, 13163.
40. Khaldi, M.I.E.; Draa, A. Surrogate-assisted evolutionary optimisation: a novel blueprint and a state of the art survey. *Evolutionary Intelligence* **2024**, *17*, 2213–2243.
41. Guo, H.; Qiu, W.; Ma, Z.; Zhang, X.; Zhang, J.; Gong, Y.J. Advancing cma-es with learning-based cooperative coevolution for scalable optimization. *arXiv preprint arXiv:2504.17578* **2025**.
42. Parisi, G.I.; Kemker, R.; Part, J.L.; Kanan, C.; Wermter, S. Continual lifelong learning with neural networks: A review. *Neural networks* **2019**, *113*, 54–71.
43. Eiben, A.; Horvath, M.; Kowalczyk, W.; Schut, M.C. Reinforcement learning for online control of evolutionary algorithms. In Proceedings of the Engineering Self-Organising Systems: 4th International Workshop, ESOA 2006, Hakodate, Japan, May 9, 2006, Revised and Invited Papers 4. Springer, 2007, pp. 151–160.
44. Tatsis, V.A.; Ioannidis, D. Online Cluster-Based Parameter Control for Metaheuristic. *arXiv preprint arXiv:2504.05144* **2025**.
45. Bolufé-Röhler, A.; Han, W. A data-centric approach to parameter tuning, an application to differential evolution. In Proceedings of the 2023 IEEE Congress on Evolutionary Computation (CEC). IEEE, 2023, pp. 1–9.
46. Bolufé-Röhler, A.; Luke, J. A data-centric machine learning approach for controlling exploration in estimation of distribution algorithms. In Proceedings of the 2022 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS). IEEE, 2022, pp. 1–9.
47. Johnn, S.N.; Darvariu, V.A.; Handl, J.; Kalcsics, J. A graph reinforcement learning framework for neural adaptive large neighbourhood search. *Computers & Operations Research* **2024**, *172*, 106791.
48. Pei, J.; Mei, Y.; Liu, J.; Zhang, M.; Yao, X. Adaptive Operator Selection for Meta-Heuristics: A Survey. *IEEE Transactions on Artificial Intelligence* **2025**.
49. Lim, D.; Jin, Y.; Ong, Y.S.; Sendhoff, B. Generalizing surrogate-assisted evolutionary computation. *IEEE Transactions on Evolutionary Computation* **2009**, *14*, 329–355.
50. Chung, I.B.; Park, D.; Choi, D.H. Surrogate-based global optimization using an adaptive switching infill sampling criterion for expensive black-box functions. *Structural and Multidisciplinary Optimization* **2018**, *57*, 1443–1459.
51. Bertsimas, D.; Margaritis, G. Global optimization: a machine learning approach. *Journal of Global Optimization* **2025**, *91*, 1–37.
52. Monteiro, R.; Sau, K. Landscape-Sketch-Step: An AI/ML-Based Metaheuristic for Surrogate Optimization Problems. *arXiv preprint arXiv:2309.07936* **2023**.
53. Reijnen, R.; Zhang, Y.; Bukhsh, Z.; Guzek, M. Deep reinforcement learning for adaptive parameter control in differential evolution for multi-objective optimization. In Proceedings of the 2022 IEEE Symposium Series on Computational Intelligence (SSCI). IEEE, 2022, pp. 804–811.
54. López-Ibáñez, M.; Dubois-Lacoste, J.; Cáceres, L.P.; Birattari, M.; Stützle, T. The irace package: Iterated racing for automatic algorithm configuration. *Operations Research Perspectives* **2016**, *3*, 43–58.
55. Huang, C.; Li, Y.; Yao, X. A survey of automatic parameter tuning methods for metaheuristics. *IEEE transactions on evolutionary computation* **2019**, *24*, 201–216.
