Evolutionary computation and explainable AI: a roadmap to understandable intelligent systems

Zhou, R.; Bacardit, J.; Brownlee, A. E.I.; Cagnoni, S.; Fyvie, M.; Iacca, G.; ... ; Hu, T.

Evolutionary Computation and Explainable AI: A Roadmap to Understandable Intelligent Systems

Ryan Zhou, Jaume Bacardit, Alexander Edward Ian Brownlee, Stefano Cagnoni, *Senior Member, IEEE*, Martin Fyvie, Giovanni Iacca, *Senior Member, IEEE*, John McCall, *Member, IEEE*, Niki van Stein, *Member, IEEE*, David Walker, and Ting Hu

***Abstract*—Artificial intelligence methods are being increasingly applied across various domains, but their often opaque nature has raised concerns about accountability and trust. In response, the field of explainable AI (XAI) has emerged to address the need for human-understandable AI systems. Evolutionary computation (EC), a family of powerful optimization and learning algorithms, offers significant potential to contribute to XAI, and vice versa. This article provides an introduction to XAI and reviews current techniques for explaining machine learning (ML) models. We then explore how EC can be leveraged in XAI and examine existing XAI approaches that incorporate EC techniques. Furthermore, we discuss the application of XAI principles within EC itself, investigating how these principles can illuminate the behavior and outcomes of EC algorithms, their (automatic) configuration, and the underlying problem landscapes they optimize. Finally, we discuss open challenges in XAI and highlight opportunities for future research at the intersection of XAI and EC. Our goal is to demonstrate EC's suitability for addressing current explainability challenges and to encourage further exploration of these methods, ultimately contributing to the development of more understandable and trustworthy ML models and EC algorithms.**

***Index Terms*—Evolutionary computation (EC), explainability, interpretability, machine learning (ML).**

I. Introduction

THE PROLIFERATION of AI has brought with it an increasing need to understand the reasoning behind its outputs and decisions. While AI methods can learn complex relationships in data and provide solutions to challenging problems, they are often driving decisions that can have significant real-world impacts. The use of predictive models in medicine, hiring, and the justice system has raised concerns about fairness and transparency, and the growing adoption of large language models in commercial products has heightened the importance of avoiding harmful content. Similarly, the application of optimization in areas, such as scheduling and logistics [1] requires users to have a robust grasp of the system's operations, as they remain accountable for any adverse outcomes. Consequently, it is crucial not only to improve our models and algorithms but also to understand and explain the factors driving their prediction or optimization decisions. While active research is ongoing to improve the fairness and safety of AI, this survey focuses on the latter challenge: understanding and explaining AI systems.

Recent AI advancements have heavily relied on "black-box" approaches. Deep learning, ensemble models, and stochastic optimization algorithms may have well-defined structures, but the processes leading to their decisions are often too complex for human comprehension. In response to this challenge, the field of explainable XI (XAI) has emerged [2].

XAI is an umbrella term encompassing research on methods designed to improve human understanding of AI systems' decisions and knowledge capture. It aims to develop techniques that explain AI's decisions, predictions, or recommendations in human understandable terms. These explanations foster trust, improve the system robustness by highlighting potential biases and failures, and provide researchers with insights to better understand, validate, and debug systems effectively. Moreover, they play a pivotal role in ensuring regulatory compliance and enhancing human–machine interactions, allowing users to better discern when they can rely on an AI system's conclusions.

Evolutionary computation (EC) is a powerful approach to AI, with algorithms capable of tackling both optimization and machine learning (ML) tasks. In the context of EC, two directions associated with XAI emerge: 1) the application of XAI principles to decision making within EC and 2) the use of EC to enhance explainability within ML, where the majority of XAI research is currently focused. A growing body of work is developing in both areas, partly fueled by events, such as the workshop on EC and XAI held at GECCO in 2022–2024.

The aim of this article is to provide a critical review of research conducted at the intersection of EC and XAI. We

Received 11 June 2024; revised 19 September 2024; accepted 29 September 2024. Date of publication 23 October 2024; date of current version 8 October 2025. The work of Ting Hu was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) under Discovery Grant RGPIN-2023-03302. This article was approved by Associate Editor M. I. Heywood. *(Corresponding author: Ryan Zhou.)*

Ryan Zhou and Ting Hu are with the School of Computing, Queen's University, Kingston, ON K7L 3N6, Canada (e-mail: ryan.zhou@queensu.ca).

Jaume Bacardit is with the School of Computing Science, Newcastle University, NE1 7RU Newcastle upon Tyne, U.K.

Alexander Edward Ian Brownlee is with the Department of Computing Science and Mathematics, University of Stirling, FK9 4LA Stirling, U.K.

Stefano Cagnoni is with the Dipartimento di Ingegneria e Architettura, University of Parma, 43121 Parma, Italy.

Martin Fyvie is with the National Subsea Centre, Robert Gordon University, AB10 7QB Aberdeen, U.K.

Giovanni Iacca is with the Department of Information Engineering and Computer Science, Universita degli Studi di Trento, 38122 Trento, Italy.

John McCall is with the IDEAS Research Institute, Robert Gordon University, AB10 7QB Aberdeen, U.K.

Niki van Stein is with the Leiden Institute of Advanced Computer Science, Universiteit Leiden, 2300 RA Leiden, The Netherlands.

David Walker is with the Department of Computer Science, University of Exeter, EX4 4QJ Exeter, U.K.

Digital Object Identifier 10.1109/TEVC.2024.3476443

present a taxonomy of methods and highlight potential avenues for future work, expanding on initial directions proposed in the field of EC [3], [4], [5].

The remainder of this article is structured as follows. Section II introduces the foundational concepts in XAI, such as the nature of explanations and the distinctions between interpretability and explainability, and provides motivation for strengthening the link between XAI and EC. Section III discusses how EC can be used *for* XAI. Section IV examines how XAI can be *applied to* EC. Section V addresses the ongoing challenges and potential opportunities. Section VI provides our final thoughts and conclusions.

II. Explainable AI

XAI aims to improve the *understandability* of AI systems–the degree to which humans can comprehend how a system makes decisions, the reasoning behind those decisions, and their potential implications. Importantly, understandability is subjective and can vary from user to user, depending on their background, experience, and familiarity with the AI system.

XAI employs two general approaches: 1) designing algorithms or models that are easier to understand without external aids and 2) providing explanations which aid understanding by illuminating an AI system's output process, highlighting significant features and interactions, or revealing potential issues. Even if an AI system is too complex for direct human comprehension, it can be considered *explainable* if it can be understood with the help of these explanations.

Explainability is crucial for several reasons as follows.

1) *Trust:* Explainability directly influences users' willingness to adopt and rely on AI results [6]. For ML models, it allows users to understand the decision-making process. For optimization, it demonstrates why obtained solutions are reliable.
2) *Validity:* Explanations can reveal whether a solution truly solves the problem or merely exploits an error in the problem definition or a spurious data relationship. This helps avoid surprising or frustratingly incorrect results [7].
3) *Real-World Applicability:* Explanations can reveal important characteristics for optimality, allowing refinement of solutions to better fit real-world problems. This is particularly useful when subtle rules or preferences are difficult to codify in the initial problem definition.
4) *Regulatory Compliance:* As AI legislation increases, explanations may provide necessary audit trails for implemented solutions.
5) *Bias Detection:* Explanations can help identify unwanted biases in ML predictions, especially when goals like "fairness" are not explicitly coded in the training cost function.

A. What Is Explanation?

Defining what constitutes an explanation is challenging. Informally, an explanation aims to answer the question: "why?". Prior work has framed explanations in various ways, including providing causal information [8], noncausal explanations [9], or deductive arguments [10]. In this article, we define an *explanation* as a tool to help humans understand certain aspects of a model or an algorithm. The ultimate purpose of an explanation is to serve as an interface between the model/algorithm and the user, delivering information in a more accessible form. Importantly, an explanation need not capture the full behavior of the model/algorithm but should communicate important insights about it.

Such insights may include the answers to questions, such as [3] follows.

1) Is the model solving the correct problem, and has the problem been formulated correctly?
2) What are the patterns the model uses for predictions, and are they as expected?
3) Why did the model make this prediction instead of another, and what would change its prediction?
4) Is the model biased, and are its decisions fair?

Explanations can take on multiple forms, including visualizations, numerical outputs, data instances, or text descriptions [2]. They may also be part of an ongoing dialogue between a human and an explainer [8], [11].

B. Explainability and Interpretability

The terms interpretability and explainability are often used interchangeably, but we distinguish them as related yet distinct aspects of understanding a model [12], [13]. Similar to understandability, both explainability and interpretability are subjective and depend on the user's knowledge and experience.

Interpretability refers to a human's ability to follow a model's decision-making process without external aids. Simple models, like small decision trees or symbolic representations, are considered interpretable. However, as models grow in size or complexity (e.g., random forests and neural networks), they become harder to follow, aligning with the notion that interpretability exists on a spectrum [12]. Simpler models may sacrifice accuracy for ease of understanding, while more complex ones, though more accurate, are less interpretable.

Explainability, on the other hand, refers to the ability to provide human-understandable insights into a model's decisions, even if the exact logic is too complex to trace. Explanations do not need to capture the model's full behavior; instead, they offer glimpses into how it works, using methods like feature importance, local approximations, or input comparisons.

As shown in Fig. 1, the more complex a system, the greater the effort required to understand it. Below a certain threshold, models are intrinsically interpretable. Beyond that, explanations are needed to aid understanding. For example, multidimensional models may become understandable with visual aids or feature importance metrics. As the complexity increases further, at some point it becomes impractical to explain the model fully. Explanations of large language models, for example, may still leave key behaviors unexplained. However, this points toward two ways of achieving explainability, which can work in tandem: make the model simpler to understand or improve our explanation techniques.

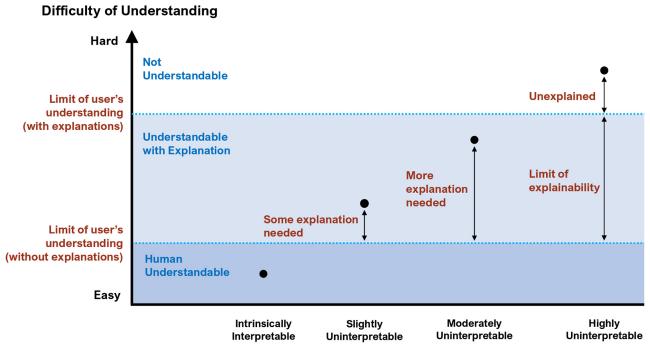


Fig. 1. As models and solutions become more difficult to understand, the amount of explanation required increases. Simple solutions (the left point) may not require any explanation at all and are intrinsically interpretable. Others (the middle two points) may lie beyond the ability of a human to grasp easily, but can be understood with explanation. Finally, some models (the right point) may remain incomprehensible even with the current best efforts at explanation.

C. Why EC and XAI?

EC is an AI approach inspired by biological evolution, with applications in optimization, ML, engineering design, and artificial life. EC encompasses evolutionary algorithms, such as genetic algorithms (GAs), genetic programming (GP), and evolution strategies (ESs), and extends to swarm intelligence algorithms like particle swarm optimization. These techniques typically use populations of solutions and operators that introduce variation and diversity to explore large regions of the search space.

EC techniques possess unique strengths that can address current challenges in XAI [14], [15]. First, as detailed in later sections, EC has a proven track record of creating symbolic or interpretable models (e.g., decision trees or rule-based systems). By constructing solutions from intrinsically interpretable components, EC-derived solutions can ensure the interpretability of the resulting models. Additionally, EC can generate interpretable approximations of more complex models, producing explanations for their behavior.

Second, the inherent flexibility of evolutionary methods, such as their ability to perform derivative-free, black-box optimization,[1] makes them versatile tools for scenarios where other methods struggle. For instance, EC can optimize models accessible only through APIs that provide predictions without revealing internal logic. This flexibility also enables EC to handle customized metrics, such as interpretability metrics, that are not easily optimized via gradient descent. Furthermore, EC can be combined with other algorithms to create hybrid methods or meta-optimizers.

A particularly valuable feature of EC is multiobjective optimization, crucial in XAI where there is often a tradeoff between the model accuracy and human interpretability or complexity of the explanation. EC can balance these objectives, and by leveraging diversity metrics or quality-diversity algorithms, it can generate a variety of explanations tailored to different users or aspects of the model.

