2022-07-19

An Explainable Visualisation of the Evolutionary Search Process

Walter, Mathew

http://hdl.handle.net/10026.1/19155

10.1145/3520304.3533984
GECCO'22: Proceedings of the Genetic and Evolutionary Computation Conference Companion
Association for Computing Machinery

An Explainable Visualisation of the Evolutionary Search Process

Mathew J. Walter
mathew.walter@plymouth.ac.uk
School of Engineering, Computing
and Mathematics,
University of Plymouth
Plymouth, UK

David J. Walker
david.walker@plymouth.ac.uk
School of Engineering, Computing
and Mathematics,
University of Plymouth
Plymouth, UK

Matthew J. Craven
matthew.craven@plymouth.ac.uk
School of Engineering, Computing
and Mathematics,
University of Plymouth
Plymouth, UK

ABSTRACT

The comprehension of the Evolutionary Algorithm (EA) search process is often eluded by challenges of transparency inherent to *black-box* EAs, thus affecting algorithm enhancement and hyper-parameter optimisation. In this work, we develop algorithm insight by introducing the Population Dynamics Plot (PopDP). PopDP is a novel and intuitive visualisation capable of visualising the population of solutions, the parent-offspring lineage, solution perturbation operators, and the search process journey. We apply PopDP to NSGA-II to demonstrate the insight attained and the effectiveness of PopDP for visualising algorithm search on a series of discrete dual- and many-objective knapsack problems of different complexities, and our results demonstrate that the method can be used to produce a visualisation in which the lineage of solutions can be clearly seen. We also consider the efficacy of the proposed explainable visualisation against emerging approaches to benchmarking explainable AI methods and consider the accessibility of the resulting visualisations.

CCS CONCEPTS

• **Computing methodologies** → *Discrete space search*; • **Mathematics of computing** → *Dimensionality reduction*; • **Human-centered computing** → **Visualization techniques**.

KEYWORDS

Visualisation, Evolutionary Computation, Explainability, Multi- and Many-objective Optimisation

ACM Reference Format:
Mathew J. Walter, David J. Walker, and Matthew J. Craven. 2022. An Explainable Visualisation of the Evolutionary Search Process. In *Proceedings of The Genetic and Evolutionary Computation Conference 2022 (GECCO '22).* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Evolutionary algorithms (EAs) use operators inspired by evolution in the natural world to perturb a solution/population of solutions to an optimum/optima, making EAs an effective strategy for solving optimisation problems. State-of-the-art EAs can provide many benefits over conventional optimisation techniques and are required where no gradient information is available and no exact approaches can be devised. However, the utilisation of high-dimensional data and the complex relationships between operators and selection pressure conceal the intuition of EAs and hence yield EAs with *black-box* algorithm states.

Comprehending EAs is fundamental to enhancing EA development and transparency. This work addresses the challenges of comprehending EAs by introducing the Population Dynamics Plot (PopDP) to visualise the entire population of a dual- and a many-objective EA population as they evolve during an EA run. The PopDP visualisation shows the solutions in the objective space or a lower dimension projection of the objective space (efficacious for visualising many objective solutions), the parent-offspring lineage and the solution perturbation operators which have generated the solution. Ultimately, PopDP tells a coherent story of the search process journey in a single plot.

PopDP builds on existing visualisations that have been shown to provide informative visualisations of the evolutionary process in the objective space [9, 29]. We provide additional information visualising the lineage and operator, creating a visualisation that can exhibit the dynamics of the search process.

Noting the limitations of possible excessive information, we provide a visualisation framework for which visualising information superfluous to the Decision Maker (DM) can be expunged subject to the DM's requirements. Thus, this provides an easily adaptable visualisation to fit different problem types, i.e. discrete, continuous, multi or many-objective problems, choice of visualising linkage and operators, and also supports DM preference and accessibility.

This work utilises knapsack problems to demonstrate how PopDP can illustrate algorithm performance, complemented by implicit visual analysis. We visualise two- and four- objective problems with a range of different hyper-parameter choices to demonstrate how EA practitioners can use PopDP as an informative way of visualising the search, and hence the effect hyper-parameters have on the search process. We apply PopDP to simpler problems to first create a basis for understanding the visualisation before applying PopDP to more complex problems with a larger search space.