56. Birattari, M.; Stützle, T.; Paquete, L.; Varrentapp, K.; et al. A Racing Algorithm for Configuring Metaheuristics. In Proceedings of the Gecco. Citeseer, 2002, Vol. 2.
57. Hutter, F.; Hoos, H.H.; Leyton-Brown, K.; Stützle, T. ParamILS: an automatic algorithm configuration framework. *Journal of artificial intelligence research* **2009**, *36*, 267–306.
58. Hutter, F.; Hoos, H.H.; Leyton-Brown, K. Sequential model-based optimization for general algorithm configuration. In Proceedings of the Learning and intelligent optimization: 5th international conference, LION 5, rome, Italy, January 17–21, 2011. selected papers 5. Springer, 2011, pp. 507–523.
59. Falkner, S.; Klein, A.; Hutter, F. BOHB: Robust and efficient hyperparameter optimization at scale. In Proceedings of the International conference on machine learning. PMLR, 2018, pp. 1437–1446.
60. Xu, L.; Hutter, F.; Hoos, H.H.; Leyton-Brown, K. SATzilla: portfolio-based algorithm selection for SAT. *Journal of artificial intelligence research* **2008**, *32*, 565–606.

61. Kadioglu, S.; Malitsky, Y.; Sellmann, M.; Tierney, K. ISAC–instance-specific algorithm configuration. In *ECAI 2010*; IOS Press, 2010; pp. 751–756.
62. Xu, L.; Hoos, H.; Leyton-Brown, K. Hydra: Automatically configuring algorithms for portfolio-based selection. In *Proceedings of the Proceedings of the AAAI Conference on Artificial Intelligence, 2010, Vol. 24*, pp. 210–216.
63. Kerschke, P.; Trautmann, H. Automated algorithm selection on continuous black-box problems by combining exploratory landscape analysis and machine learning. *Evolutionary computation* **2019**, *27*, 99–127.
64. Lindauer, M.; Hoos, H.H.; Hutter, F.; Schaub, T. Autofolio: An automatically configured algorithm selector. *Journal of Artificial Intelligence Research* **2015**, *53*, 745–778.
65. Mersmann, O.; Bischl, B.; Trautmann, H.; Preuss, M.; Weihs, C.; Rudolph, G. Exploratory landscape analysis. In *Proceedings of the Proceedings of the 13th annual conference on Genetic and evolutionary computation, 2011*, pp. 829–836.
66. Muñoz, M.A.; Kirley, M.; Halgamuge, S.K. Exploratory landscape analysis of continuous space optimization problems using information content. *IEEE transactions on evolutionary computation* **2014**, *19*, 74–87.
67. Malan, K.M.; Engelbrecht, A.P. Quantifying ruggedness of continuous landscapes using entropy. In *Proceedings of the 2009 IEEE Congress on evolutionary computation. IEEE, 2009*, pp. 1440–1447.
68. Malan, K.M. A survey of advances in landscape analysis for optimisation. *Algorithms* **2021**, *14*, 40.
69. Seiler, M.V.; Prager, R.P.; Kerschke, P.; Trautmann, H. A collection of deep learning-based feature-free approaches for characterizing single-objective continuous fitness landscapes. In *Proceedings of the Proceedings of the Genetic and Evolutionary Computation Conference, 2022*, pp. 657–665.
70. Malan, K.M. Landscape-aware constraint handling applied to differential evolution. In *Proceedings of the Theory and Practice of Natural Computing: 7th International Conference, TPNC 2018, Dublin, Ireland, December 12–14, 2018, Proceedings 7*. Springer, 2018, pp. 176–187.
71. Karp, S.; Saunshi, N.; Miryoosefi, S.; Reddi, S.J.; Kumar, S. Landscape-Aware Growing: The Power of a Little LAG. *arXiv preprint arXiv:2406.02469* **2024**.