Conversely, XAI approaches can offer valuable insights into evolutionary algorithms and are currently underutilized in EC. XAI can help explain the decision-making process of EC algorithms, making it easier to debug and refine them. This is especially important in fields like engineering design or policy making, where the rationale behind a solution must be understandable to nontechnical decision makers.

Finally, XAI can enhance the interpretability of fitness landscape analyses in EC. Understanding the fitness landscape is critical for assessing the difficulty of finding optimal solutions and the effectiveness of EC algorithms. XAI-inspired visualization and interpretation tools can provide deeper insights into these landscapes, serving as explanations in their own right.

III. EC for XAI

This section covers XAI methods for ML, and the incorporation of evolutionary algorithms into such methods. As ML models have become more advanced, their complexity has increased, often boosting performance but at the cost of interpretability. Improving explainability is essential to balance this tradeoff, ensuring models are not only effective but remain understandable.

A. Explainability and Complexity

The interpretability of a model is influenced by its complexity [16]. Simpler models, like linear regressions, are generally considered to be interpretable due to their straightforward decision-making processes that humans can follow unaided. An ML model is typically deemed intrinsically interpretable when it is compact and understandable. However, as model complexity grows, interpretability diminishes and we must rely on explainability.

This raises a question: why pursue explainability over creating an intrinsically interpretable model that requires no explanation? While there are reasons to favor interpretable models when possible [13], such models are not always feasible. Moreover, even when a more complex model fits the data well, researchers warn against assuming that the reason for this is that there are underlying interpretable rules which the model has learned [17]. In such cases, post-hoc explanations provide the best path to understanding. They may not always be comprehensive, but provide insights that allow us to gain at least a partial understanding of complex models.

For any given problem, there is a minimum level of complexity required in the model to accurately model the data. If this level is low, a simple, interpretable model can suffice. However, when the problem demands more complexity, a complex model becomes necessary, and explainability is required. In either case, the complexity of the problem determines the complexity of the model. This relationship between problem complexity and model complexity is important for understanding when we need explainability.

To illustrate this, we present a framework in Fig. 2, mapping problem complexity against model complexity. The complexity of a model (or an optimization solution) includes factors,

[1]Black-box optimization refers to methods that handle problems where the internal structure, equations, or derivatives of the objective function are unknown or inaccessible. In this context, black-box means the optimizer can work with just the input–output pairs, which is distinct from the use of black-box to describe complex, incomprehensible ML models.

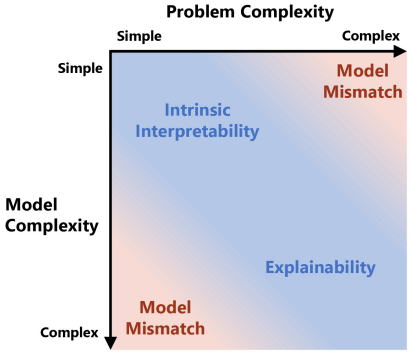


Fig. 2. Interaction between problem complexity and model complexity. Simple models are inherently interpretable and suitable for simple problems, but may not capture the full behavior of complex problems. When complex models are needed for complex problems, explainability becomes essential to understand the model's behavior.

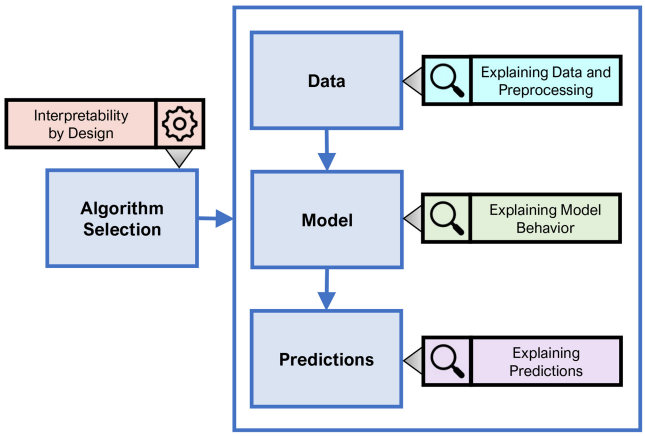


Fig. 3. Overview of the process of building an ML model, showing areas where explanations (magnifying glasses) are often applied. Also shown is the intrinsic interpretability approach (cogwheel), where the models are designed to be interpretable from the start. All these methods can be used together to form a complete picture of a model's behavior.

such as the number of parameters, depth of structure, and computational requirements. Although we treat this informally, this may be quantified through metrics, such as the model's description length [18] or parameterized complexity [16]. A model with a lengthy description, numerous parameters, or complex functions is considered more complex. While increased complexity can improve problem-solving capacity, it often reduces interpretability simply due to the model's size. For instance, neural networks with billions of parameters or deep decision trees may perform well but are difficult to interpret.

For problem complexity, we adopt an informal analogue of Kolmogorov complexity [19]. A problem's complexity is defined as the complexity of the simplest model required to capture its behavior at the desired level of accuracy. As problem complexity increases, so does the complexity of the model needed to represent it. Some problems may appear simple if we are satisfied with an approximation, but become complex when aiming for greater accuracy. Similar to how Kolmogorov complexity can only be approximated due to the undecidability of the halting problem, this notion of the problem complexity does not assume we can identify the least complex model definitively, only that it exists.

With these two axes, problem complexity and model complexity, we can identify four key scenarios as follows.

1) *Simple Problem, Simple Model:* A simple model captures the desired behavior. This model is both accurate and intrinsically interpretable, eliminating the need for explainability.
2) *Simple Problem, Complex Model:* Using a complex model for a simple problem creates a mismatch. While the model may perform well, it is unnecessarily complex and difficult to interpret. In this case, the problem is not one of explainability issue but of the model selection: a simpler model which requires no explanation would suffice. However, if such a model exists but cannot be found in practice, explainability may still be used to gain understanding.
3) *Complex Problem, Simple Model:* Applying a simple model to a complex problem results in inadequate performance. Although the model is interpretable, it fails to capture the complexity of the data to the desired degree, leading to a mismatch between the problem and the model. This can be addressed by using a more complex model (which then requires explanation), or if we are satisfied with a less accurate solution, lowering our requirements and reducing the problem to a simple one which can be solved by a simple model.
4) *Complex Problem, Complex Model:* This is the primary area of relevance for XAI. Complex models are required to solve complex problems, but are difficult to interpret. Explainability methods are essential to help users understand these models, since the models cannot be understood on their own and cannot be made intrinsically interpretable while still solving the problem adequately.

B. Types of Explanations

Explanations target various aspects of the modeling process. Here, we take a problem-focused approach, examining the entire ML pipeline, from the data to trained model (Fig. 3). Our categorization is based on the stage of the ML pipeline where explainability can be applied. This approach emphasizes that explainability is not only relevant at the end of the process, once a model is fully trained but can also enhance understanding throughout the model-building process. By aligning explanations with the pipeline stages, we aim to offer practitioners a clear roadmap for where and how explainability can be integrated.

In the following sections, we address each stage in turn. First, we introduce each category by describing examples of conventional approaches. This overview is not exhaustive but provides a primer on commonly used methods. For a more comprehensive review of current XAI methods, we refer readers to recent surveys on the topic [20], [21]. We then explore EC-based approaches within each stage, offering an overview of the state-of-the-art in combining EC and XAI.

C. Interpretability by Design

As alluded to in Section III-A, a growing consensus in the XAI community advocates for developing interpretable ML models whenever possible, rather than relying on the post-hoc explanations [2], [13]. The argument is that the post-hoc explanations often only provide local approximations, which are limited in two key ways: 1) they rarely capture the entire decision-making process of a model and 2) being approximations of another model, they can introduce errors, potentially distorting the original decision-making logic. As a result, models built to be inherently interpretable and grounded in knowledge representation are preferable when possible.

Unlike traditional ML models that often use greedy heuristics, EC methods leverage global optimization through evolutionary search. A notable example of this is learning classifier systems (LCSs), which apply either batch [22], [23] or online learning [24], [25], using reinforcement learning (RL) [24] or supervised learning [26]. While most LCS approaches generate rule-based models, alternative representations, such as decision trees [27] and hyperellipsoids [28] have also been explored. GP, as another example, also has a long history of producing interpretable, symbolic solutions to learning problems [29], [30].

To control model complexity, EC methods employ techniques to manage model size, such as minimizing bloat [31] and using fitness functions that encourage the compact rule sets based on the principles like minimum description length (MDL) [32]. Multiobjective optimization [33] and rule-editing operators [34], [35] are also used to simplify models. Sparsity in neural networks, for example, can be achieved through regularization or evolutionary pruning [36], [37].

Efforts to combine RL and EC have aimed to produce interpretable policies by using decision trees induced by GP or grammatical evolution, with RL guiding actions [38], [39]. Studies also explore quality-diversity methods [40], [41] and multiagent settings [42]. The balance between accuracy and interpretability has also been studied in genetic fuzzy systems [43], while recent work has proposed machine-learned measures of interpretability [44] and emphasized the importance of low-complexity models, particularly in GP [45].

Visualization techniques, such as heatmaps, have been effective for interpreting classification rules generated by LCS [46], particularly when combined with hierarchical clustering. Similarly, 3-D visualizations have proven useful for representing rule sets, illustrating attributes, rule generality, and estimated attribute importance [47].

D. Explaining Data and Preprocessing

We begin by discussing methods aimed at explaining the data, focusing on understanding its structure before modeling. These techniques, although not applied to the final model, are nonetheless crucial to the ML pipeline. Every model begins with data, and any patterns the model learns are derived from this data. While these methods do not explain the model itself, they provide insight into the data distribution and characteristics that shape model learning.

Exploratory analysis, data visualization, and dimensionality reduction techniques, such as principal component analysis (PCA) [48], [49] and t-distributed stochastic neighbor embedding (t-SNE) [50], help uncover patterns and potential biases in the data. These methods simplify data for easier interpretation and visualization.

Additionally, clustering and outlier detection techniques, like k-means and DBSCAN [51], identify patterns or anomalies that can affect the model performance and inform feature selection. These explanations can help identify data quality issues, biases, and preprocessing requirements.

1) Dimensionality Reduction: EC can aid in explaining data through dimensionality reduction and visualization. One approach is GP-tSNE [52], which adapts the t-SNE [50] algorithm by using evolved trees to create an interpretable mapping from original data points to embedded points. Similarly, a study [53] uses tree-based GP to generate an interpretable mapping for uniform manifold approximation and projection (UMAP) [54]. These explicit mapping functions make the process more understandable and reusable for new data.

In some cases, lower-dimensional representations are useful for both prediction and visualization. This enhances interpretability, allowing us to visualize the exact data representation seen by the model. For example, a multiobjective GP algorithm [55] is developed to optimize features for classifiability, visual interpretability, and semantic interpretability. Another study optimizes features for both visualization and downstream tasks, balancing classification metrics (accuracy, AUC, and Cohen's kappa) with visualization metrics (C-index, Davies-Bouldin, and Dunn's index) to improve clustering and separability [56].

GP has also been applied to manifold learning [57], creating reduced representations for high-dimensional datasets. While traditional black-box algorithms often lack transparency in how they map data to a reduced space, GP trees offer interpretable, Glass-box alternatives for these transformations.

2) Feature Selection and Feature Engineering: Feature selection is a common preprocessing step that selects a relevant subset of features from the original dataset. This improves both the model performance and interpretability by limiting the features a model can rely on. While similar to feature importance, which identifies key features, feature selection explicitly restricts the model to the chosen subset.

GAs are a natural and effective approach to feature selection (since solutions can represented as binary strings), and are widely used in this area [58], [59], [60]. GP is also often used, as feature selection is inherently part of the evolved program structure [61], [62], [63]. For a detailed review of GP methods in feature selection, see [64]. Swarm intelligence methods, such as particle swarm optimization, are another effective technique [65], with further discussion available in [66]. In addition to the model improvement, feature selection aids in understanding data when combined with clustering techniques [67].

Feature engineering or feature construction, creates higher-level features from basic ones. GP is well-suited for evolving these features for tasks like classification and regression [68], [69], [70], [71]. This process can enhance interpretability

by reducing the number of low-level features into more meaningful, higher-level ones that are easier to understand.

Feature engineering shares similarities with dimensionality reduction and, in some cases, overlaps with it. A study on various multitree GP algorithms for dimensionality reduction [72] demonstrates that GP-based methods perform comparably to traditional techniques.

