2022-07-09

The intersection of Evolutionary Computation and Explainable AI

Bacardit, J

10.1145/3520304.3533974
GECCO '22: Proceedings of the Genetic and Evolutionary Computation Conference Companion
ACM

The intersection of Evolutionary Computation and Explainable AI

Jaume Bacardit*
Newcastle University
Newcastle upon Tyne, UK
jaume.bacardit@newcastle.ac.uk

Alexander E.I. Brownlee*
University of Stirling
Stirling, UK
alexander.brownlee@stir.ac.uk

Stefano Cagnoni*
University of Parma
Parma, Italy
stefano.cagnoni@unipr.it

Giovanni Iacca*
University of Trento
Trento, Italy
giovanni.iacca@unitn.it

John McCall*
Robert Gordon University
Aberdeen, UK
j.mccall@rgu.ac.uk

David Walker*
University of Plymouth
Plymouth, UK
david.walker@plymouth.ac.uk

ABSTRACT

In the past decade, Explainable Artificial Intelligence (XAI) has attracted a great interest in the research community, motivated by the need for explanations in critical AI applications. Some recent advances in XAI are based on Evolutionary Computation (EC) techniques, such as Genetic Programming. We call this trend *EC for XAI*. We argue that the full potential of EC methods has not been fully exploited yet in XAI, and call the community for future efforts in this field. Likewise, we find that there is a growing concern in EC regarding the explanation of population-based methods, i.e., their search process and outcomes. While some attempts have been done in this direction (although, in most cases, those are not explicitly put in the context of XAI), we believe that there are still several research opportunities and open research questions that, in principle, may promote a safer and broader adoption of EC in real-world applications. We call this trend *XAI within EC*. In this position paper, we briefly overview the main results in the two above trends, and suggest that the EC community may play a major role in the achievement of XAI.

CCS CONCEPTS

• **Computing methodologies** → **Machine learning**; • **Theory of computation** → **Optimization with randomized search heuristics**; • **Human-centered computing** → *Human computer interaction (HCI)*.

KEYWORDS

Explainable Artificial Intelligence, Evolutionary Computation, Optimization, Machine Learning

ACM Reference Format:
Jaume Bacardit, Alexander E.I. Brownlee, Stefano Cagnoni, Giovanni Iacca, John McCall, and David Walker. 2022. The intersection of Evolutionary Computation and Explainable AI. In *Genetic and Evolutionary Computation Conference Companion (GECCO '22 Companion), July 9–13, 2022, Boston, MA, USA.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3520304.3533974

*All authors contributed equally to this work and are listed in alphabetical order.

GECCO '22, July 9–13, 2022, Boston, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/3520304.3533974

1 INTRODUCTION

The field of explainable AI (XAI) focuses on the development of algorithmic methods for the systematic extraction of knowledge and/or decision making processes out of machine learning (ML) models. Successful research in this area has historically focused on methods that mimic human reasoning, and has been motivated by the need to assess the transparency and trustworthiness of ML models. Evolutionary Computation (EC) draws from concepts found in nature to drive development in evolution-based systems, such as genetic algorithms and evolution systems. Alongside other nature-inspired metaheuristics, such as swarm intelligence (SI), the path to a solution is driven by stochastic processes that accumulate knowledge of the problem as they solve it. This also creates barriers to *explainability*: algorithms may return different solutions when re-run from the same input, and technical descriptions of these processes are often a barrier to end-user understanding and acceptance.

Recent growth in the adoption of black-box algorithms, including EC-based methods, in domains such as medical diagnosis [48], manufacturing [47], transport and logistics [13] has led to greater attention being given to the generation of explanations and their accessibility to end-users. This increased attention has helped create a fertile environment for the application of XAI techniques in the EC domain for both end-user- and researcher-focused explanation generation. Furthermore, many approaches to XAI in ML are based on search algorithms – e.g. Local Interpretable Model-Agnostic Explanations (LIME) [49] – that have the potential to draw on the expertise of the EC community; and many of the broader questions (such as what kinds of explanation are most appealing or useful to end users [3]) are faced by XAI researchers in general.

Important questions have then arisen from the application of automated decision making techniques (such as EC and ML), including:

- Why has the algorithm obtained solutions in the way that it has?
- Is the system biased?
- Has the problem been formulated correctly?
- Is the solution trustworthy and fair?

The goal of XAI and related research is to develop methods to interrogate AI processes, with the aim of answering these questions. This can support decision makers while also building trust in AI decision-support through more readily understandable explanations. The position taken in this paper is that, despite the differences in the problem formulation (ML vs optimisation), using or adapting XAI techniques to explain the processes used within EC to tackle search problems will improve the accessibility of such methods to a wider audience, increasing their uptake and impact. As well as this, we posit that EC can play an important role in improving the state-of-the-art XAI techniques that are used within the wider AI community.