The novel contributions of this work are outlined as follows:

(1) A novel explainable visualisation, PopDP, is introduced, allowing insight into visualising the population of solutions, the parent-offspring lineage, solution perturbation operators and the journey of the search process in a single plot.
(2) The efficacy of PopDP for visualising the search process for dual and many-objective discrete binary knapsack problems is examined.

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

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

(3) We evaluate PopDP in terms of emerging explainable AI concepts.

The remainder of this paper is structured as follows: Section 2 begins by providing a brief overview of the existing and relevant EA visualisation literature concerned with the contributions of this work. The methodology and experimental setup are contained in Section 3, detailing the problem and PopDP framework. In Section 4, we present the results and provide implicit analysis. We also evaluate the framework regarding emerging explainable AI concepts from the literature. Finally, we summarise and conclude the work in Section 5.

2 EXISTING LITERATURE

In this section, we review the existing literature relating to the contributions of this work.

The work of [13], begins to set out the motives for explainable evolutionary metaheuristics and considers the trajectory of solutions in a search to enhance explainability. However, existing work considering the explainability of evolutionary computing methods is novel and nascent. So, instead, we include a selection of EA visualisation, which we contrast with this work.

For a comprehensive taxonomy of Pareto front EA visualisation, see the work of [12]. However, this work will only examine niche literature that considers either visualisations of EA lineage or visualisations of EA operators.

Some of the earlier work [26] recognises the significance of providing the DM with high-level pictorial representations rather than primal methods of matrices or genotypical string representations of solutions to facilitate the development of EAs.

Relevant literature includes the work of [6], which considers understanding lineage via a proposed novel tree structure to examine exploration and exploitation in EAs. The result allows the DM to determine parent-offspring relationships and operators used for generation to analyse the behaviour of the evolution process. Coincidentally, they also apply this research to a multi-objective 0/1 knapsack problem. In contrast, unlike the work we propose, objective space information is not available, so population structures are not visible and thus only contains a subset of the critical information we present.

Pedigree graphs or family trees predicated upon genealogy can also visualise parent-offspring relationships. But like the research of [6] the objective space is not present in the visualisation; thus, the DM can not easily understand the complete journey of the solutions. The author of [17] endeavours to alleviate this issue by providing a framework to visualise pedigree graphs in conjunction with a fitness value graph. However, we present a visualisation (PopDP) that can show both of these characteristics in a single plot, and PopDP allows the visualisation of each solution without 'cluttering' the figure. Further, illustrations of the operators used to generate the solutions are not visible from this proposed framework, unlike PopDP.

ELICIT [7] and GAVEL [15] are both interactive tools that can also visualise lineage among solutions, and the operators used to propagate solutions. However, these visualisations also do not display the objective space population structures, and thus, subsequently suffer from similar limitations as the work previously reviewed.

An example of work that attempts to visualise operators is that of [30]. This work implements a barcode-like visualisation to examine genetic operators and genetic material. While this visualisation provides an abundance of information, reading and analysing large barcodes is no trivial task, particularly for examining many solutions. Similarly, the EAVis tool [16] was developed to visualise operators facilitating solution perturbation in a table-like data format in conjunction with separate fitness plots. Furthermore, GeneaQuilts [2] provides an interactive tool to visualise EA genealogies compactly.

DU (Diversity Usage) maps [22] provides a single heatmap that displays the diversity of genotypes in an EA population and the degree to which each genotype contributes to phenotype during evolution. This visualisation is applied to EA design.

Most recently, VisEvol [4] provides visual analytics tools for the evolutionary optimisation of machine learning hyperparameters. Again this is an interactive tool that displays many different existing visualisations, which can, with multiple visualisations, show a wealth of information.

Aside from the field of EAs, the existing literature for visualising solution ancestries appears far more prevalent in Genetic Programming (GP) [3, 8, 19–21, 25]. Most of the GP domain work relies heavily on the fundamentals of tree and graph theory to express interpretability.