E. Explaining Model Behavior

Once a model is trained, understanding how it functions can still be challenging, even if we have access to its internal mechanisms. For example, being able to examine the weights of a neural network does not necessarily make the overall model more interpretable. In these cases, explanations help bridge the gap by providing insights into how the model operates. These methods aim to explain the model's internal structure either at the global level or for specific subcomponents.

1) Feature Importance: Global feature importance explains a model's dependence on each feature by assigning a score that reflects the significance of each feature to the model's predictions. This helps identify which features the model is using and whether it aligns with human expectations. This technique can also aid in model optimization and feature selection by highlighting less important features. Some models, like decision trees and random forests, offer built-in feature importance measures [73], while others use more general methods like partial dependence plots [74] and permutation feature importance [75]. EC can further refine feature importance by measuring interactions between features, evolving groups that reveal higher-order interactions [76].

2) Global Model Approximations: Global model approximations, also known as model extraction or global surrogates, approximate a black-box model with a simpler, interpretable one. This concept is related to knowledge distillation in deep learning [77], [78], but with the added goal of enhancing interpretability. One approach to this is the use of interpretable decision sets to approximate a model's behavior [79]. GP is well-suited for the model extraction [80], as it can evolve decision trees that replicate the predictions of a black-box model while minimizing complexity. This method preserves accuracy while producing more interpretable models compared to other extraction techniques.

3) Domain-Specific Knowledge Extraction: In specific domains, EC has been used to extract meaningful knowledge from ML models. For example, classification rules evolved by EC methods have been applied to protein structure prediction [81]. Similarly, EC-based techniques have been used to infer biological functional networks [82], leading to experimentally validated gene discoveries in plants [83]. Knowledge representations in rule-based systems can also shape the patterns captured, resulting in different insights from the same data, as demonstrated in molecular biology [84]. In neuro-evolution, EC has been applied to discover interpretable plasticity rules [85], [86], as well as self-interpretable agents that rely on selective inputs [87].

4) Explaining Neural Networks: While we have focused thus far on methods which apply to a variety of models, deep learning requires specific techniques due to the complexity and black-box nature of the models. Explaining these models is exceptionally difficult due to the large number of parameters.

In image classification, the large number of input features (pixels) poses a problem for many explanation methods. Dimensionality reduction techniques, like clustering pixels into "superpixels," help identify important regions for predictions. Multiobjective algorithms can optimize for minimal superpixels while maximizing the model confidence [88].

Efforts to explain internal representations [89] include methods that create invertible mappings from complex latent spaces to simpler, interpretable ones, offering insight into how the models process information [90].

Recently, "mechanistic interpretability" has gained attention, aiming to reverse-engineer neural networks by analyzing activation patterns to reveal the underlying algorithms [91], [92]. This approach has explained phenomena like "grokking," where the models initially memorize data but later generalize after extended training [93].

EC is well-suited for explaining neural networks, notably through the ability to construct small interpretable explanations and to prioritize exploration. EC methods have been used to map out decision boundaries and input spaces in language models [94], [95], and symbolic regression has provided interpretable mathematical expressions that describe network gradients [96].

F. Explaining Predictions

This approach focuses on explaining a specific prediction rather than the model's overall behavior. The explanation only needs to capture how the model arrived at the particular prediction in question.

1) Local Explanations: Local explanation approaches aim to approximate a model's behavior for a specific prediction rather than explaining the model globally. One widely used method, local interpretable model-agnostic explanations (LIMEs) [97], generates data points near the input and fits a linear model to approximate the black-box model's behavior in that local region. While this approach does not capture the model's global performance, it provides a locally faithful explanation for one data point.

An alternative approach, GP explainer (GPX) [98], uses GP to evolve symbolic expressions that better capture local patterns than LIME's linear approximation. GPX constructs a local explanation by sampling neighboring data points and evolving a model that reflects the behavior of the black-box model more effectively, particularly when linearity assumptions do not hold.

Another example, local rule-based explanations (LOREs) [99], introduces an evolutionary algorithm to generate a neighborhood of points around the prediction. These points are classified either similarly or differently from the original prediction, and a decision tree is used to capture the local behavior. The evolutionary algorithm ensures a dense and diverse set of neighborhood points, allowing the decision tree to provide a more robust local explanation.

In addition to approximating a local model, Shapley additive explanations (SHAP) [100] offers a way to assess feature importance for individual predictions by approximating the Shapley values from the game theory. These values represent each feature's contribution to the prediction, with SHAP using efficient sampling techniques to make this computation feasible for practical use.

2) Counterfactuals: Counterfactual explanations provide insight by offering hypothetical scenarios where the model would make a different decision. For example, "the model would have approved the loan if the income were $5000 higher" explains how a change in input affects the model's output [101]. These explanations are intuitive, model-agnostic, and provide users with actionable steps, or recourse, to achieve desired outcomes [102]. However, since they focus on single instances, they offer limited insight into the model's global behavior.

Diverse counterfactual explanations (DiCEs) [103] generates counterfactuals that are valid (produce a different outcome), proximal (similar to the original input), and diverse (distinct from each other). Diversity enhances the usefulness of explanations by offering multiple perspectives on how the model behaves. DiCE achieves this using determinantal point processes [104], which balance proximity and diversity while ensuring that only a few features differ from the original input.

EC is well suited for generating counterfactuals thanks to its black-box optimization capabilities and ability to handle multiple objectives. CERTIFAI [105] applies a GA to generate counterfactual explanations by sampling instances on the opposite side of the model's decision boundary. The GA then optimizes the population of counterfactuals by minimizing their distance from the original input instance. This approach measures robustness (based on how far counterfactuals are from the input) and fairness (comparing robustness across different feature values). GeCo [106] also uses a GA, but with additional constraints to ensure plausibility, such as avoiding unrealistic changes (e.g., altering age or gender). The algorithm prioritizes proximity to the decision boundary and minimizes feature changes to simplify the explanation. Multiobjective counterfactuals (MOCs) [107] explicitly optimizes for multiple criteria using a modified NSGA-II algorithm. MOC balances four objectives: 1) achieving the desired model output; 2) maintaining proximity to the input; 3) minimizing feature changes; and 4) ensuring plausibility (measured by distance to real data points). The use of mixed integer ESs (MIESs) [108] allows MOC to search both discrete and continuous spaces efficiently.

3) Adversarial Examples: Adversarial examples are closely related to counterfactuals but are designed to intentionally produce incorrect predictions [109]. These examples are created by applying small perturbations to inputs, changing their classification while keeping them perceptually similar to the original. Adversarial examples highlight failure modes in models and serve as potential attack vectors, especially for deep learning systems.

One approach [110] generates adversarial examples by modifying just a single pixel in an image. This contrasts with previous methods that altered multiple pixels and were more noticeable. Using differential evolution, the method optimizes the pixel coordinates and perturbation in RGB space, demonstrating that a single pixel is often enough to fool the model. Other works have explored ESs [111], differential evolution [112], multiobjective evolutionary algorithms [113], and the clonal selection algorithm [114] to generate adversarial image perturbations.

Adversarial examples are not limited to image models. In natural language processing, adversarial examples have been generated for sentiment analysis and textual entailment models [115]. In this case, adversarial inputs are semantically and syntactically similar to the original, making the attack harder to detect. A GA optimizes the input for a target label, with mutations changing words to their nearest neighbors in a word embedding model and removing words to ensure the context remains intact.

G. Assessing Explanations

In addition to generating explanations, EC can be used to assess or enhance the quality of other explanation methods. One study [116] proposes two metrics to evaluate the robustness of an explanation: 1) worst-case misinterpretation discrepancy and 2) probabilistic interpretation robustness. Interpretation discrepancy measures the difference between two interpretations, before and after input perturbation. A robust interpretation should have a low discrepancy. A GA is used to optimize two worst-case scenarios: 1) the largest discrepancy when the classification remains unchanged and 2) the smallest discrepancy when the classification changes (adversarial example). The second metric, probabilistic misinterpretation, estimates the likelihood of significant interpretation changes under these conditions using subset simulation.

EC can also be applied to adversarial attacks on explanations themselves. One such method, AttaXAI [117], evolves images that appear similar to the original input with the same model prediction but an arbitrary explanation map. Experiments show that, using pairs of images, AttaXAI can create a new image with the same appearance and prediction as the first image, while the explanation map resembles that of the second.

IV. XAI for EC

In this section, we shift the focus to explainability for EC and optimization methods in general. Similar to the previous discussion, optimization algorithms often involve lengthy, complex processes to find optimal or near-optimal solutions for decision makers. Explanations in this context help answer the overarching questions introduced in Section II. We view the optimization process (Fig. 4) as comprising four key stages: 1) algorithm selection; 2) parameter tuning; 3) iterative search; and 4) solution analysis. Each stage presents opportunities for explainability, which we will elaborate on in the following sections.

A. Interpretability by Design

A key challenge in optimization lies in designing and formulating objectives and solution representations. Interpretability plays a crucial role in these design choices, as the way

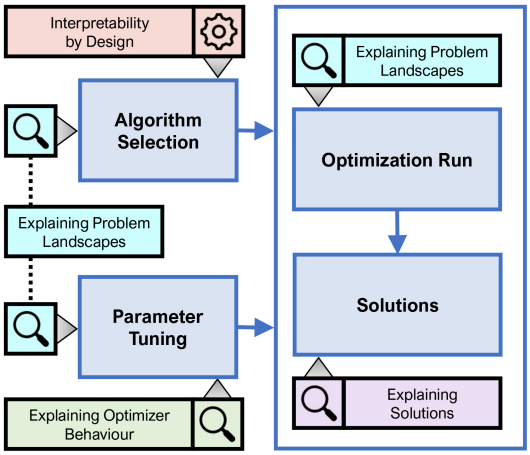


Fig. 4. Overview of the process of using an optimization algorithm, showing areas where the explanations (magnifying glasses) can be applied.

a problem is defined directly impacts how it is explained. To improve interpretability, more direct representations and explicit encoding of variables, objectives, and constraints may be preferred. For example, a mixed-integer linear programming (MILP) formulation, whether solved by mathematical optimization or EC, is preferred than a black-box function evaluation. Matheuristics [118] have proven successful in this area. Another approach [119] uses decision trees to provide interpretable rules for selecting solutions, with trees constructed by MILP or heuristics.

Handling components of an objective separately, rather than combining them, allows for post-hoc analysis of the evolutionary process. This motivates the use of lexicographical approaches to tournament selection, such as those proposed for constraints [120] and multiple objectives [121] or lexicase selection in GP [122]. Multiobjective problems can also be made more understandable using post-hoc multiobjective evolutionary algorithms [123], [124], [125] that approximate a Pareto front, enabling decision makers to better understand tradeoffs between objectives, rather than guessing through weighted sums. This topic will be revisited in Section IV-C.

Direct representations are often favored for explainability, as they align more closely with real-world decision variables, making explanations easier in applied settings. However, there is a tradeoff: indirect representations, such as hyperNEAT [126], [127] and grammatical evolution, often outperform direct representations like classical GP trees [128]. Direct formulations allow for greater control of operators, tailored to the problem at the hand. For example, gray-box optimization [129] leverages domain knowledge with direct encodings in combinatorial problems to improve performance. As with ML (see Section III-D2), representations must balance granularity and interpretability. Overly detailed representations become too dense to interpret, while high-level abstractions may lose real-world relevance.

There may also be opportunities for evolutionary algorithms to select or engineer interpretable features for optimization problems. An example on this research idea is achieved through cooperative coevolution, a technique already successful in large-scale global optimization [130].

The choice of algorithmic framework also affects explainability. Greedy algorithms or steepest ascent hill-climbers, being deterministic, provide a single, easily traceable path, making them more interpretable than stochastic or population-based algorithms. Estimation of distribution algorithms [131], [132], [133], [134] construct explicit problem representations and clear mathematical routes to solutions, although these processes can become complex and harder to interpret.

B. Explaining Problem Landscapes

Landscape analysis focuses on understanding the interactions between algorithms, their operators, and solution representations. While this approach emphasizes understanding *how* the search proceeds, rather than *why* specific solutions are chosen, both aspects are crucial for improving explainability in optimization.

1) Landscape Analysis and Trajectories: Landscape analysis [135] is a key intersection between XAI and EC, offering tools to understand algorithm behavior based on problem features, predict performance, and optimize algorithm selection and configuration. Recent works have focused on explainable landscape-aware predictions [136], [137].