The remainder of this paper provides discussion around these themes: first, we provide the motivation for strengthening the link between XAI and EC. Then, in Section 3 we illustrate some concepts related to how XAI can be relevant *within* EC. After, in Section 4 we discuss how EC can be used *for* XAI. Finally, we provide the conclusions in Section 5.

2 MOTIVATION

Explainability is important for several reasons. Perhaps the most crucial is **trust**. The research community is already largely convinced of the value of EC approaches, and keen to increase the uptake of EC tools and methods by non-EC experts. Central to this is convincing users that they can trust the solutions that are generated, by knowing *what* makes that solution better than anything (or at least something) else, which might be seen as synonymous with knowing *why* the solution was chosen. It is also important to consider that such an explanation will likely be important in the future to provide an audit trail for the decisions underpinning an implemented solution, as legislation regulating the use of AI increases.

Extending this theme is that of **validity**. EC methods (and optimisers in general) only optimise the function they have been given. Explaining why a solution was chosen might help us know if it solves the actual problem, or if it just exploits an error or loophole in the problem's definition. This can lead to surprising or even amusing results [36], but can also simply yield frustratingly incorrect solutions to a problem.

EC is stochastic and, as a result, some noise in the generated solutions is likely if not unavoidable. Different runs can produce similar solutions of equal quality but solutions can also feature artefacts that have no impact on their quality. Thus, another motivation is whether we can explain which characteristics of the solution are crucial: its **malleability**. Which variables could be refined or amended for aesthetic or implementation purposes?

Finally, when we define a problem, it is often hard to fully codify all the real-world goals of the system. Subtle rules (for example, "I prefer not to work late on Fridays"; or "Joe likes to drive that route because it ends near his house") are typically used to judge solutions *after* the optimisation is completed. We can generate lots of diverse solutions in order to "optimise" these goals post-hoc but we propose that, better still, an explanation could again reveal which characteristics are important for optimality, allowing one to refine the solutions and better *fit* the real-world problem. This also relates to one of the motivating factors behind interactive EC – we want something that is mathematically optimised, but also something that corresponds to the problem owner's hard-to-codify intuition. By incorporating XAI into interactive EC we could make it easier for the problem owner to interact with the optimiser, see [65].

2.1 Primer on XAI concepts

The next two sections will briefly introduce our view on the relationship between EC and XAI as can already be detected in the current literature or foreseen in the near future based on the intersection of the goals of XAI and the correspondingly most relevant features of EC methods. What we mainly would like to point out is that such a relationship is fluidly bi-directional, which means that we recognise the opportunity both for XAI concepts to contribute to a better understanding of EC methods and for EC to provide valuable tools to solve the problems raised by XAI.

In view of this, we point the reader to an early review where the concepts and terms relevant to XAI are discussed and defined [1, 3, 28]. Without discussing their subtleties any further, we just mention terms like *understandability*, *comprehensibility*, *interpretability*, *transparency*, along with, of course, *explainability*, to give even just an intuitive flavour of the main context within which the interrelationships between EC and XAI occur.

A further reason why we prefer not to delve too deeply and precisely into the definition of such terms is provided by the same review, which highlights that the term "explainability" may assume very different meanings, depending on the target of the explanation. In fact, the latter may be: an end user of an AI system, or an expert in the application domain; someone who is affected by the AI-based decision, or a regulatory entity whose duty is to protect her/him; someone who has the power to make a decision induced by an AI system that will affect many people, etc.; up to a data scientist or software developer, such as an EC researcher, who is expected to embed such concepts into computer applications or use them to gain new insights on the technologies she/he studies and uses. The latter is the viewpoint adopted in the next sections.

In introducing the concept of explainability for EC methods, one should not forget that most methods studied by EC and SI are based on some nature-inspired metaphors that are, themselves, descriptions/explanations of the inner mechanisms that drive such methods. Thus, one may be tempted to use the metaphor to interpret and forecast the method's behaviour. This approach is not always effective, especially when the nature-inspired algorithm reflects the metaphor only partially or, even worse, as pointed out by [55], the metaphor is just a way to disguise an existing algorithm as something different.

3 XAI WITHIN EC

There are a number of ways in which explainability might be useful within EC. Having generated a set of candidate solutions to a problem, a useful explanation of those solutions would identify their important characteristics. For example, in the case of mixed-integer problems, which combinations of variables strongly influence solution quality, and which can be ignored? For a permutation problem, in which order should particular pairs of elements be placed? How important is it that they are placed in that specific order?