Much of the work described recognises the importance and challenges of visualising lineage and operators for better algorithm understanding and development. But often these visualisations can be challenging to interpret and require multiple subsequent visualisations. This work recognises and attempts to ameliorate all these issues in a single plot that can visualise lineage, the operators and objective space. To the best of our knowledge, until now, no specific visualisations exist to visualise all three characteristics in a single plot to provide an information-rich and comprehensive understanding of the evolutionary process.

3 METHODOLOGY

This section begins by describing a dimensionality reduction method used for visualising EA populations in the objective space. We then introduce the solution operator colouring metric and lineage, the dual- and many-objective knapsack problems, and the experimental parameters and visualisation generation framework.

3.1 Objective Space Projection

Visualising high dimension population subsets such as the Pareto front in the objective space is already a non-trivial task. When attempting to visualise the whole high dimensional population, the complexity of the problem is compounded. We, therefore, need to reduce the high dimensional population to a more manageable number of spatial dimensions. The research of [28] showed MDS effectively reduced the dimensionality of a population by projecting the solutions into a lower manifold space whilst maintaining a similar structure.

The examination by [9] showed how MDS and other dimension reduction methods could effectively reduce the dimensionality of a population using statistical techniques and inference. They also proposed visualising the generations in the z-axis, which worked well to visualise single-objective functions and small population sizes. However, MDS contains a computationally difficult PCA process (MDS has a complexity of $O(CN^2 + N^3)$, where C is the cost of computing and accessing each entry of the dissimilarity matrix and N is the size of the input array). The population size and the large number of function evaluations required to reduce many objectives would result in significantly long MDS memory requirements and compute time.

Subsequently, the work of [29] proposed a method that had a fast compute time (the computation requires $O(CnN + knN + n^3)$ time, where n is the number of landmarks (arbitrarily chosen subset from the population) and k is the dimensionality of the input matrix) that could handle large populations and many generations that demonstrated properties akin to MDS for EA visualisation. The crux of this method was the implementation of LMDS [10] for projection. This framework is compatible with the DM's preferred dimension reduction technique; we adapt the work of [29] for the objective space projection to expedite the compute time for large populations. While this work showed solutions in the objective space, we note that it did not display solution lineage and the offspring producing operator utilised to generate each solution.

In order to project in the dimension reduced objective space, we first need to save all population coordinate data along with the generation the solution was formed during an EA run. We then create a population coordinate dissimilarity matrix $\mathbf{D}$ along with arbitrarily chosen landmarks n (a random subset of the solution coordinates) and apply LMDS, formally: given a pairwise $n \times n$ distance dissimilarity matrix $\mathbf{D} = (d_{i,j})$, find n vectors $x_1, \ldots, x_n \in \mathbb{R}^K$ such that $||x_i - x_j|| \approx d_{i,j}$ for all $i, j \in \{1, \ldots, n\}$. With the created eigenvector $\mathbf{E}_k$ and eigenvalue $\mathbf{\Lambda}_k$ matrices we can reconstruct the coordinate data in 2 dimensions.

The $k \times n$ matrix $\mathbf{Y}$ can be constructed by

$$\mathbf{Y} = \mathbf{E}_k \mathbf{\Lambda}_k^{\frac{1}{2}}. \tag{1}$$

Finally, we use distance-based triangulation to plot the remaining non-landmark points. We can plot these points in the x- and y- axes. We plot the generation number in the z-axis.

Note, that we only apply dimension reduction in cases where the number of objectives is greater than two.

3.2 Visualisation Generation

To generate a PopDP visualisation, we employ an evolutionary algorithm to generate solutions to a problem. During the optimisation process, a record is kept of how each solution was generated and the solution generation mechanism. We also use this record to determine solution parent-offspring relationships.