An algorithm's behavior can be described by its trajectory through the search space, as the sequence of points visited during a run. This trajectory captures insightful information, such as when solutions are discovered, when the algorithm converges prematurely, or gets stuck in a local optimum. Search trajectory networks [138] visualize these paths, helping to explain the algorithm's progress for various problem-algorithm combinations.

Search trajectory analysis has also been suggested as a promising technique for XAI in EC [139]. By applying PCA to solutions, it becomes possible to capture dominant features in each generation, visualize algorithm progress, and relate components to known global optima. Other work [140] proposes using simple descriptive statistics to characterize optimization trajectories, which can then be used in ML methods for performance prediction or automatic algorithm configuration.

Population dynamics plots [141] visualize EA progress and convergence behavior by tracing the lineage of solutions and their proximity to feasibility boundaries. These visualizations are especially useful in multiobjective problems like knapsack optimization, projecting multidimensional solutions into interpretable 2-D forms.

Another approach involves creating surrogate fitness models biased toward solutions visited during the evolutionary search [142], [143]. Probing these models reveals insights into variable sensitivity and intervariable relationships, offering another lens through which to study the algorithm's trajectory.

Research into hyperheuristics [144] and parameter selection [145] highlights certain parameter configurations that enable an evolutionary algorithm to perform well across diverse functions. Exploring whether simpler, more explainable parameter settings can also lead to generalist solutions could be a promising area of research, aligning with principles like Occam's razor.

2) User-Guided Evolution: Allowing users to interact with the model-building process can enhance explainability. One approach [146] combines parallel coordinate plots with a multiobjective EA, enabling users to define areas of interest where they want solutions.

Quality-diversity or illumination algorithms, like MAP-Elites [147], [148], offer another way to explore solution landscapes. These algorithms generate diverse, high-quality solutions along user-defined dimensions, helping users understand how solution quality varies with respect to different parameters. A future direction could involve designing algorithms that provide human-interpretable explanations throughout the process, incorporating user feedback during the search, similar to preference-based multiobjective optimization [149].

The efficiency of MAP-Elites can be improved through hybrid approaches [148] using a surrogate model and an intelligent sampling of fitness evaluation. Another application of MAP-Elites [150] addresses the lack of user involvement. By filtering the solution space and offering users a set of solutions to choose from, users can gain influence over what constitutes a "good" solution. More recently, MAP-Elites has been extended to extract explainable rules from its archives [151]. This work addresses the challenge of interpreting thousands of solutions by using GP and rule induction to generate a small set of rules that describe the characteristics of the solutions produced by the optimizer.

C. Explaining Solutions

The solutions generated by optimization, whether they are Pareto fronts, populations, or single solutions, can be examined for explanatory insights. This post-hoc analysis explores alternative causes to explain solution quality and reveal underlying aspects of the model and the algorithm.

1) Interpreting Solutions: Interpretability is related to the concept of *backbones* [152] in optimization, which represent critical components of a solution. For example, in a satisfiability decision problem, a backbone consists of literals that are true in every model. Identifying these features in a solution can provide an explanation for its quality.

Dimensionality reduction techniques are helpful in explaining optimizer solutions. For instance, using multiple correspondence analysis (MCA) can decompose search trajectories, projecting them into lower-dimensional spaces to highlight feature importance at different stages of the search [153]. This helps interpreting the influences that impact the quality of solutions in single-objective problems.

In multiobjective optimization, the tradeoff between a solution's explainability and its accuracy has been explored [154]. By applying step-wise regularization to linear regression models generated by an optimizer, the complexity of explanation representations is reduced while maintaining predictive ability and interpretability.

The concept of innovisation [155], [156] aims to identify shared design principles in multiobjective solutions, explaining Pareto-optimality by highlighting key principles. More recently, efforts to maintain coherence between Pareto-front solutions have been explored [157], offering a smoother view of transitions in the solution space between solutions in Pareto-front approximations.

2) Visualization of Solutions: In many-objective optimization, a challenge is to visualize Pareto-front approximation with more than three objectives, as human cognition struggles with comprehending higher-dimensional spaces. Visualization efforts have focused on three approaches: 1) techniques that display solutions in terms of all objectives; 2) identifying and discarding redundant objectives to allow for standard visualization methods; and 3) using feature extraction to create new, easier-to-visualize coordinate sets.

The first approach includes techniques like parallel coordinate plots [158], [159] and heatmaps [160], which are widely used to visualize the large datasets. However, both suffer from clarity issues: parallel coordinate plots can obscure solutions due to overlap, and heatmaps often have arbitrary ordering of rows and columns, making relationships between solutions and objectives difficult to interpret. Improvements, such as reordering objectives to highlight tradeoffs [161] or using clustering techniques [160], [162] have helped clarify these visualizations. Interactive features in parallel coordinate plots [159] also reduce cognitive load by allowing users to filter out irrelevant solutions.

The second and third approaches involve dimensionality reduction and feature extraction techniques like PCA [163], self-organizing maps (SOMs) [164], and multidimensional scaling (MDS) [165], which project objective vectors from $\mathbb{R}^{M>3}$ into lower-dimensional spaces ($\mathbb{R}^{M\in 2,3}$). This allows for the use of standard visualization tools like scatter plots. However, these projections can disconnect decision makers from the original objectives, potentially causing confusion. Improvements like varying color schemes based on objectives or annotating projected solutions with key information (e.g., best/worst solutions) [162] help mitigate this issue. Further improvements [166] have focused on identifying and visualizing the edges of the Pareto front to better illustrate distances to extreme solutions in lower-dimensional spaces.

D. Explaining Optimizer Behavior

The analysis of optimizers is closely related to landscape analysis, as researchers seek to distinguish between the effects of an optimizer's internal mechanics and the influence of the search landscape. One approach uses special functions, such as the f_0 function [167], a uniform random fitness function, or constant functions, to repeatedly assess behavioral patterns and observe the distribution of final solutions. Another method uses large and diverse benchmark sets or gradually alters benchmark function properties using affine combinations [168], [169].

Behavior-based benchmarks offer deeper insights into the dynamics of metaheuristics. The bias toolbox [170], [171] analyzes structural bias (SB) in optimization algorithms, which refers to intrinsic biases in iterative optimization algorithms that drive search toward certain regions of the solution space, independent of the objective function. The toolbox helps identify the presence, intensity, and nature of SB, while recent

work [172] applies deep learning and XAI techniques to detect and analyze SB patterns, shedding light on the algorithm improvement areas.

The concept of "explainable benchmarking" [173] introduces a framework and its software that dissects the performance of optimization algorithms by analyzing their components and hyperparameters. Using TreeSHAP and other XAI techniques, the framework visualizes the contribution of each component to overall performance on various types of objective functions. Similarly, f-ANOVA [174] helps quantify which modular components of an algorithm contribute most to its optimization performance, providing insights into the relationship between the algorithm configuration, problem landscape characteristics, and algorithm performance.

Another method for explaining algorithm behavior involves comparing historical benchmarking experiments and unifying their results using ontologies [175]. The OPTION ontology [176] provides a semantic vocabulary for annotating algorithms, problems, and evaluation metrics, improving interoperability and reasoning. Recent work [177] extends OPTION to represent modular black-box optimization algorithms and builds knowledge graphs to predict performance based on modular algorithm configurations, leading to explainable predictions.

V. Research Outlook

The works discussed above are not exhaustive, and as the field of XAI continues to grow rapidly, we anticipate more studies exploring the intersection of EC and XAI. In particular, we expect an increasing number of hybrid systems that combine EC-induced interpretable models with black-box models for feature extraction and data manipulation. Such combinations could leverage the strengths of both approaches, fully exploiting the exploration capabilities unique to EC.

A. Challenges

One of the main challenges for evolutionary approaches to XAI, and for XAI in general, is scalability. As data and ML models become more complex, the number of parameters and features to be optimized increase as well. Methods that work well on small models and datasets may become computationally expensive when scaled up on larger ones. Yet, large models are the most incomprehensible and in need of explanation, making scalability crucial for applying XAI methods to more complex models. In particular, producing fully interpretable global explanations that accurately capture the model behavior while being simple enough to understand becomes more challenging as models scale up, necessitating local explanations or a focus on explaining specific properties or components of the model.

EC offers a promising automated approach to explainability, using evolutionary search to find local explanations and to optimize specific properties. While counterfactual examples have explored this concept, it could be extended to other explanation types. Evolutionary ML has proposed various scaling mechanisms [178] that could be adapted for EC-based XAI methods.

Another challenge is incorporating domain knowledge into XAI. Current XAI methods are typically broad and problem-agnostic, but integrating subject matter expertise or prior knowledge can improve explanation quality. For example, evaluating how well a genomics model aligns with existing gene associations can help validate its results. Domain knowledge can be incorporated through expert rules, constraints, or structured data like graphs. It can also improve interpretability by constraining models to focus on plausible associations or exclude irrelevant features.

EC methods are well-suited for leveraging domain knowledge for building better models due to 1) their global search capabilities with robust and complex optimization; 2) their potential for hybridization with local search mechanisms tailored to exploit domain knowledge; and 3) their flexibility in exploration mechanisms to use domain knowledge.

B. Opportunities

We see several opportunities for future research using EC for XAI. One promising direction is the use of multiple objectives to optimize explanations. Explainability is inherently a multiobjective problem, as explanations must be faithful to the ML model while remaining simple enough to be interpretable. EC is well-suited for this, offering a framework for balancing these objectives. Incorporating multiobjective optimization into explanation methods could significantly enhance explanation quality.

The strength of EC on searching for diverse and novel solutions also presents opportunities for improving explanations. Quality-diversity algorithms can generate a range of explanations that offer different perspectives on a model's behavior. For example, applying this approach to counterfactual explanations could showcase a variety of model behaviors. Previous work [151], [179], [180] has demonstrated the explanatory value of search space illumination, but there remains potential for new methods that better interpret and analyze solution sets to support decision making.

Incorporating user feedback is another promising direction, for both EC and XAI. Explanations are meant for human users, making their quality subjective and dependent on individual preferences. By integrating user feedback into the evolutionary process, explanations can become more tailored and continuously improved. Additionally, developing better metrics for evaluating explanation quality is essential to avoid overwhelming users. Future research could explore designing new operators and algorithms that explicitly generate explanations as part of the search process.

Finally, innovations in visualization, interactivity, and sensitivity analysis will further enrich both EC and XAI. Advanced visualization techniques can help users better understand complex solution spaces and relationships between variables. Interactive tools that let users adjust parameters offer deeper insights into model behavior, while sensitivity analysis reveals how changes in inputs affect outcomes. Together, these methods improve user understanding by highlighting key features and making explanations more personalized and intuitive.

C. Real-World Impacts

As AI becomes increasingly integrated into real-world applications, developing effective AI explanation methods is critical. It is equally important to consider the practical benefits that XAI research can bring. Here, we highlight a few application areas where evolutionary approaches to XAI can have a significant impact.

Healthcare is an especially high-stake domain. Without explanations, incomprehensible models may be ignored by clinicians, wasting resources, while flawed models may cause harm to patients. Even models with few errors may exhibit systematic biases, such as underdiagnosing certain patient groups [181]. Explainability can help identify these errors and biases [182].

In the financial sector, AI models are widely used for fraud detection and risk assessment. Systematic biases in these models can also be harmful, such as disproportionately denying loans to certain groups. Additionally, regulatory requirements often mandate explainability of AI systems to ensure compliance and transparency.

Explainability has the potential to drive advancements in engineering and scientific discovery. AI is used in fields like materials design, drug discovery, and genomics, where explanations can uncover underlying mechanisms, support hypothesis generation, and validate domain knowledge.

In natural language processing, foundation models, generalist deep learning models trained on the vast datasets and fine tuned for specific tasks, have advanced rapidly and been deployed widely [183]. These models can perform tasks beyond what they are specifically trained for, but how they make decisions or generate outputs remain unclear. Any errors or biases in these models may be propagated to application-specific models built on the top of them. As foundation models become more pervasive, understanding their behavior and identifying failure modes will be crucial for ensuring their reliability in applications.

VI. Conclusion

We have demonstrated a strong mutual connection between EC and XAI. However, several research opportunities remain underexplored, including: 1) developing tools, whether analytical, visual, data-driven, or model-based, to explain EC methods, their internal functioning, results, and properties/settings/instances that make an algorithm suitable for a given task; 2) defining how EC solutions should be verified and the level of problem knowledge needed to interpret them; and 3) leveraging EC's strengths to provide effective explanations or to evolve interpretable models by design. Another key area for exploration is the relationship between XAI and neuroevolution or neural architecture search, particularly regarding the link between optimized architectures and explainability (e.g., smaller networks may be easier to explain).