Of particular importance to explaining the evolutionary process is identifying when particular solution characteristics were chosen by the algorithm, and which characteristics they "defeated" in the search process. Highlighting characteristics that have survived in the population for a long time indicates the presence of a strong evolutionary trait.

As well as explaining the way in which a population of solutions has been evolved, trust can be engendered by enabling problem owners to interrogate the solutions they are provided with. By undertaking "what-if"-style analysis by the user, they can gain an understanding of the solutions the algorithm has generated by exploring the alternatives. This might be done by keeping track of the solutions that a candidate solution has replaced, as well as by enabling the user to interact with a solution to examine how manipulating variables change the solution for the better or worse.

Research in XAI and EC find some common ground when the internal dynamics of EC algorithms are explored, even if the related work is motivated by different goals in the two contexts.

The following are examples of research topics frequently dealt with in EC that share some goals and insights with XAI, i.e., have already searched similar explanations and may probably benefit from more explicit and aware context sharing in the future.

Landscape analysis [41] is, arguably, one of the main points of contact between XAI and EC. Landscape analysis, in fact, encompasses a set of tools that aim to understand and explaining algorithm behaviour, based on features of the problem, as well as predict algorithm performance and perform automatic algorithm configuration and selection. In this area, some works that explicitly aim at explainable landscape-aware prediction [58, 59] have been proposed recently.

Studies about **hyper-heuristics** [22] and **parameter selection** [54], instead, have highlighted that specific parameter settings allow EC methods to exhibit a "generalistic" behaviour, i.e., to perform generally well even on very different types of functions. The search for such settings has been shown to be effective, for instance, in selecting solutions from the Pareto fronts of multi-objective optimisation problems [60]. Stemming from these considerations, it might be worth exploring whether the search for simple configurations motivated by an easier explainability of the corresponding system may also lead to generalistic solutions, as ML theory (and Occam's razor) seem to suggest.

3.1 Existing work

While the current term "explainability" appears little in the EC literature, a number of existing works might be said to fall into this territory. Deb et al. [18, 19] proposed "innovization" to identify common principles among Pareto-optimal solutions for multi-objective optimisation problems, and gain greater insight into the design process. The idea is that such common principles represent properties that ensure Pareto-optimality and are, by extension, valuable to the problem as a whole. In a similar way, the older concept of *backbones* [56] represent components of a solution that are critical to its optimality. In a satisfiability decision problem, the backbone of a formula is the set of literals which are true in every mode. Identification of such characteristics in a solution could form part of an explanation of its quality.

Another relatively recent line of work [51] concerns how to make sure that the solutions in a multi-objective front are actually not too dissimilar from each other (they belong to the same mode), so that the expert can see a smooth transition in solution space when traversing the Pareto approximation set.

Quality-diversity or illumination algorithms, such as MAP-Elites [24, 46], can be used as an alternative to generate a diverse set of high-quality solutions that can be explored to better accommodate user preferences (including post-hoc considerations of quality like those suggested above). An approach described by [63, 64] that explicitly links this to trust used an interactive decision-making tool to allow users to choose the solutions.

3.2 Evaluating explanations within EC

A strong focus within XAI is how to evaluate explanations. EC has a considerable amount to learn from existing work in this area, as many of the principles used to evaluate explanations of ML models will be transferable.

Given the overlap between visualisation work and XAI, there is value in considering the approaches used to evaluate the quality of a visualisation. There, a common approach is to undertake a *usability study*. A range of methods for doing so exist, and typically involve presenting a domain specialist with a visualisation and asking them to use it to glean information and understanding about the topic it represents. Quantitative information can be obtained using questionnaires that ask the user about their experience using the visualisation, and the visualisation itself can be instrumented so that the user's use of it can be analysed. Deeper insight can be obtained by asking for qualitative feedback. While there have been some usability studies relating to visualisation within EC (e.g. [43, 67]), further efforts towards explanations through visualisation are possible. For instance, a usability study might be constructed that evaluates the accessibility of a proposed explanation, and gauges the extent to which it explains the aspect of EC that it seeks to.

Another approach to analysing visualisation quality is to compare it to a taxonomy of use cases: a useful visualisation will enable a user to complete a number of the tasks the taxonomy describes. ML research has proposed a number of benchmarks for explainability tools, and these could easily be adapted for use within EC. Approaches have considered evaluating the extent to which methods generate explanations, are interpreted as explanations by the user, and are of use [68]. Rosenfeld [50] proposes an evaluation based on measuring the change in use between black-box models and XAI, the explanations' simplicity, the amount of input needed by the user, and the stability of the explanation. Nothing about these evaluation techniques makes them specific to ML, and all of them can be usefully related to EC.