72. Chen, T.; Chen, X.; Chen, W.; Heaton, H.; Liu, J.; Wang, Z.; Yin, W. Learning to optimize: A primer and a benchmark. *Journal of Machine Learning Research* **2022**, *23*, 1–59.
73. Shala, G.; Biedenkapp, A.; Awad, N.; Adriaensen, S.; Lindauer, M.; Hutter, F. Learning step-size adaptation in CMA-ES. In *Proceedings of the Parallel Problem Solving from Nature–PPSN XVI: 16th International Conference, PPSN 2020, Leiden, The Netherlands, September 5–9, 2020, Proceedings, Part I 16*. Springer, 2020, pp. 691–706.
74. Yang, Q.; Chu, S.C.; Pan, J.S.; Chou, J.H.; Watada, J. Dynamic multi-strategy integrated differential evolution algorithm based on reinforcement learning for optimization problems. *Complex & Intelligent Systems* **2024**, *10*, 1845–1877.
75. Zhao, F.; Zhou, H.; Xu, T.; et al. A self-learning differential evolution algorithm with population range indicator. *Expert Systems with Applications* **2024**, *241*, 122674.
76. Chen, M.; Feng, C.; Cheng, R. MetaDE: Evolving Differential Evolution by Differential Evolution. *IEEE Transactions on Evolutionary Computation* **2025**.
77. Bolufé-Röhler, A.; Yuan, Y. Machine learning for determining the transition point in hybrid metaheuristics. In *Proceedings of the 2021 IEEE Congress on Evolutionary Computation (CEC). IEEE, 2021*, pp. 1115–1122.
78. Szénási, S.; Légrádi, G. Machine learning aided metaheuristics: A comprehensive review of hybrid local search methods. *Expert Systems with Applications* **2024**, *258*, 125192.
79. Nomura, M.; Akimoto, Y.; Ono, I. Cma-es with learning rate adaptation. *ACM Transactions on Evolutionary Learning* **2025**, *5*, 1–28.
80. Han, D.; Du, W.; Wang, X.; Du, W. A surrogate-assisted evolutionary algorithm for expensive many-objective optimization in the refining process. *Swarm and Evolutionary Computation* **2022**, *69*, 100988.
81. Wang, Y.; Zhang, Y. A K-means Clustering-Based hybrid offspring generation mechanism in Evolutionary Multi-Objective Optimization. *IEEE Access* **2021**, *9*, 167642–167651.
82. Zhang, Y.; Li, Z.; Zhang, H.; Yu, Z.; Lu, T. Fuzzy c-means clustering-based mating restriction for multiobjective optimization. *International Journal of Machine Learning and Cybernetics* **2018**, *9*, 1609–1621.
83. Wang, Z.J.; Zhan, Z.H.; Lin, Y.; Yu, W.J.; Wang, H.; Kwong, S.; Zhang, J. Automatic niching differential evolution with contour prediction approach for multimodal optimization problems. *IEEE Transactions on Evolutionary Computation* **2019**, *24*, 114–128.
84. Tsang, W.W.; Lau, H.Y. Clustering-based multi-objective immune optimization evolutionary algorithm. In *Proceedings of the International Conference on Artificial Immune Systems*. Springer, 2012, pp. 72–85.

85. Lim, S.M.; Sultan, A.B.M.; Sulaiman, M.N.; Mustapha, A.; Leong, K.Y. Crossover and mutation operators of genetic algorithms. *International journal of machine learning and computing* **2017**, *7*, 9–12.
86. Eiben, A.; Horvath, M.; Kowalczyk, W.; Schut, M.C. Reinforcement learning for online control of evolutionary algorithms. In Proceedings of the International Workshop on Engineering Self-Organising Applications. Springer, 2006, pp. 151–160.
87. von Eschwege, D.; Engelbrecht, A. Soft Actor-Critic Approach to Self-Adaptive Particle Swarm Optimisation. *Mathematics* **2024**, *12*, 3481.