As XAI continues to grow, its importance for AI cannot be overstated. With the increasing deployment of ML and optimization systems in real-world applications, understanding these intelligent systems and their learned behaviors is more critical than ever. EC is well-positioned to contribute. In this article, we explored various paradigms for explaining ML models and how EC can fit within these frameworks. EC excels at optimizing difficult interpretability metrics, handling non-differentiability, population-based diversity, and multiobjective optimization, offering distinct advantages for XAI. While some methods have begun leveraging these strengths, much more remains to be explored.

References

[1] Z.-G. Chen, Z.-H. Zhan, S. Kwong, and J. Zhang, "Evolutionary computation for intelligent transportation in smart cities: A survey," *IEEE Comput. Intell. Mag.*, vol. 17, no. 2, pp. 83–102, May 2022.

[2] W. J. Murdoch, C. Singh, K. Kumbier, R. Abbasi-Asl, and B. Yu, "Interpretable machine learning: Definitions, methods, and applications," *Proc. Nat. Acad. Sci.*, vol. 116, no. 44, pp. 22071–22080, 2019.

[3] J. Bacardit, A. E. I. Brownlee, S. Cagnoni, G. Iacca, J. McCall, and D. Walker, "The intersection of evolutionary computation and explainable AI," in *Proc. Genet. Evol. Comput. Conf. Compan.*, Jul. 2022, pp. 1757–1762.

[4] Y. Mei, Q. Chen, A. Lensen, B. Xue, and M. Zhang, "Explainable Artificial Intelligence by genetic programming: A survey," *IEEE Trans. Evol. Comput.*, vol. 27, no. 3, pp. 621–641, Jun. 2023.

[5] R. Zhou and T. Hu, "Evolutionary approaches to explainable machine learning," in *Handbook of Evolutionary Machine Learning*, W. Banzhaf, P. Machado, and M. Zhang, Eds., Singapore: Springer Nat., 2024, pp. 487–506.

[6] T. Miller, "Are we measuring trust correctly in explainability, interpretability, and transparency research?" Aug. 2022, *arXiv:2209.00651*.

[7] J. Lehman, J. Clune, and D. Misevic, "The surprising creativity of digital evolution: A collection of anecdotes from the evolutionary computation and artificial life research communities," 2020, *arXiv:1803.03453*.

[8] T. Miller, "Explanation in artificial intelligence: Insights from the social sciences," *Artif. Intell.*, vol. 267, pp. 1–38, Feb. 2019.

[9] C. Ginet, "In defense of a non-causal account of reasons explanations," *J. Ethics*, vol. 12, pp. 229–237, Sep. 2008.

[10] H. Veatch, "Carl G. Hempel aspects of scientific explanation and other essays in the philosophy of science. New York: The Free Press, 1965. 505 pp," *Philosophy Sci.*, vol. 37, no. 2, pp. 312–314, Jun. 1970.

[11] D. Slack, S. Krishna, H. Lakkaraju, and S. Singh, "Explaining machine learning models with interactive natural language conversations using TalkToModel," *Nat. Mach. Intell.*, vol. 5, no. 8, pp. 873–883, Aug. 2023.

[12] Z. C. Lipton, "The mythos of model interpretability," *Commun. ACM*, vol. 61, no. 10, pp. 36–43, 2018.

[13] C. Rudin, "Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead," *Nat. Mach. Intell.*, vol. 1, pp. 206–215, May 2019.

[14] A. B. Arrieta et al., "Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI," *Inf. Fusion*, vol. 58, pp. 82–115, Jun. 2020.

[15] T. Hu, "Genetic programming for interpretable and explainable machine learning," in *Genetic Programming Theory and Practice XIX*, L. Trujillo, S. M. Winkler, S. Silva, and W. Banzhaf, Eds., Singapore: Springer Nat., 2023, pp. 81–90.

[16] P. Barceló, M. Monet, J. Pérez, and B. Subercaseaux, "Model interpretability through the lens of computational complexity," in *Advances in Neural Information Processing Systems*, vol. 33. Red Hook, NY, USA: Curran Associates, Inc., 2020, pp. 15487–15498.

[17] U. Hasson, S. A. Nastase, and A. Goldstein, "Direct fit to nature: An evolutionary perspective on biological and artificial neural networks," *Neuron*, vol. 105, no. 3, pp. 416–434, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S089662731931044X

[18] G. E. Hinton and D. Van Camp, "Keeping the neural networks simple by minimizing the description length of the weights," in *Proc. 6th Annu. Conf. Comput. Learn. Theory*, 1993, pp. 5–13.

[19] B. Rylander, T. Soule, and J. Foster, "Computational complexity, genetic programming, and implications," in *Proc. 4th Eur. Conf. Genetic Program.*, 2001, pp. 348–360.

[20] R. Dwivedi et al., "Explainable AI (XAI): Core ideas, techniques, and solutions," *ACM Comput. Surveys*, vol. 55, no. 9, p. 194, Jan. 2023.

[21] W. Saeed and C. Omlin, "Explainable AI (XAI): A systematic meta-survey of current challenges and future opportunities," *Knowl.-Based Syst.*, vol. 263, Mar. 2023, Art. no. 110273.

[22] J. Bacardit, "Pittsburgh genetics-based machine learning in the data mining era: Representations, generalization, and run-time," Ph.D. dissertation, Dept. Comput. Sci., Ramon Llull Univ., Barcelona, Spain, 2004.

[23] J. Bacardit, E. K. Burke, and N. Krasnogor, "Improving the scalability of rule-based evolutionary learning," *Memet. Comput.*, vol. 1, pp. 55–67, Mar. 2009.

[24] S. W. Wilson, "Classifier fitness based on accuracy," *Evol. Comput.*, vol. 3, no. 2, pp. 149–175, 1995.

[25] J. H. Holland et al., "What is a learning classifier system?" in *Proc. Int. Workshop Learn. Classifier Syst.*, 1999, pp. 3–32.

[26] E. Bernadó-Mansilla and J. M. Garrell-Guiu, "Accuracy-based learning classifier systems: Models, analysis and applications to classification tasks," *Evol. Comput.*, vol. 11, no. 3, pp. 209–238, 2003.

[27] X. Llora and J. M. Garrell, "Evolution of decision trees," in *Proc. Catalan Conf. Artif. Intell.*, 2001, pp. 115–122.

[28] M. V. Butz, "Kernel-based, ellipsoidal conditions in the real-valued XCS classifier system," in *Proc. Genet. Evol. Comput. Conf.*, 2005, pp. 1835–1842.

[29] W. La Cava et al., "Contemporary symbolic regression methods and their relative performance," 2021, *arXiv:2107.14351*.

[30] R. J. Smith and M. I. Heywood, "Interpreting tangled program graphs under partially observable Dota 2 invoker tasks," *IEEE Trans. Artif. Intell.*, vol. 5, no. 4, pp. 1511–1524, Apr. 2024.

[31] W. B. Langdon, "Fitness causes bloat in variable size representations," School Comput. Sci., Univ. Birmingham, Birmingham, U.K., Rep. CSRP-97-14, 1997.

[32] J. Bacardit and J. M. Garrell, "Bloat control and generalization pressure using the minimum description length principle for a Pittsburgh approach learning classifier system," in *Proc. Int. Workshop Learn. Classifier Syst.*, 2007, pp. 59–79.

[33] X. Llora, D. E. Goldberg, I. Traus, and E. Bernadó, "Accuracy, parsimony, and generality in evolutionary learning systems via multiobjective selection," in *Proc. 5th Int. Workshop Learn. Classifier Syst.*, 2003, pp. 118–142.

[34] J. Bacardit and N. Krasnogor, "Performance and efficiency of memetic Pittsburgh learning classifier systems," *Evol. Comput. J.*, vol. 17, no. 3, pp. 307–342, 2009.

[35] M. A. Franco, N. Krasnogor, and J. Bacardit, "Post-processing operators for decision lists," in *Proc. 14th Annu. Conf. Genet. Evol. Comput.*, 2012, pp. 847–854.

[36] H. Shang, J.-L. Wu, W. Hong, and C. Qian, "Neural network pruning by cooperative coevolution," May 2022, *arXiv:2204.05639*.

[37] R. Zhou and T. Hu, "Evolving better initializations for neural networks with pruning," in *Proc. Compan. Conf. Genet. Evol. Comput.*, Jul. 2023, pp. 703–706.

[38] D. Hein, S. Udluft, and T. A. Runkler, "Interpretable policies for reinforcement learning by genetic programming," *Eng. Appl. Artif. Intell.*, vol. 76, pp. 158–169, Nov. 2018.

[39] L. L. Custode and G. Iacca, "Evolutionary learning of interpretable decision trees," 2020, *arXiv:2012.07723*.

[40] A. Ferigo, L. L. Custode, and G. Iacca, "Quality diversity evolutionary learning of decision trees," in *Proc. 38th ACM/SIGAPP Symp. Appl. Comput.*, 2023, pp. 425–432.

[41] A. Ferigo, L. L. Custode, and G. Iacca, "Quality–diversity optimization of decision trees for interpretable reinforcement learning," *Neural Comput. Appl.*, vol. 2023, pp. 1–12, Nov. 2023.

[42] M. Crespi, A. Ferigo, L. L. Custode, and G. Iacca, "A population-based approach for multi-agent interpretable reinforcement learning," *Appl. Soft Comput.*, vol. 147, Nov. 2023, Art. no. 110758.

[43] M. Galende, G. Sainz, and M. J. Fuente, "Accuracy-interpretability balancing in fuzzy models based on multiobjective genetic algorithm," in *Proc. Eur. Control Conf. (ECC)*, 2009, pp. 3915–3920.

[44] M. Virgolin, A. De Lorenzo, F. Randone, E. Medvet, and M. Wahde, "Model learning with personalized interpretability estimation (ML-PIE)," 2021, *arXiv:2104.06060*.

[45] M. Virgolin, E. Medvet, T. Alderliesten, and P. A. Bosman, "Less is more: A call to focus on simpler models in genetic programming for interpretable machine learning," 2022, *arXiv:2204.02046*.

[46] R. J. Urbanowicz, A. Granizo-Mackenzie, and J. H. Moore, "An analysis pipeline with statistical and visualization-guided knowledge discovery for Michigan-style learning classifier systems," *IEEE Comput. Intell. Mag.*, vol. 7, no. 4, pp. 35–45, Nov. 2012.

[47] Y. Liu, W. N. Browne, and B. Xue, "Visualizations for rule-based machine learning," *Nat. Comput.*, vol. 1, pp. 1–22, Jan. 2021.

[48] K. Pearson, "LIII. On lines and planes of closest fit to systems of points in space," *London, Edinburgh, Dublin Philosoph. Mag. J. Sci.*, vol. 2, no. 11, pp. 559–572, Nov. 1901.

[49] H. Hotelling, "Analysis of a complex of statistical variables into principal components," *J. Educ. Psychol.*, vol. 24, no. 6, pp. 417–441, 1933.

[50] L. van der Maaten and G. Hinton, "Visualizing data using T-SNE," *J. Mach. Learn. Res.*, vol. 9, no. 86, pp. 2579–2605, 2008.

[51] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. KDD*, 1996, pp. 226–231.

[52] A. Lensen, B. Xue, and M. Zhang, "Genetic programming for evolving a front of interpretable models for data visualization," *IEEE Trans. Cybern.*, vol. 51, no. 11, pp. 5468–5482, Nov. 2021.

[53] F. Schofield and A. Lensen, "Using genetic programming to find functional mappings for UMAP embeddings," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jun. 2021, pp. 704–711.

[54] L. McInnes, J. Healy, N. Saul, and L. Großberger, "UMAP: Uniform manifold approximation and projection," *J. Open Source Softw.*, vol. 3, no. 29, p. 861, Sep. 2018.

[55] I. Icke and A. Rosenberg, "Multi-objective genetic programming for visual Analytics," in *Genetic Program.* (Lecture Notes in Computer Science), S. Silva, J. A. Foster, M. Nicolau, P. Machado, and M. Giacobini, Eds. Heidelberg, Germany: Springer, 2011, pp. 322–334.