In the explainable ML community a few objective, quantitative measures have been proposed to evaluate explanation accuracy based on a ground truth, such as in [20]. For example, the "comprehensiveness" and "sufficiency" metrics quantify the correctness of explanations by removing items of training data one at a time aligned to the features highlighted by the explanation, and confirming the change in the model prediction. Similar approaches could be taken to measure impact on quality of solutions found by EC methods. Another important aspect could be the stability

in explanations, so that the same explanation system produces the same result reliably, or, alternatively, so that different explanation systems are in agreement (although, in some cases, different perspectives/ways of explaining something may be helpful, e.g., for different audiences).

4 EC FOR XAI

After exploring "what XAI can do for EC", let us take an opposite view and explore "what EC can do for XAI". As known, EC consists of a set of optimisation techniques that may be applied extensively, and often very generically, to a large number of problems. It has therefore reached the stage of a mature discipline that provides ready-to-use solutions to problems which require, from the viewpoint of its outcome, the optimisation of a certain target function or, more generally and importantly, an effective, goal-driven global exploration of a solution space. From this very pragmatic viewpoint, evolutionary algorithms, and metaheuristics in general, can be seen as powerful out-of-the-box tools that can be applied to a huge variety of problems, which may of course include XAI.

As stated in Section 1, taking LIME as an example, the earliest and possibly the most frequently described approaches to XAI try to model the complex, black-box models (generated, for instance, by deep learning networks) by decomposing the global model into a set of simpler local models, explainable or easy to describe. In [26, 27], for instance, the exploration properties of genetic algorithms are used to generate synthetic neighborhoods for learning local interpretable predictors.

In this section, we would like to go beyond this straightforward, "impersonal" use of EC, to try and highlight EC applications to XAI that leverage EC methods' intrinsic properties, i.e., highlight methods in which the evolutionary process is not just a replaceable alternative to other possible approaches to solving a more abstract problem (optimisation, search, etc.) but define a new class of methods for which artificial evolution is the main driving force.

As anticipated earlier, EC has been applied for many years now to the generation of *white-box* (also called *glass-box*) ML models, such as decision trees [12, 21, 32, 33, 40, 52] or sets of classification rules [2, 4, 5, 38, 42] and, more recently, for approximating black-box models with a globally equivalent white-box model, i.e., a decision tree induced by Genetic Programming (GP) [23], Unlike traditional ML approaches for the generation of models using such knowledge representations, which mostly use greedy approaches for model generation, EC methods leverage the global optimisation capabilities of evolutionary search. The previous methods use a batch learning strategy for model building. Alternatively, Learning Classifier Systems (LCS) [10, 11, 30, 62, 69] generate sets of classification rules using an online learning approach using either reinforcement learning (RL) [69] or supervised learning [9].

Similar attempts at combining RL and EC have tried to obtain interpretable policies for RL tasks by combining decision trees induced by GP or Grammatical Evolution with RL acting on the leaves while the policy interacts with the environment [14–16, 29].

Some other works in this area have explicitly focused on addressing the interpretability question in white-box models. The balance between accuracy and interpretability has been explored in the context of genetic fuzzy systems [25]. In this regard, some recent studies have proposed machine-learned quantifiable measures of interpretability [65], while others [66] have emphasised the importance to focus on low-complexity models, especially in the context of GP. Another important aspect in ML, that is fairness, has been instead addressed in [34], where explicit fairness constraints have been introduced in GP to obtain fair classifiers.

Visualisation techniques in the shape of heatmaps have been used to represent the sets of classification rules generated by LCS [61]. This technique was particularly effective when combined with hierarchical clustering to reorder rows (instances) and columns (features) of a dataset, as this enabled an effective global view of how the problem domain was partitioned across the classification rules, and what features were relevant for each partition. Alternatively, 3D visualisation approaches have also showed to be a very effective tool to represent complete rule sets generated by LCSs [39], by using different axes to represent attributes, levels of generality of the rules in which these attributes were involved, and estimated attribute importance.

Moreover, Genetic Programming has been effectively used for the machine learning task of manifold learning [37], i.e., the creation of (ideally) reduced data representations for high-dimensionality datasets to facilitate the work of downstream machine learning algorithms. Often, this task is solved by black-box algorithms that perform a mapping from an original space to a reduced one, without a clear explanation on how this mapping is designed. On the other hand, genetic programming trees offer an interpretable alternative for this task with white-box transformation operations.

More recently, EC has been adopted as a tool to generate *counterfactual* explanations for ML models, i.e., synthetic input samples that are as close as possible to a given input sample, but for which the model gives a different outcome [45]. This is for instance the case of the CERTIFAI tool [53]. A multi-objective approach was instead proposed in [17], where the objectives are the closeness to the input sample, and a measure of changes needed for the counterfactual explanation. These EC-methods for counterfactual explanations are especially useful when the ML model is non-differentiable or, if it is, when access to the gradient is not provided.