Once the 2-D objective values or the dimension reduced 2-D objective values (if the problem is multi- or many- objective) and the operator and solution lineage arrays have been produced in the EA run, we can render a PopDP visualisation. We first plot the 2-D objective values in the x- and y- axis with the generation number in the z-axis. We then connect the corresponding parents with their crossover created offspring via a dotted black line. To enhance the clarity of the visualisation, links between solutions generated exclusively with mutation operations are omitted – work is ongoing to identify a way in which this information can be usefully incorporated without causing cognitive overload for the DM.

Finally, with the information obtained from the solution recording, we can colour the unfeasible solutions red. We also assume that infeasible solutions are of minimal importance to the DM, and therefore we do not include operator generation symbology for infeasible solutions (i.e., infeasible solutions generated by any mechanism will be identified as a red circle). For feasible solutions, we colour the solutions created with mutation green and forge the shape of the solution into a diamond. The solutions generated by crossover are represented as a black cross. A blue circle represents solutions unaffected by an operator.

This framework can be adapted to the DM's preference, e.g., increasing the information shown on the visualisation, such as colouring specific solutions the DM wishes to track or non-dominating solutions gold, reducing the information displayed on the visualisation or changing colours to suit a visually colour-impaired DM.

3.3 Multi-objective Knapsack Problem

The binary multi-objective knapsack problem was first introduced in the work of [31]. The dual-, multi- and many- objective binary knapsack problem can formally be defined as: given a set of m items and a set of n knapsacks, where c_i is the capacity of knapsack i, $g_{i,j}$ is the gain (i.e., profit, customer preference) and $w_{i,j}$ is the weight of item j according to knapsack i, the task is to finds a vector $x = (x_1, \ldots, x_m) \in \{0, 1\}^m$, such that $f(x) = (f_1(x), \ldots, f_n(x))$ is maximised, where $f_i(x) = \sum_{j=1}^{m} g_{i,j} \cdot x_j$ for all $i \in 1, \ldots, n$, subject to $\sum_{j=1}^{m} w_{i,j} \cdot x_j \leq c_i$.

3.4 Experiment Parameters

Unless explicitly stated in the figure caption, the default set-up parameters are as follows: NSGA-II [11] is used for generating the solutions for a problem. Binary crossover with a probability parameter of 0.3 is implemented. The mutation operator is the bit flip with a probability set at 0.3. The population size is set to 10 solutions. The number of generations is set to 10.

It is important to note that we do not intend to produce an optimised algorithm for these problems as this is not the objective of this work, and artefacts and abnormalities of a non-optimal algorithm can be informative to visualise to the reader. However, this by no means detracts from this work.

Each chromosome contains seven alleles - this is the number of decision variables (possible items in the knapsack) for all objectives. The problems contain one knapsack. The knapsack problems contain either two (profits and client preference) or four objectives (profit, client preference, reduction in CO_2 emissions and time between packing score) as stated in the section heading. All four objectives are to be maximised. The single knapsack constraint is a weight capacity of $c_1 = 29$. The maximum objective values for the dual knapsack problems are: 37 profit ($f_1(x) = 37$) and 22 preference ($f_2(x) = 22$). The maximum objective values for the simpler many objective knapsack problems are $f_1(x) = 37$, $f_2(x) = 22$, $f_3(x) = 30$ and $f_4(x) = 19$ and for the complex many objective problem: $f_1(x) = 95$, $f_2(x) = 95$, $f_3(x) = 83$ and $f_4(x) = 55$ with a capacity $c_1 = 99$ and 20 decision variables. We run the complex problem for 100 generations.

4 RESULTS

This section aims to visualise and analyse the evolutionary search process of an EA run with PopDP. We start by visualising dual-objective knapsack problems. We then visualise many-objective knapsack problems. We aim to demonstrate the effectiveness of PopDP to manifest the search with its three key characteristics: visualising the objective space, visualising solution generation mechanisms and visualising lineage. We finally consider evaluating PopDP against nascent explainable AI benchmarks. PopDP is most effective as a 3-D interactive visualisation - to compensate for this, we present the results from three different perspectives, with varying elevation and azimuth.