[56] A. Cano, S. Ventura, and K. J. Cios, "Multi-objective genetic programming for feature extraction and data visualization," *Soft Comput.*, vol. 21, no. 8, pp. 2069–2089, Apr. 2017.

[57] A. Lensen, B. Xue, and M. Zhang, "Genetic programming for manifold learning: Preserving local topology," *IEEE Trans. Evol. Comput.*, vol. 26, no. 4, pp. 661–675, Aug. 2022.

[58] S. Sayed, M. Nassef, A. Badr, and I. Farag, "A nested genetic algorithm for feature selection in high-dimensional cancer microarray datasets," *Expert Syst. Appl.*, vol. 121, pp. 233–243, May 2019.

[59] Z. Sha, T. Hu, and Y. Chen, "Feature selection for polygenic risk scores using genetic algorithm and network science," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jun. 2021, pp. 802–808.

[60] Y. Xue, Y. Tang, X. Xu, J. Liang, and F. Neri, "Multi-objective feature selection with missing data in classification," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 6, no. 2, pp. 355–364, Apr. 2022.

[61] T. Hu, "Can genetic programming perform explainable machine learning for bioinformatics?" in *Genetic Programming Theory and Practice XVII*. Cham, Switzerland: Springer, 2020.

[62] T. Hu et al., "An evolutionary learning and network approach to identifying key metabolites for osteoarthritis," *PLoS Comput. Biol.*, vol. 14, no. 3, 2018, Art. no. e1005986.

[63] C. Sha, M. Cuperlovic-Culf, and T. Hu, "SMILE: Systems metabolomics using interpretable learning and evolution," *BMC Bioinf.*, vol. 22, no. 1, p. 284, 2021.

[64] B. Xue, M. Zhang, W. N. Browne, and X. Yao, "A survey on evolutionary computation approaches to feature selection," *IEEE Trans. Evol. Comput.*, vol. 20, no. 4, pp. 606–626, Aug. 2016.

[65] Y. Xue, B. Xue, and M. Zhang, "Self-adaptive particle swarm optimization for large-scale feature selection in classification," *ACM Trans. Knowl. Discov. Data*, vol. 13, no. 5, p. 50, 2019.

[66] B. H. Nguyen, B. Xue, and M. Zhang, "A survey on swarm intelligence approaches to feature selection in data mining," *Swarm Evol. Comput.*, vol. 54, May 2020, Art. no. 100663.

[67] E. Hancer, B. Xue, and M. Zhang, "A survey on feature selection approaches for clustering," *Artif. Intell. Rev.*, vol. 53, no. 6, pp. 4519–4545, 2020.

[68] W. La Cava and J. H. Moore, "Learning feature spaces for regression with genetic programming," *Genet. Program. Evolv. Mach.*, vol. 21, no. 3, pp. 433–467, Sep. 2020.

[69] Z. Li, J. He, X. Zhang, H. Fu, and J. Qin, "Toward high accuracy and visualization: An interpretable feature extraction method based on genetic programming and non-overlap degree," in *Proc. IEEE Int. Conf. Bioinf. Biomed. (BIBM)*, 2020, pp. 299–304.

[70] M. Muharram and G. D. Smith, "Evolutionary constructive induction," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 11, pp. 1518–1528, Nov. 2005.

[71] M. Virgolin, T. Alderliesten, and P. A. N. Bosman, "On explaining machine learning models by evolving crucial and compact features," *Swarm Evol. Comput.*, vol. 53, Mar. 2020, Art. no. 100640.

[72] T. Uriot, M. Virgolin, T. Alderliesten, and P. A. N. Bosman, "On genetic programming representations and fitness functions for interpretable dimensionality reduction," in *Proc. Genet. Evol. Comput. Conf.*, Jul. 2022, pp. 458–466.
[73] L. Breiman, "Random forest," *Mach. Learn.*, vol. 45, pp. 5–32, Oct. 2001.
[74] J. H. Friedman, "Multivariate adaptive regression splines," *Ann. Stat.*, vol. 19, no. 1, pp. 1–67, 1991.
[75] A. Fisher, C. Rudin, and F. Dominici, "All models are wrong, but many are useful: Learning a variable's importance by studying an entire class of prediction models simultaneously," *J. Mach. Learn. Res.*, vol. 20, no. 177, pp. 1–81, 2019.
[76] J. Robertson, C. Stinson, and T. Hu, "A bio-inspired framework for machine bias interpretation," in *Proc. AAAI/ACM Conf. AI, Ethics, Soc.*, Jul. 2022, pp. 588–598.
[77] C. Buciluǎ, R. Caruana, and A. Niculescu-Mizil, "Model compression," in *Proc. 12th ACM SIGKDD Int. Conf. Knowl. Disc. Data Min.*, Aug. 2006, pp. 535–541.
[78] G. E. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," 2015, *arXiv:1503.02531*.
[79] H. Lakkaraju, E. Kamar, R. Caruana, and J. Leskovec, "Interpretable & explorable approximations of black box models," 2017, *arXiv:1707.01154*.
[80] B. P. Evans, B. Xue, and M. Zhang, "What's inside the black box? A genetic programming method for interpreting complex machine learning models," in *Proc. Genet. Evol. Comput. Conf. (GECCO)*, 2019, pp. 1012–1020.
[81] J. Bacardit, J. D. Hirst, M. Stout, J. Blazewicz, and N. Krasnogor, "Coordination number prediction using learning classifier systems: Performance and interpretability," in *Proc. Genet. Evol. Comput. Conf.*, New York, NY, USA, 2006, pp. 247–254.
[82] N. Lazzarini, P. Widera, S. Williamson, R. Heer, N. Krasnogor, and J. Bacardit, "Functional networks inference from rule-based machine learning models," *BioData Min.*, vol. 9, no. 1, pp. 1–23, 2016.
[83] G. W. Bassel, E. Glaab, J. Marquez, M. J. Holdsworth, and J. Bacardit, "Functional network construction in Arabidopsis using rule-based machine learning on large-scale data sets," *Plant Cell Online*, vol. 23, no. 9, pp. 3101–3116, Sep. 2011.
[84] S. Baron, N. Lazzarini, and J. Bacardit, "Characterising the influence of rule-based knowledge representations in biological knowledge extraction from transcriptomics data," in *Proc. Eur. Conf. Appl. Evol. Comput.*, 2017, pp. 125–141.
[85] H. D. Mettler, M. Schmidt, W. Senn, M. A. Petrovici, and J. Jordan, "Evolving neuronal plasticity rules using cartesian genetic programming," 2021, *arXiv:2102.04312*.
[86] J. Jordan, M. Schmidt, W. Senn, and M. A. Petrovici, "Evolving to learn: Discovering interpretable plasticity rules for spiking networks," 2020, *arXiv:2005.14149*.
[87] Y. Tang, D. Nguyen, and D. Ha, "Neuroevolution of self-interpretable agents," in *Proc. Genet. Evol. Comput. Conf.*, 2020, pp. 414–424.
[88] B. Wang, W. Pei, B. Xue, and M. Zhang, "A multi-objective genetic algorithm to evolving local interpretable model-agnostic explanations for deep neural networks in image classification," *IEEE Trans. Evol. Comput.*, vol. 28, no. 4, pp. 903–917, Aug. 2024.
[89] W. Samek, G. Montavon, S. Lapuschkin, C. J. Anders, and K.-R. Müller, "Explaining deep neural networks and beyond: A review of methods and applications," *Proc. IEEE*, vol. 109, no. 3, pp. 247–278, Mar. 2021.
[90] T. Adel, Z. Ghahramani, and A. Weller, "Discovering interpretable representations for both deep generative and discriminative models," in *Proc. 35th Int. Conf. Mach. Learn.*, Jul. 2018, pp. 50–59.
[91] N. Cammarata et al., "Thread: Circuits," Distill, Las Vegas, NV, USA, 2020. [Online]. Available: https://distill.pub/2020/circuits/
[92] N. Elhage et al., "A mathematical framework for transformer circuits," Transformer Circuits Thread, 2021. [Online]. Available: https://transformer-circuits.pub/2021/framework/index.html
[93] N. Nanda, L. Chan, T. Lieberum, J. Smith, and J. Steinhardt, "Progress measures for grokking via mechanistic interpretability," Oct. 2023, *arXiv:2301.05217*.
[94] M. Saletta and C. Ferretti, "A grammar-based evolutionary approach for assessing deep neural source code classifiers," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jul. 2022, pp. 1–8.
[95] H. Bradley, H. Fan, T. Galanos, R. Zhou, D. Scott, and J. Lehman, "The OpenELM library: Leveraging progress in language models for novel evolutionary algorithms," in *Genetic Programming Theory and Practice XX*. Singapore: Springer, 2024, pp. 177–201.
[96] S. J. Wetzel, "Closed-form interpretation of neural network classifiers with symbolic regression gradients," Jan. 2024, *arXiv:2401.04978*.
[97] M. T. Ribeiro, S. Singh, and C. Guestrin, "'Why should i trust you?' Explaining the predictions of any classifier," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Disc. Data Min.*, 2016, pp. 1135–1144.
[98] L. A. Ferreira, F. G. Guimarães, and R. Silva, "Applying genetic programming to improve interpretability in machine learning models," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, 2020, pp. 1–8.
[99] R. Guidotti, A. Monreale, S. Ruggieri, D. Pedreschi, F. Turini, and F. Giannotti, "Local rule-based explanations of black box decision systems," 2018, *arXiv:1805.10820*.
[100] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst. (NeurIPS)*, 2017, pp. 4768–4777.
[101] S. Wachter, B. Mittelstadt, and C. Russell, "Counterfactual explanations without opening the black box: Automated decisions and the GDPR," *Harvard J. Law Technol.*, vol. 31, no. 2, pp. 841–887, 2018.
[102] A.-H. Karimi, G. Barthe, B. Schölkopf, and I. Valera, "A survey of algorithmic recourse: Definitions, formulations, solutions, and prospects," 2020, *arXiv:2010.04050*.
[103] R. K. Mothilal, A. Sharma, and C. Tan, "Explaining machine learning classifiers through diverse counterfactual examples," in *Proc. ACM Conf. Fairness, Accountability, Transparency*, Jan. 2020, pp. 607–617.
[104] A. Kulesza and B. Taskar, "Determinantal point processes for machine learning," *Found. Trends Mach. Learn.*, vol. 5, nos. 2–3, pp. 123–286, 2012.
[105] S. Sharma, J. Henderson, and J. Ghosh, "CERTIFAI: A common framework to provide explanations and analyse the fairness and robustness of black-box models," in *Proc. AAAI/ACM Conf. AI, Ethics, Soc.*, New York NY USA, Feb. 2020, pp. 166–172.
[106] M. Schleich, Z. Geng, Y. Zhang, and D. Suciu, "GeCo: Quality counterfactual explanations in real time," *Proc. VLDB Endow.*, vol. 14, no. 9, pp. 1681–1693, May 2021.
[107] S. Dandl, C. Molnar, M. Binder, and B. Bischl, "Multi-objective counterfactual explanations," in *Parallel Problem Solving From Nature – PPSN XVI* (Lecture Notes in Computer Science), T. Bäck et al., Eds. Cham, Switzerland: Springer Int., 2020, pp. 448–469.
[108] R. Li et al., "Mixed integer evolution strategies for parameter optimization," *Evol. Comput.*, vol. 21, no. 1, pp. 29–64, 2013.
[109] C. Molnar, *Interpretable Machine Learning: A Guide for Making Black Box Models Explainable*, Leanpub, Victoria, BC, Canada, 2022. [Online]. Available: https://leanpub.com/interpretable-machine-learning
[110] J. Su, D. V. Vargas, and K. Sakurai, "One pixel attack for fooling deep neural networks," *IEEE Trans. Evol. Comput.*, vol. 23, no. 5, pp. 828–841, Oct. 2019.
[111] H. Qiu, L. L. Custode, and G. Iacca, "Black-box adversarial attacks using evolution strategies," in *Proc. Genet. Evol. Comput. Conf. Compan.*, 2021, pp. 1827–1833.
[112] C. Li, W. Yao, H. Wang, T. Jiang, and X. Zhang, "Bayesian evolutionary optimization for crafting high-quality adversarial examples with limited query budget," *Appl. Soft Comput.*, vol. 142, Jul. 2023, Art. no. 110370.
[113] T. Suzuki, S. Takeshita, and S. Ono, "Adversarial example generation using evolutionary multi-objective optimization," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jun. 2019, pp. 2136–2144.
[114] L. L. Custode and G. Iacca, "One run to attack them all: Finding simultaneously multiple targeted adversarial perturbations," in *Proc. IEEE Symp. Ser. Comput. Intell. (SSCI)*, 2021, pp. 1–8.
[115] M. Alzantot, Y. Sharma, A. Elgohary, B.-J. Ho, M. B. Srivastava, and K.-W. Chang, "Generating natural language adversarial examples," in *Proc. Conf. Empir. Methods Nat. Lang. Process.*, Brussels, Belgium, 2018, pp. 2890–2896.
[116] W. Huang, X. Zhao, G. Jin, and X. Huang, "SAFARI: Versatile and efficient evaluations for robustness of interpretability," 2022, *arXiv:2208.09418*.
[117] S. V. Tamam, R. Lapid, and M. Sipper, "Foiling explanations in deep neural networks," 2022, *arXiv:2211.14860*.
[118] M. Fischetti and M. Fischetti, "Matheuristics," in *Handbook of Heuristics*, vol. 1. Cham, Switzerland: Springer Int., 2018, pp. 121–153. [Online]. Available: https://link.springer.com/referencework/10.1007/978-3-319-07124-4#bibliographic-information
[119] M. Goerigk and M. Hartisch, "A framework for inherently interpretable optimization models," *Eur. J. Oper. Res.*, vol. 310, no. 3, pp. 1312–1324, 2023.
[120] K. Deb, "An efficient constraint handling method for genetic algorithms," *Comput. Methods Appl. Mech. Eng.*, vol. 186, nos. 2–4, pp. 311–338, 2000.