Finally, domain-specific studies have also been performed. For instance, the classification rules evolved by EC methods have been analysed in the domain of protein structure prediction [6]. Furthermore, biological functional networks (i.e., graphs) can be inferred by mining the structure of ensembles of rule sets evolved by EC methods [35]. A topological analysis of such networks led to the experimentally-verified discovery of the function of several genes (in the biological sense of the word) for the *Arabidopsis Thaliana* plant organism [8]. Knowledge representations for rules can be constrained in a variety of ways, which shape the data patterns captured by the sets of classification rules using such representations. This leads, potentially, to the extraction of different knowledge from the same data depending on the chosen representation, as was studied for molecular biology datasets [7]. In the field of neuro-evolution, EC methods have instead been used to discover interpretable plasticity rules [31, 44, 70] or to produce self-interpretable agents [57], i.e., agents that (through self-attention) access a smaller fraction of the input, for which interpretable policies are possible.

The list of works mentioned above is not meant to be exhaustive and, as the XAI field is rapidly growing, it is likely that more

studies based on EC aimed at achieving XAI will appear in the near future. For instance, we believe that ever more studies will focus on hybrid systems, e.g., combining EC-induced interpretable models and black-box models for feature extraction and low-level data manipulation. Such a combination has the potential to leverage the benefit of both areas of ML, and fully exploit the exploration capabilities that represent a unique feature of EC.

5 CONCLUSION

We have shown that there is a strong mutual connection between XAI and EC. However, we believe that there are still several research opportunities that have not be thoroughly explored yet, which should mainly aim at: 1) devising tools, be them analytical, visual, data-driven, model-based, etc., to explain EC methods, i.e., their internal functioning, their results, and what properties/settings/instances make an algorithm suitable for achieving the result; 2) defining how solutions provided by EC methods should be checked and verified, and evaluating how much problem knowledge is actually needed to understand these solutions; and 3) fully exploiting the main features of EC methods (e.g., their exploration of "illumination" capabilities) to either provide a posteriori explanations (e.g., in the form of local explanations, or approximations of black-box models) or generate white-box models that are explainable by design. Another important challenge relates to the connection between XAI and neuroevolution (and, in general, neural architecture search): for instance, is there any link between optimized architectures and explainability? (e.g., smaller networks may be easier to explain). We consider these opportunities as the basis for a potential bridge between EC and general AI (where machine/deep learning is currently mainstream), and believe that the EC community may play a fundamental role in the promising research area of XAI.

REFERENCES

[1] Amina Adadi and Mohammed Berrada. 2018. Peeking inside the black-box: a survey on explainable artificial intelligence (XAI). *IEEE Access* 6 (2018), 52138–52160.

[2] Ji Hua Ang, Kay Chen Tan, and AA Mamun. 2010. An evolutionary memetic algorithm for rule extraction. *Expert Systems with Applications* 37, 2 (2010), 1302–1315.

[3] Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador García, Sergio Gil-López, Daniel Molina, Richard Benjamins, et al. 2020. Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Information Fusion* 58 (2020), 82–115.

[4] Jaume Bacardit. 2004. *Pittsburgh Genetics-Based Machine Learning in the Data Mining era: Representations, generalization, and run-time.* Ph.D. Dissertation. Ramon Llull University, Barcelona, Spain.

[5] Jaume Bacardit, Edmund K. Burke, and Natalio Krasnogor. 2009. Improving the scalability of rule-based evolutionary learning. *Memetic Computing* 1 (2009), 55–67. Issue 1.

[6] Jaume Bacardit, Jonathan D. Hirst, Michael Stout, Jacek Blazewicz, and Natalio Krasnogor. 2006. Coordination Number Prediction Using Learning Classifier Systems: Performance and interpretability. In *Genetic and Evolutionary Computation Conference*. ACM Press, New York, NY, USA, 247–254.

[7] Simon Baron, Nicola Lazzarini, and Jaume Bacardit. 2017. Characterising the influence of rule-based knowledge representations in biological knowledge extraction from transcriptomics data. In *European Conference on the Applications of Evolutionary Computation*. Springer, Cham, 125–141.

[8] G. W. Bassel, E. Glaab, J. Marquez, M. J. Holdsworth, and J. Bacardit. 2011. Functional Network Construction in Arabidopsis Using Rule-Based Machine Learning on Large-Scale Data Sets. *The Plant Cell Online* 23, 9 (Sept. 2011), 3101–3116.

[9] Ester Bernadó-Mansilla and Josep Maria Garrell-Guiu. 2003. Accuracy-Based Learning Classifier Systems: Models, Analysis and Applications to Classification Tasks. *Evolutionary Computation* 11, 3 (2003), 209–238.