4.1 Dual-objective Knapsack

In Figure 1, the initially generated solutions can be seen in the first generation. Through the employment of red colouring, we can identify that 2/10 of the initial solutions are infeasible. Numerous initial infeasible solutions could suggest the DM has a good spread of initial solutions in the search space. However, these solutions could use better generation mechanisms to create more solutions in the feasible regions. More niche solution generation mechanisms could expedite the search process at the cost of a lower initial diversity.

Furthermore, utilising the visualisation during an online EA run could provide the necessary insight to allow the DM to drive the search in a more desirable area, such as restricting solutions to particular search regions.

One of the most striking artefacts of the search process is the vertical 'stacks' of solutions. Some of the solutions in these stacks begin and are created at the initial generation. Then the solutions remain optimal in regards to the current population fitness; thus, these solutions are not removed during selection from the population and hence are carried forward into subsequent generations. The mutation and crossover operators act on these solutions from these stacks to form new solutions, as seen by the dotted line connecting the parent to the crossover offspring. This solution linkage allows one to trace each solution back to its initial crossover ancestor directly. With this particular implementation of NSGA-II, if solutions created by crossover are located in the same position as existing solutions in the population, they appear to overwrite one another. Hence we see a black circle, which results from a blue circle symbol and a black cross symbol written over the top of each other. This algorithm artefact highlights a weakness of this algorithm implementation; thus, a DM could stop the acceptance of duplicate solutions to resolve this issue.

For the initial solution of the stack (for solutions not created in the first generation), we see either a green diamond to indicate creation by mutation or a black cross to indicate creation by crossover. The stacks have different lengths allowing the DM to quickly determine the generation or part of the algorithm process where the operator created the solution. For example, in Figure 1, at approximately client preference objective value 15 and profit objective value 35 $(15, 35)$, a stack forms at the initial generation, and the solution is strong enough to remain in the population for four subsequent generations.

We can also identify the optimum feasible solution at (x, y) position $(37, 22)$. To locate optimal solutions, one can look at the non-red coloured solutions from a top-down perspective (best seen in Figure 1c) and choose solutions that yield maximum values in the two objective spaces. Then, using the framework we highlighted in Section 3, we could colour this solution or a whole Pareto front another colour to help distinguish it from the rest of the population.

Some solutions generated appear to be lone mutation solutions suggesting the perturbation took the solution out to non-dominate space, and selection rejected the poor fitness solutions from the population. We can also see the same effect with some unperturbed solutions as better dominating solutions are created, and the deficient dominated solutions are demoted out of the population during the non-dominating sort and crowding distance components of NSGA-II. This effect can be seen in Figure 1b, where a blue stack of solutions disappears at generation 3.

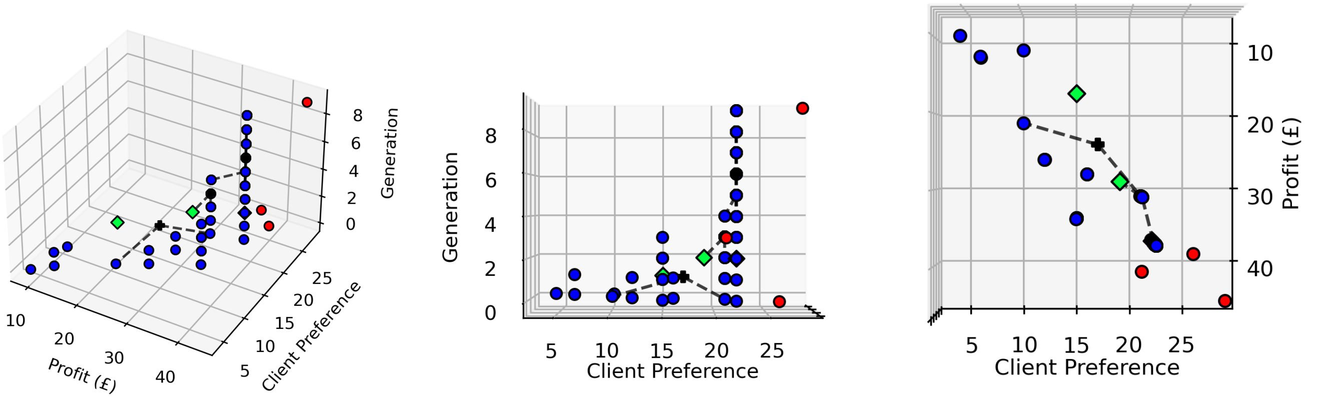
On rare occasions, a single back dotted line leading to a crossover produced solution exists in the visualisation. This is an artefact of the algorithm, showing how the algorithm can choose two identical parents for crossover. These indicators allow one to identify further algorithm weaknesses and limitations from PopDP.