[121] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.

[122] T. Helmuth, L. Spector, and J. Matheson, "Solving uncompromising problems with lexicase selection," *IEEE Trans. Evol. Comput.*, vol. 19, no. 5, pp. 630–643, Oct. 2015.

[123] Q. Xu, Z. Xu, and T. Ma, "A survey of multiobjective evolutionary algorithms based on decomposition: Variants, challenges and future directions," *IEEE Access*, vol. 8, pp. 41588–41614, 2020.

[124] J. G. Falcón-Cardona and C. A. Coello Coello, "Indicator-based multi-objective evolutionary algorithms: A comprehensive survey," *ACM Comput. Surveys*, vol. 53, no. 2, pp. 1–35, 2020.

[125] Y. Hua, Q. Liu, K. Hao, and Y. Jin, "A survey of evolutionary algorithms for multi-objective optimization problems with irregular Pareto fronts," *IEEE/CAA J. Automatica Sinica*, vol. 8, no. 2, pp. 303–318, Feb. 2021.

[126] K. O. Stanley, D. B. D'Ambrosio, and J. Gauci, "A hypercube-based encoding for evolving large-scale neural networks," *Artif. life*, vol. 15, no. 2, pp. 185–212, 2009.

[127] D. B. D'Ambrosio, J. Gauci, and K. O. Stanley, "HyperNEAT: The first five years," in *Growing Adaptive Machines: Combining Development and Learning in Artificial Neural Networks*. Heidelberg, Germany: Springer, 2014, pp. 159–185.

[128] T. Nyathi and N. Pillay, "Comparison of a genetic algorithm to grammatical evolution for automated design of genetic programming classification algorithms," *Expert Syst. Appl.*, vol. 104, pp. 213–234, Aug. 2018.

[129] L. D. Whitley, F. Chicano, and B. W. Goldman, "Gray box optimization for Mk landscapes (NK landscapes and MAX-kSAT)," *Evol. Comput.*, vol. 24, no. 3, pp. 491–519, 2016.

[130] Z. Yang, K. Tang, and X. Yao, "Large scale evolutionary optimization using cooperative coevolution," *Inf. Sci.*, vol. 178, no. 15, pp. 2985–2999, 2008.

[131] P. Larrañaga and J. A. Lozano, *Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation*, vol. 2. New York, NY, USA: Springer, 2001.

[132] J. Ceberio, A. Mendiburu, and J. A. Lozano, *Estimation of Distribution Algorithms for Permutation-Based Problems*, Intelligent Systems Group, Dept. Comput. Sci. Artif. Intell., Univ. Basque Country, Leioa, Spain, May 2011.

[133] M. Hauschild and M. Pelikan, "An introduction and survey of estimation of distribution algorithms," *Swarm Evol. Comput.*, vol. 1, no. 3, pp. 111–128, 2011.

[134] J. A. Lozano, P. Larrañaga, I. Inza, and E. Bengoetxea, *Towards a New Evolutionary Computation: Advances on Estimation of Distribution Algorithms* (Studies in Fuzziness and Soft Computing). Heidelberg, Germany: Springer-Verlag, 2006.

[135] K. M. Malan, "A survey of advances in landscape analysis for optimisation," *Algorithms*, vol. 14, no. 2, p. 40, Jan. 2021.

[136] R. Trajanov, S. Dimeski, M. Popovski, P. Korošec, and T. Eftimov, "Explainable landscape-aware optimization performance prediction," in *Proc. Symp. Ser. Comput. Intell.*, New York, NY, USA, 2021, pp. 1–8.

[137] R. Trajanov, S. Dimeski, M. Popovski, P. Korošec, and T. Eftimov, "Explainable landscape analysis in automated algorithm performance prediction," 2022, *arXiv:2203.11828*.

[138] G. Ochoa, K. M. Malan, and C. Blum, "Search trajectory networks: A tool for analysing and visualising the behaviour of metaheuristics," *Appl. Soft Comput.*, vol. 109, Sep. 2021, Art. no. 107492.

[139] M. Fyvie, J. A. W. McCall, and L. A. Christie, "Towards explainable metaheuristics: PCA for trajectory mining in evolutionary algorithms," in *Artificial Intelligence XXXVIII*, M. Bramer and R. Ellis, Eds. Cham, Switzerland: Springer Int., 2021, pp. 89–102.

[140] G. Cenikj, G. Petelin, C. Doerr, P. Korošec, and T. Eftimov, "DynamoRep: Trajectory-based population dynamics for classification of black-box optimization problems," in *Proc. Genet. Evol. Comput. Conf.*, 2023, pp. 813–821.

[141] M. J. Walter, D. J. Walker, and M. J. Craven, "An explainable visualisation of the evolutionary search process," in *Proc. Genet. Evol. Comput. Conf. Compan.*, 2022, pp. 1794–1802.

[142] A. Wallace, A. E. I. Brownlee, and D. Cairns, "Towards explaining metaheuristic solution quality by data mining surrogate fitness models for importance of variables," in *Artificial Intelligence XXXVIII*, M. Bramer and R. Ellis, Eds. Cham, Switzerland: Springer Int., 2021, pp. 58–72.

[143] M. Singh, A. E. I. Brownlee, and D. Cairns, "Towards explainable metaheuristic: Mining surrogate fitness models for importance of variables," in *Proc. Genet. Evol. Comput. Conf. Compan.*, 2022, pp. 1785–1793.

[144] J. H. Drake, A. Kheiri, E. Özcan, and E. K. Burke, "Recent advances in selection hyper-heuristics," *Eur. J. Oper. Res.*, vol. 285, no. 2, pp. 405–428, 2020.

[145] S. K. Smit and A. Eiben, "Parameter tuning of evolutionary algorithms: Generalist vs. specialist," in *Proc. Eur. Conf. Appl. Evol. Comput.*, 2010, pp. 542–551.

[146] N. Urquhart, "Combining parallel coordinates with multi-objective evolutionary algorithms in a real-world optimisation problem," in *Genet. Evol. Comput. Conf.*, Jul. 2017, pp. 1335–1340.

[147] J.-B. Mouret and J. Clune, "Illuminating search spaces by mapping elites," 2015, *arXiv:1504.04909*.

[148] A. Gaier, A. Asteroth, and J.-B. Mouret, "Data-efficient exploration, optimization, and modeling of diverse designs through surrogate-assisted illumination," in *Proc. Genet. Evol. Comput. Conf.*, 2017, pp. 99–106.

[149] K. Miettinen and M. M. Mäkelä, "Interactive multiobjective optimization system WWW-NIMBUS on the Internet," *Comput. Oper. Res.*, vol. 27, nos. 7–8, pp. 709–723, 2000.

[150] N. Urquhart, M. Guckert, and S. Powers, "Increasing trust in meta-heuristics using MAP-elites," in *Proc. Genet. Evol. Comput. Conf. Compan.*, 2019, pp. 1345–1348.

[151] N. Urquhart, S. Höle, and E. Hart, "Automated, explainable rule extraction from MAP-elites archives," in *Proc. EvoApplications*, 2021, pp. 258–272.

[152] J. Slaney and T. Walsh, "Backbones in optimization and approximation," in *Proc. IJCAI*, 2001, pp. 254–259.

[153] M. Fyvie, J. A. W. McCall, L. A. Christie, A.-C. Zăvoianu, A. E. I. Brownlee, and R. Ainslie, "Explaining a staff rostering problem by mining trajectory variance structures," in *Proc. 43rd SGAI Int. Conf. Artif. Intell. Artif. Intell. XL*, 2023, pp. 275–290.

[154] T. M. Banda, A.-C. Zăvoianu, A. Petrovski, D. Wöckinger, and G. Bramerdorfer, "A multi-objective evolutionary approach to discover explainability tradeoffs when using linear regression to effectively model the dynamic thermal behaviour of electrical machines," *ACM Trans. Evol. Learn. Optim.*, vol. 4, no. 1, p. 3, Feb. 2024.

[155] K. Deb and A. Srinivasan, "Innovization: Discovery of innovative design principles through multiobjective evolutionary optimization," in *Multiobjective Problem Solving from Nature: From Concepts to Applications*, J. Knowles, D. Corne, K. Deb, Eds., Heidelberg, Germany: Springer, 2008, pp. 243–262.

[156] K. Deb, S. Bandaru, D. Greiner, A. Gaspar-Cunha, and C. C. Tutum, "An integrated approach to automated innovization for discovering useful design principles: Case studies from engineering," *Appl. Soft Comput.*, vol. 15, pp. 42–56, 2014.

[157] R. J. Scholman, A. Bouter, L. R. Dickhoff, T. Alderliesten, and P. A. Bosman, "Obtaining smoothly navigable approximation sets in bi-objective multi-modal optimization," 2022, *arXiv:2203.09214*.

[158] A. Inselberg and B. Dimsdale, "Parallel coordinates," in *Human-Machine Interactive Systems* (Languages and Information Systems). Boston, MA, USA: Springer, 2009, pp. 199–233.

[159] N. B. Urquhart, "Evaluating the performance of an evolutionary tool for exploring solution fronts," in *Proc. 21st Int. Conf. Appl. Evol. Comput.*, 2018, pp. 523–537.

[160] A. Pryke, S. Mostaghim, and A. Nazemi, "Heatmap visualization of population based multi objective algorithms," in *Proc. 4th Int. Conf. Evol. Multi-Criterion Optim.*, 2007, pp. 361–375.

[161] R. J. Lygoe, M. Cary, and P. J. Fleming, "A real-world application of a many-objective optimisation complexity reduction process," in *Evolutionary Multi-Criterion Optimization*, R. C. Purshouse, P. J. Fleming, C. M. Fonseca, S. Greco, and J. Shaw, Eds. Heidelberg, Germany: Springer, 2013, pp. 641–655.

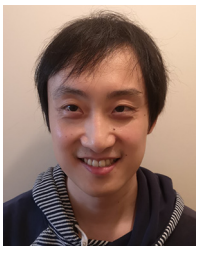
[162] D. J. Walker, R. Everson, and J. E. Fieldsend, "Visualizing mutually nondominating solution sets in many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 17, no. 2, pp. 165–184, Apr. 2013.

[163] I. T. Jolliffe, *Principal Component Analysis*. New York, NY, USA: Springer, 2002.

[164] T. Kohonen, *Self-Organising Maps*. Berlin, Germany: Springer, 1995. [Online]. Available: https://link.springer.com/book/10.1007/978-3-642-97610-0#bibliographic-information

[165] J. W. Sammon, "A nonlinear mapping for data structure analysis," *IEEE Trans. Comput.*, vol. C-18, no. 5, pp. 401–409, May 1969.