[10] Larry Bull. 2004. Learning classifier systems: A brief introduction. In *Applications of learning classifier systems*. Springer, Cham, 1–12.

[11] Larry Bull and Tim Kovacs. 2005. Foundations of learning classifier systems: An introduction. In *Foundations of Learning Classifier Systems*. Springer, Cham, 1–17.

[12] Erick Cantu-Paz and Chandrika Kamath. 2000. *Using evolutionary algorithms to induce oblique decision trees.* Technical Report. Lawrence Livermore National Lab., CA, US.

[13] Zong-Gan Chen, Zhi-Hui Zhan, Sam Kwong, and Jun Zhang. 2022. Evolutionary Computation for Intelligent Transportation in Smart Cities: A Survey. *IEEE Computational Intelligence Magazine* 17, 2 (2022), 83–102.

[14] Leonardo Lucio Custode and Giovanni Iacca. 2020. Evolutionary learning of interpretable decision trees. arXiv:2012.07723.

[15] Leonardo Lucio Custode and Giovanni Iacca. 2021. A co-evolutionary approach to interpretable reinforcement learning in environments with continuous action spaces. In *Symposium Series on Computational Intelligence (SSCI)*. IEEE, New York, NY, USA, 1–8.

[16] Leonardo Lucio Custode and Giovanni Iacca. 2022. Interpretable pipelines with evolutionarily optimized modules for RL tasks with visual inputs. arXiv:2202.04943.

[17] Susanne Dandl, Christoph Molnar, Martin Binder, and Bernd Bischl. 2020. Multi-objective counterfactual explanations. In *International Conference on Parallel Problem Solving from Nature*. Springer, Cham, 448–469.

[18] Kalyanmoy Deb, Sunith Bandaru, David Greiner, António Gaspar-Cunha, and Cem Celal Tutum. 2014. An integrated approach to automated innovization for discovering useful design principles: Case studies from engineering. *Applied Soft Computing* 15 (2014), 42–56.

[19] K. Deb and A. Srinivasan. 2008. Innovization: Discovery of Innovative Design Principles Through Multiobjective Evolutionary Optimization. In *Multiobjective Problem Solving from Nature: From Concepts to Applications*, Joshua Knowles et al. (Eds.). Springer, Berlin, Heidelberg, 243–262.

[20] Jay DeYoung, Sarthak Jain, Nazneen Fatema Rajani, Eric Lehman, Caiming Xiong, Richard Socher, and Byron C Wallace. 2019. ERASER: A benchmark to evaluate rationalized NLP models. arXiv:1911.03429.

[21] Yashesh Dhebar, Kalyanmoy Deb, Subramanya Nageshrao, Ling Zhu, and Dimitar Filev. 2020. Interpretable-AI Policies using Evolutionary Nonlinear Decision Trees for Discrete Action Systems. arXiv:2009.09521.

[22] John H Drake, Ahmed Kheiri, Ender Özcan, and Edmund K Burke. 2020. Recent advances in selection hyper-heuristics. *European Journal of Operational Research* 285, 2 (2020), 405–428.

[23] Benjamin P Evans, Bing Xue, and Mengjie Zhang. 2019. What's inside the black-box? a genetic programming method for interpreting complex machine learning models. In *Genetic and Evolutionary Computation Conference*. ACM, New York, NY, USA, 1012–1020.

[24] Adam Gaier, Alexander Asteroth, and Jean-Baptiste Mouret. 2017. Data-efficient exploration, optimization, and modeling of diverse designs through surrogate-assisted illumination. In *Genetic and Evolutionary Computation Conference*. ACM, New York, NY, USA, 99–106.

[25] Marta Galende, G Sainz, and Maria J Fuente. 2009. Accuracy-interpretability balancing in fuzzy models based on multiobjective genetic algorithm. In *2009 European Control Conference (ECC)*. IEEE, New York, NY, USA, 3915–3920.

[26] Riccardo Guidotti. 2018. LORE - Local Rule-based Explanations. https://github.com/riccotti/LORE

[27] Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Dino Pedreschi, Franco Turini, and Fosca Giannotti. 2019. Local Rule-Based Explanations of Black Box Decision Systems. arXiv:1805.10820v.

[28] Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Franco Turini, Fosca Giannotti, and Dino Pedreschi. 2018. A survey of methods for explaining black box models. *ACM computing surveys (CSUR)* 51, 5 (2018), 1–42.

[29] Daniel Hein, Steffen Udluft, and Thomas A. Runkler. 2018. Interpretable policies for reinforcement learning by genetic programming. *Engineering Applications of Artificial Intelligence* 76 (2018), 158–169.