4.2 Hyper-parameter Analysis

We next consider a dichotomy of crossover and mutation hyper-parameters (these identified in figure captions) and contrast the visualisation with visualisations of previous parameters to visually evaluate the perturbation effect from mutation and crossover with PopDP.

Setting the mutation value to one and crossover value to zero, we can see the results of this run in Figure 2a. From the ubiquitous green diamonds, we can determine that mutation is a dominant operator; the algorithm's artefacts or the hyper-parameters may account for the distribution difference. In this case, we can confirm the mutation parameter is set to one and hence is the cause of this effect. We observe populations with larger Euclidean distances between solutions when contrasted to runs that utilised both mutation and crossover operators, suggesting the mutation operator in this implementation yields very diverse populations. In this particular implementation, it appears mutation is the dominant driving force for the diversity of solutions, which can be seen throughout multiple runs. The mutation operator seems to produce a larger perturbation than crossover. As a result, we see that many more infeasible solutions are retained, leading the DM to consider using less mutation and more crossover to enhance the search process. Given mutation is still producing new feasible solutions in the final generation this may indicate to the DM that additional EA evaluations or more suitable hyperparameters are appropriate.

For the following example, we set the mutation parameter to zero and the crossover parameter to one (Figure 2d). In this example, we see crossover yielding many more feasible solutions than previous results with no crossover. From the PopDP visualisation, crossover appears to converge on a few optimum solutions with little diversification quickly. As the solutions generated by crossover have



(a) Azimuth and elevation angles are are 35° and -60° respectively.

(b) Azimuth and elevation angles are are 0° and 0° respectively.

(c) Azimuth and elevation angles are 90° and 0° respectively.

Figure 1: The dual-objective knapsack problem solutions generated with a mutation probability of 0.3 and crossover probability of 0.3. The ◆ solutions were created with mutation. The 🔵 solutions are non perturbed. The + solutions were created with crossover. The 🔴 solutions are infeasible solutions.

smaller Euclidean distances than those generated by a mutation in previous runs, it seems crossover is a driving force for exploitation in this implementation, limiting the search to a specific region early in the process. Furthermore, it is an interesting consequence that even infeasible solutions employed to generate new solutions can produce good feasible solutions, as seen in Figure 2d.

Next, we consider setting both crossover and mutation values to 0.5. The results of this can be seen in Figure 2g. We notice a better balance of exploration and exploitation. We see a few diverse solutions produced by mutation in addition to some streams of solutions assembling as more of a front rather than a single optimum solution. Populations have not yet converged to the optimum objective value, implying that better hyperparameters likely exist.

With this visualisation, we track parental lineage; we can see which solutions were produced with mutation. Furthermore, we can view perturbation sizes and the trajectory of the solutions as they explore and exploit the search space, creating a comprehensive and interpretable picture of the search process. In the next section, we transfer PopDP and the newfound knowledge of dual-dimensional problem analysis to evaluate many-dimensional problems.