[166] R. M. Everson, D. J. Walker, and J. E. Fieldsend, "Life on the edge: Characterising the edges of mutually non-dominating sets," *Evol. Comput.*, vol. 22, no. 3, pp. 479–501, Sep. 2014.
[167] A. V. Kononova, D. W. Corne, P. De Wilde, V. Shneer, and F. Caraffini, "Structural bias in population-based algorithms," *Inf. Sci.*, vol. 298, pp. 468–490, Mar. 2015.
[168] D. Vermetten, F. Ye, and C. Doerr, "Using affine combinations of BBOB problems for performance assessment," in *Proc. Genet. Evol. Comput. Conf.*, 2023, pp. 873–881.
[169] D. Vermetten, F. Ye, T. Bäck, and C. Doerr, "MA-BBOB: Many-affine combinations of BBOB functions for evaluating AutoML approaches in noiseless numerical black-box optimization contexts," in *Proc. Int. Conf. Autom. Mach. Learn.*, 2023, pp. 7–1.
[170] D. Vermetten, B. van Stein, F. Caraffini, L. L. Minku, and A. V. Kononova, "BIAS: A toolbox for benchmarking structural bias in the continuous domain," *IEEE Trans. Evol. Comput.*, vol. 26, no. 6, pp. 1380–1393, Dec. 2022.
[171] D. Vermetten, F. Caraffini, B. van Stein, and A. V. Kononova, "Using structural bias to analyse the behaviour of modular CMA-ES," in *Proc. Genet. Evol. Comput. Conf. Compan.*, 2022, pp. 1674–1682.
[172] B. Van Stein, D. Vermetten, F. Caraffini, and A. V. Kononova, "Deep bias: Detecting structural bias using explainable AI," in *Proc. Compan. Conf. Genet. Evol. Comput.*, 2023, pp. 455–458.
[173] N. van Stein, D. Vermetten, A. V. Kononova, and T. Bäck, "Explainable benchmarking for iterative optimization heuristics," 2024, *arXiv:2401.17842*.
[174] A. Nikolikj, A. Kostovska, D. Vermetten, C. Doerr, and T. Eftimov, "Quantifying individual and joint module impact in modular optimization frameworks," 2024, *arXiv:2405.11964*.
[175] A. Yaman, A. Hallawa, M. Coler, and G. Iacca, "Presenting the ECO: Evolutionary computation ontology," in *Proc. Eur. Conf. Appl. Evol. Comput.*, 2017, pp. 603–619.
[176] A. Kostovska, D. Vermetten, C. Doerr, S. Džeroski, P. Panov, and T. Eftimov, "Option: Optimization algorithm benchmarking ontology," in *Proc. Genet. Evol. Comput. Conf. Compan.*, 2021, pp. 239–240.
[177] A. Kostovska, D. Vermetten, S. Džeroski, P. Panov, T. Eftimov, and C. Doerr, "Using knowledge graphs for performance prediction of modular optimization algorithms," in *Applications of Evolutionary Computation*, J. Correia, S. Smith, and R. Qaddoura, Eds., Cham, Switzerland: Springer Nat., 2023, pp. 253–268.
[178] J. Bacardit and X. Llorà, "Large-scale data mining using genetics-based machine learning," *Wiley Interdiscipl. Rev. Data Min. Knowl. Disc.*, vol. 3, no. 1, pp. 37–61, 2013.
[179] S. P. Walton, B. J. Evans, A. A. M. Rahat, J. Stovold, and J. Vincalek, "Does mapping elites illuminate search spaces? A large-scale user study of MAP–elites applied to human–AI collaborative design," 2024, *arXiv:2402.07911*.
[180] H. Zhang, Q. Chen, B. Xue, W. Banzhaf, and M. Zhang, "MAP-elites for genetic programming-based ensemble learning: An interactive approach [AI-eXplained]," *IEEE Comput. Intell. Mag.*, vol. 18, no. 4, pp. 62–63, Nov. 2023.
[181] L. Seyyed-Kalantari, H. Zhang, M. B. A. McDermott, I. Y. Chen, and M. Ghassemi, "Underdiagnosis bias of artificial intelligence algorithms applied to chest radiographs in under-served patient populations," *Nat. Med.*, vol. 27, no. 12, pp. 2176–2182, 2021.
[182] A. Arias-Duart, F. Parés, D. Garcia-Gasulla, and V. Gimenez-Abalos, "Focus! Rating XAI methods and finding biases," in *Proc. IEEE Int. Conf. Fuzzy Syst.*, 2022, pp. 1–8.
[183] R. Bommasani et al., "On the opportunities and risks of foundation models," 2021, *arXiv:2108.07258*.



Ryan Zhou received the B.Eng. degree from Cornell University, Ithaca, NY, USA, in 2020, and the M.Sc. degree from Trent University, Peterborough, ON, Canada, in 2020. He is currently pursuing the Ph.D. degree with the Queen's University, Kingston, ON, Canada under Dr. Hu and Dr. Feng.

His research interests include neuroevolution, evolutionary computing, deep learning, and interpretability.

Jaume Bacardit received the B.Eng. and M.Eng. degrees in computer engineering and the Ph.D. degree in computer science from Ramon Llull University, Barcelona, Spain, in 1998, 2000, and 2004, respectively.

He is a Professor of Artificial Intelligence with Newcastle University, Newcastle upon Tyne, U.K. He has 100+ peer-reviewed publications that have attracted 700+ citations and an H-index of 40 in Google Scholar. His research interests include development of machine learning methods for large-scale problems, design of explainable AI methods, and application of these methods to a broad range of problems, mostly in biomedical domains.

Alexander Edward Ian Brownlee received the B.S. and Ph.D. degrees in computing science from Robert Gordon University, Aberdeen, U.K., in 2005 and 2009, respectively.

He is a Senior Lecturer with the Division of Computing Science and Mathematics, University of Stirling, Stirling, U.K., where he leads the Data Science and Intelligent Systems Research Group. He has published over 80 peer-reviewed papers on these topics. He has worked with several leading businesses, including BT, KLM, and IES on industrial applications of optimization and machine learning. His main research topics are in search-based optimization methods and machine learning, with a focus on explainable decision support, having applications in civil engineering, transportation and software engineering.

Dr. Brownlee is an Editorial Board Member of the journal *Complex And Intelligent Systems*. He has also co-chaired several workshops and tutorials at Genetic and Evolutionary Computation Conference, CEC, and Parallel Problem Solving from Nature on explainable AI and genetic improvement of software.

Stefano Cagnoni (Senior Member, IEEE) received the master's degree in electronic engineering and the Ph.D. degree in biomedical engineering from the University of Florence, Florence, Italy, in 1988 and 1994, respectively.

He was a Visiting Scientist with Massachusetts Institute of Technology, Cambridge, MA, USA, from 1993 to 1994, and then a Postdoctoral Fellow with the University of Florence. Since 1997, he has been with the University of Parma, Parma, Italy, where he is currently an Associate Professor of Computer Engineering. His research principally regards evolutionary algorithms applications to image analysis and processing, machine learning, and pattern recognition.

Dr. Cagnoni earned the "Evostar Award" in recognition of his outstanding contribution to Evolutionary Computation in 2009. He is a member of the IEEE CIS Task Force on Evolutionary Computer Vision and Image Processing.

Martin Fyvie received the M.Sc. degree in data science and the Ph.D. degree in computer science (specializing in XAI and evolutionary computing), Robert Gordon University, Aberdeen, U.K., in 2020 and 2024, respectively.

Prior to this he worked as a Data Scientist in industrial optimization, digital-twinning, and simulation for the oil and gas and manufacturing industries for over a decade. He is currently a Research Fellow with the National Subsea Centre, Robert Gordon University, Aberdeen, U.K. His research interests include evolutionary algorithms and their industrial applications, explainable AI, net zero energy transition, and multiobjective optimization.

Giovanni Iacca (Senior Member, IEEE) received the B.Sc. and M.Sc. degrees from the Politecnico di Bari, Bari, Italy, and the Ph.D. degree from the University of Jyväskylä, Jyväskylä, Finland, in 2011.

He is currently an Associate Professor of Information Engineering with the Department of Information Engineering and Computer Science, University of Trento, Trento, Italy, where he founded the Distributed Intelligence and Optimization Lab. Previously, he was a Postdoctoral Researcher with RWTH Aachen, Aachen, Germany, from 2017 to 2018; the University of Lausanne and EPFL, Lausanne, Switzerland, from 2013 to 2016; and INCAS3, Assen, The Netherlands, from 2012 to 2016; and worked for several years in industry in the areas of software engineering and industrial automation. His research focuses on computational intelligence, distributed systems, explainable AI, and machine learning.

Prof. Iacca has received the Best Paper Award at EvoApps 2017 and UKCI 2012. He is Co-PI of the PATHFINDER-CHALLENGE project "SUSTAIN" from 2022 to 2026. Previously, he was Co-PI of the FET-Open project "PHOENIX" from 2015 to 2019. He is an Associate Editor of IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION. He is actively involved in the organization of tracks and workshops at some of the top conferences on computational intelligence and regularly serves as a reviewer for several journals and conference committees.

John McCall (Member, IEEE) received the B.Sc. (Hons.), M.Sc., and SERC-funded Ph.D. degrees in mathematics from the University of Aberdeen, Aberdeen, U.K., from 1982 to 1990.

He is a Professorial Lead of Predictive Data Analytics with the National Subsea Centre, Robert Gordon University, Aberdeen, U.K. He has researched in machine learning, search, and optimization for over 30 years, making novel contributions to a range of nature-inspired optimization algorithms and predictive machine learning methods, including EDA, PSO, ACO, and GA. He specializes in industrially applied optimization and decision support (Industry 4.0), working with major international companies in energy and telecommunications as well as a diverse range of SMEs. This work has attracted £Ms in direct industrial funding and grants from U.K. and EU research councils and technology centers. He is the Founding Director of Celerum, which specializes in freight logistics and of PlanSea Solutions, which focuses on marine logistics planning. He has 170+ peer-reviewed publications in books, international journals, and conferences. These have received 3300+ citations with an H-index of 26.

Dr. McCall has served on the Editorial Board for *IEEE Computational Intelligence Magazine*, IEEE SYSTEMS, MAN AND CYBERNETICS JOURNAL, and *Journal of Complex And Intelligent Systems*. He frequently organizes workshops and special sessions at leading international conferences, including most recently Evolutionary Computation and Explainable AI at ACM GECCO from 2022 to 2024. He served as a member for the IEEE Evolutionary Computing Technical Committee and is a member of ACM.

Niki van Stein (Member, IEEE) received the Ph.D. degree in computer science from Leiden University, Leiden, The Netherlands, in 2018, under the supervision of Dr. Bäck with a thesis on data-driven modeling and optimization of industrial processes.

She is an Assistant Professor with Leiden Institute of Advanced Computer Science, Leiden University, specializing in explainable artificial intelligence (XAI). Since January 2022, she has led the XAI Research Group and is a Member of the Management Team of the Natural Computing Cluster. With over 75 peer-reviewed publications and multiple awards, including best paper recognitions at GECCO and the IEEE Symposium Series on Computational Intelligence, she has made significant contributions to the fields of EC and XAI. In addition to her academic role, she has a rich entrepreneurial background as a co-founder of multiple companies, including Emerald-IT, Leiderdorp, The Netherlands, where she served as CTO till 2023. She is a supporter of open science and maintains several open-source packages that facilitate reproducibility in research. Her research focuses on intersection of machine learning, optimization, and XAI, with applications in predictive maintenance, time-series analysis, and engineering design.

David Walker received the B.Sc. and Ph.D. degrees in computer science from the University of Exeter, Exeter, U.K., in 2007 and 2013, respectively.

From 2013 to 2018, he was a Postdoctoral Research Fellow with the University of Exeter, working on a range of topics relating to evolutionary computation and machine learning. He is currently a Senior Lecturer of Computer Science with the University of Exeter, and was a Lecturer of Computer Science with the University of Plymouth, Plymouth, U.K. His Ph.D. thesis titled "Visualization and Ordering of Many-Objective Populations." His research interests include evolutionary computation, multiobjective optimization, explainable artificial intelligence, and visualization.

Dr. Walker is a member of the IEEE Task Force on many objective optimization.

Ting Hu received the B.Sc. and M.Sc. degrees from Wuhan University, Wuhan, China, in 2003 and 2005, respectively, and the Ph.D. degree from Memorial University, St. John's, NL, Canada, in 2010.

She was a Postdoctoral Fellow with Dartmouth College, Hanover, NH, USA. She is currently an Associate Professor of Computing with Queen's University, Kingston, ON, Canada. Her research interests include evolutionary computing, explainable AI, and machine learning applications.