[30] John H Holland, Lashon B Booker, Marco Colombetti, Marco Dorigo, David E Goldberg, Stephanie Forrest, Rick L Riolo, Robert E Smith, Pier Luca Lanzi, Wolfgang Stolzmann, et al. 1999. What is a learning classifier system?. In *International Workshop on Learning Classifier Systems*. Springer, Cham, 3–32.

[31] Jakob Jordan, Maximilian Schmidt, Walter Senn, and Mihai A Petrovici. 2020. Evolving to learn: discovering interpretable plasticity rules for spiking networks. arXiv:2005.14149.

[32] Marek Krętowski. 2004. An evolutionary algorithm for oblique decision tree induction. In *International Conference on Artificial Intelligence and Soft Computing*. Springer, Cham, 432–437.

[33] Marek Krętowski and Marek Grześ. 2005. Global induction of oblique decision trees: an evolutionary approach. In *Intelligent Information Processing and Web Mining*. Springer, Cham, 309–318.

[34] William La Cava and Jason H Moore. 2020. Genetic programming approaches to learning fair classifiers. In *Genetic and Evolutionary Computation Conference*. ACM, New York, NY, USA, 967–975.

[35] Nicola Lazzarini, Paweł Widera, Stuart Williamson, Rakesh Heer, Natalio Krasnogor, and Jaume Bacardit. 2016. Functional networks inference from rule-based machine learning models. *BioData mining* 9, 1 (2016), 1–23.

[36] Joel Lehman, Jeff Clune, and Dusan Misevic. 2020. The surprising creativity of digital evolution: A collection of anecdotes from the evolutionary computation and artificial life research communities. , 274–306 pages. arXiv:1803.03453.

[37] Andrew Lensen, Bing Xue, and Mengjie Zhang. 2021. Genetic Programming for Manifold Learning: Preserving Local Topology. *IEEE Transactions on Evolutionary Computation* early access (2021), 15 pages.

[38] J Juan Liu and J Tin-Yau Kwok. 2000. An extended genetic rule induction algorithm. In *Congress on Evolutionary Computation*, Vol. 1. IEEE, New York, NY, USA, 458–463.

[39] Yi Liu, Will N Browne, and Bing Xue. 2021. Visualizations for rule-based machine learning. *Natural Computing* 1 (2021), 1–22.

[40] X. Llorá and S.W. Wilson. 2004. Mixed Decision Trees: Minimizing Knowledge Representation Bias in LCS. In *Genetic and Evolutionary Computation Conference*. Springer-Verlag, Berlin Heidelberg, 797–809.

[41] Katherine Mary Malan. 2021. A Survey of Advances in Landscape Analysis for Optimisation. *Algorithms* 14, 2 (Jan. 2021), 40.

[42] Romaissaa Mazouni and Abdellatif Rahmoun. 2015. AGGE: A Novel Method to Automatically Generate Rule Induction Classifiers Using Grammatical Evolution. In *Intelligent Distributed Computing VIII*. Springer, Cham, 279–288.

[43] Eric Medvet, Marco Virgolin, Mauro Castelli, Peter AN Bosman, Ivo Gonçalves, and Tea Tušar. 2018. Unveiling evolutionary algorithm representation with DU maps. *Genetic Programming and Evolvable Machines* 19, 3 (2018), 351–389.

[44] Henrik D Mettler, Maximilian Schmidt, Walter Senn, Mihai A Petrovici, and Jakob Jordan. 2021. Evolving Neuronal Plasticity Rules using Cartesian Genetic Programming. arXiv:2102.04312.

[45] Ramaravind K Mothilal, Amit Sharma, and Chenhao Tan. 2020. Explaining machine learning classifiers through diverse counterfactual explanations. In *Conference on Fairness, Accountability, and Transparency*. ACM, New York, NY, USA, 607–617.

[46] Jean-Baptiste Mouret and Jeff Clune. 2015. Illuminating search spaces by mapping elites.

[47] Victor Oduguwa, Ashutosh Tiwari, and Rajkumar Roy. 2005. Evolutionary computing in manufacturing industry: an overview of recent applications. *Applied Soft Computing* 5, 3 (2005), 281–299.

[48] Carlos Andrés Pena-Reyes and Moshe Sipper. 2000. Evolutionary computation in medicine: an overview. *Artificial Intelligence in Medicine* 19, 1 (2000), 1–23.

[49] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "Why should I trust you?". Explaining the predictions of any classifier. In *International Conference on Knowledge Discovery and Data Mining*. ACM SIGKDD, New York, NY, USA, 10 pages.

[50] Avi Rosenfeld. 2021. Better Metrics for Evaluating Explainable Artificial Intelligence. In *International Conference on Autonomous Agents and MultiAgent Systems* (Virtual Event, United Kingdom) *(AAMAS '21)*. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 45–50.