88. Guo, H.; Ma, Y.; Ma, Z.; Chen, J.; Zhang, X.; Cao, Z.; Zhang, J.; Gong, Y.J. Deep reinforcement learning for dynamic algorithm selection: A proof-of-principle study on differential evolution. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* **2024**, *54*, 4247–4259.
89. Bentley, P.J.; Lim, S.L.; Gaier, A.; Tran, L. COIL: Constrained optimization in learned latent space: Learning representations for valid solutions. In Proceedings of the Proceedings of the Genetic and Evolutionary Computation Conference Companion, 2022, pp. 1870–1877.
90. Wang, X.; Jin, Y.; Schmitt, S.; Olhofer, M. Transfer learning based co-surrogate assisted evolutionary bi-objective optimization for objectives with non-uniform evaluation times. *Evolutionary computation* **2022**, *30*, 221–251.
91. Lim, J.; Jang, Y.S.; Chang, H.S.; Park, J.C.; Lee, J. Multi-objective genetic algorithm in reliability-based design optimization with sequential statistical modeling: an application to design of engine mounting. *Structural and Multidisciplinary Optimization* **2020**, *61*, 1253–1271.
92. Smith-Miles, K.A. Cross-disciplinary perspectives on meta-learning for algorithm selection. *ACM Computing Surveys (CSUR)* **2009**, *41*, 1–25.
93. Durgut, R.; Aydin, M.E.; Rakib, A. Transfer learning for operator selection: A reinforcement learning approach. *Algorithms* **2022**, *15*, 24.
94. Lange, R.; Tian, Y.; Tang, Y. Evolution transformer: In-context evolutionary optimization. In Proceedings of the Proceedings of the Genetic and Evolutionary Computation Conference Companion, 2024, pp. 575–578.
95. Song, L.; Gao, C.; Xue, K.; Wu, C.; Li, D.; Hao, J.; Zhang, Z.; Qian, C. Reinforced in-context black-box optimization. *arXiv preprint arXiv:2402.17423* **2024**.
96. Bharath, S.; Vasuki, A. Adaptive energy loss optimization in distributed networks using reinforcement learning-enhanced crow search algorithm. *Scientific Reports* **2025**, *15*, 12165.
97. Li, B.; Yang, Y.; Liu, D.; Zhang, Y.; Zhou, A.; Yao, X. Accelerating surrogate assisted evolutionary algorithms for expensive multi-objective optimization via explainable machine learning. *Swarm and Evolutionary Computation* **2024**, *88*, 101610.
98. Li, Z.; Yuan, H.; Huang, K.; Ni, C.; Ye, Y.; Chen, M.; Wang, M. Diffusion model for data-driven black-box optimization. *arXiv preprint arXiv:2403.13219* **2024**.
99. Krishnamoorthy, S.; Mashkaria, S.M.; Grover, A. Diffusion models for black-box optimization. In Proceedings of the International Conference on Machine Learning. PMLR, 2023, pp. 17842–17857.
100. Lin, G.Y.; Chen, Z.G.; Liu, C.; Jiang, Y.; Kwong, S.; Zhang, J.; Zhan, Z.H. A Landscape-Aware Differential Evolution for Multimodal Optimization Problems. *IEEE Transactions on Evolutionary Computation* **2025**.
101. Lange, R.T.; Schaul, T.; Chen, Y.; Zahavy, T.; Dallibard, V.; Lu, C.; Singh, S.; Flennerhag, S. Discovering Evolution Strategies via Meta-Black-Box Optimization, 2023, [[arXiv:cs.NE/2211.11260](https://arxiv.org/abs/cs.NE/2211.11260)].
102. Li, B.; Yang, Y.; Liu, D.; Zhang, Y.; Zhou, A.; Yao, X. Accelerating Surrogate Assisted Evolutionary Algorithms Via Explainable Machine Learning. *Available at SSRN 4699560* **2024**.

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