[51] Renzo J Scholman, Anton Bouter, Leah RM Dickhoff, Tanja Alderliesten, and Peter AN Bosman. 2022. Obtaining Smoothly Navigable Approximation Sets in Bi-Objective Multi-Modal Optimization. arXiv:2203.09214.

[52] Amin Shali, Mohammad Reza Kangavari, and Bahareh Bina. 2007. Using genetic programming for the induction of oblique decision trees. In *International Conference on Machine Learning and Applications*. IEEE, New York, NY, USA, 38–43.

[53] Shubham Sharma, Jette Henderson, and Joydeep Ghosh. 2020. CERTIFAI: A common framework to provide explanations and analyse the fairness and robustness of black-box models. In *Conference on AI, Ethics, and Society*. AAAI/ACM, New York, NY, USA, 166–172.

[54] Selmar K Smit and AE Eiben. 2010. Parameter tuning of evolutionary algorithms: Generalist vs. specialist. In *European Conference on the Applications of Evolutionary Computation*. Springer, Berlin, Heidelberg, 542–551.

[55] Kenneth Sörensen. 2015. Metaheuristics — the metaphor exposed. *International Transactions in Operational Research* 22, 1 (2015), 3–18.

[56] Walsh T and Slaney J. 2001. Backbones in optimization and approximation. In *IJCAI*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 254––259.

[57] Yujin Tang, Duong Nguyen, and David Ha. 2020. Neuroevolution of self-interpretable agents. In *Genetic and Evolutionary Computation Conference*. ACM, New York, NY, USA, 414–424.

[58] Risto Trajanov, Stefan Dimeski, Martin Popovski, Peter Korošec, and Tome Eftimov. 2021. Explainable Landscape-Aware Optimization Performance Prediction. In *Symposium Series on Computational Intelligence*. IEEE, New York, NY, USA, 01–08.

[59] Risto Trajanov, Stefan Dimeski, Martin Popovski, Peter Korošec, and Tome Eftimov. 2022. Explainable Landscape Analysis in Automated Algorithm Performance Prediction. arXiv:2203.11828.

[60] Roberto Ugolotti, Laura Sani, and Stefano Cagnoni. 2019. What can we learn from multi-objective meta-optimization of evolutionary algorithms in continuous domains? *Mathematics* 7, 3 (2019), 232.

[61] Ryan J Urbanowicz, Ambrose Granizo-Mackenzie, and Jason H Moore. 2012. An analysis pipeline with statistical and visualization-guided knowledge discovery for michigan-style learning classifier systems. *IEEE computational intelligence magazine* 7, 4 (2012), 35–45.

[62] Ryan J Urbanowicz and Jason H Moore. 2015. ExSTraCS 2.0: description and evaluation of a scalable learning classifier system. *Evolutionary intelligence* 8, 2 (2015), 89–116.

[63] Neil Urquhart. 2017. Combining parallel coords with multi-objective evolut algorithms in a real-world optimisation problem. In *Genetic and Evolutionary Computation Conference*. ACM, New York, NY, USA, 1335–1340.

[64] Neil Urquhart, Michael Guckert, and Simon Powers. 2019. Increasing Trust in Meta-heuristics Using MAP-elites. In *Genetic and Evolutionary Computation Conference - Companion* (Prague, Czech Republic) *(GECCO '19)*. ACM, New York, NY, USA, 1345–1348.

[65] Marco Virgolin, Andrea De Lorenzo, Francesca Randone, Eric Medvet, and Mattias Wahde. 2021. Model Learning with Personalized Interpretability Estimation (ML-PIE). arXiv:2104.06060.

[66] Marco Virgolin, Eric Medvet, Tanja Alderliesten, and Peter AN Bosman. 2022. Less is More: A Call to Focus on Simpler Models in Genetic Programming for Interpretable Machine Learning.

[67] David J. Walker. 2018. Visualisation with treemaps and sunbursts in many-objective optimisation. *Genetic Programming and Evolvable Machines* 19, 3 (2018), 421–452.

[68] Rebekah Wegener and Jörg Cassens. 2021. Explainable AI: Intrinsic, Dialogic, and Impact Measures of Success. In *Workshop on Operationalizing Human-centered Perspectives in Explainable AI*. ACM, New York, NY, USA, 7 pages.

[69] Stewart W Wilson. 1995. Classifier fitness based on accuracy. *Evolutionary computation* 3, 2 (1995), 149–175.

[70] Anil Yaman, Giovanni Iacca, Decebal Constantin Mocanu, Matt Coler, George Fletcher, and Mykola Pechenizkiy. 2021. Evolving plasticity for autonomous learning under changing environmental conditions. *Evolutionary computation* 29, 3 (2021), 391–414.