4.3 Many-objective Knapsack

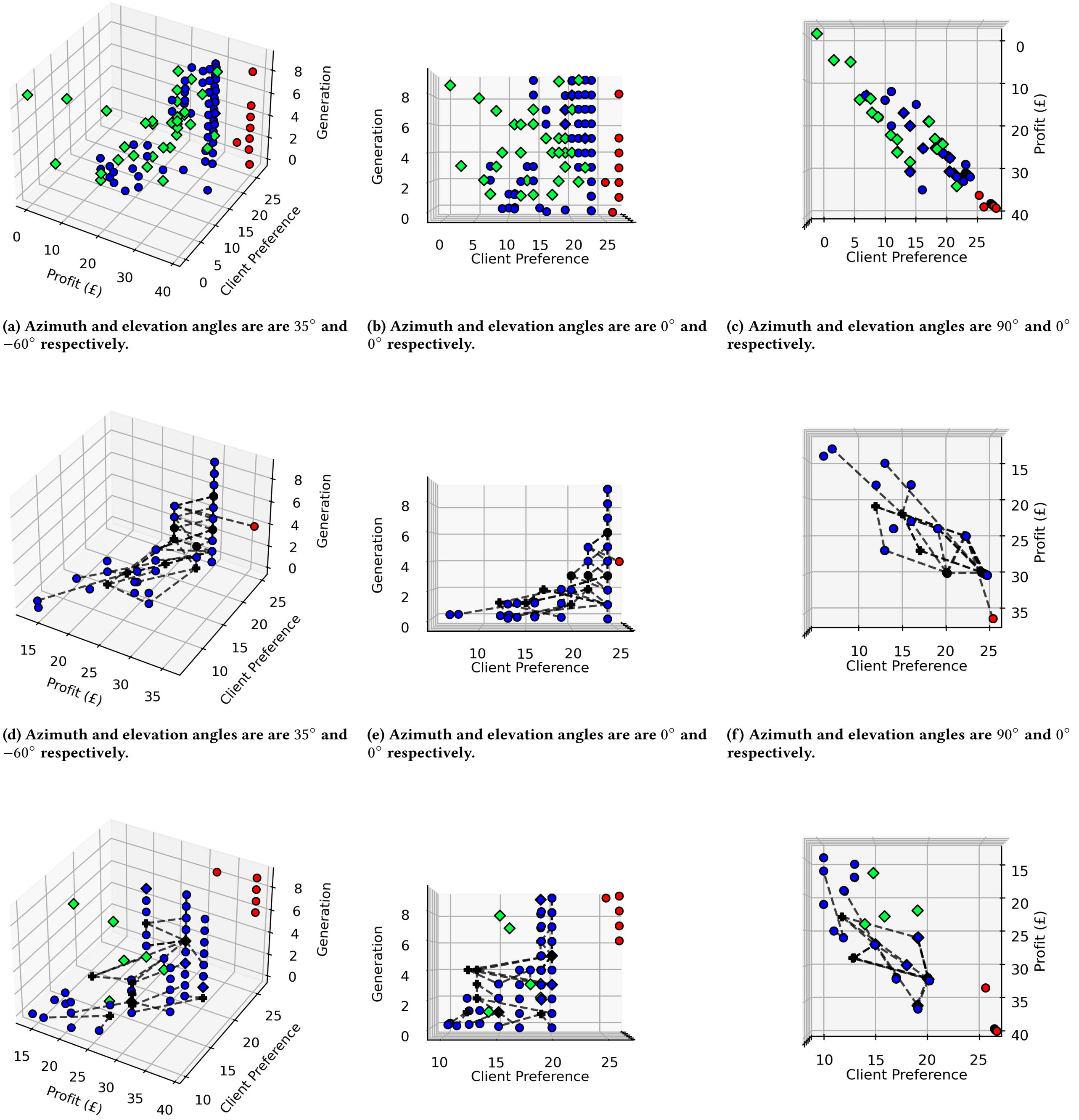
PopDP is not limited by the number of problem objectives, and in this section, we consider its application to a 4-objective instance of the knapsack problem ($m = 4$). In this case, we reduce the 4-objective space to two dimensions using LMDS. During the dimensionality reduction process, it is noteworthy to understand that we create more interpretability (i.e., are able to project solutions into a more manageable objective space) but at the cost of losing some of the original spatial information.

As the initial problem objectives have been reduced to preserve maximum variation, we create two new objectives; these labelled y_1 and y_2 (as seen in Figure 3), by integrating linear combinations of the initial four objectives.

The many-objective problem is more complex to interpret due to the spatial information loss. We can see the solutions converging along multiple non-axis trajectories (the multiple objectives), unlike the dual-objective problems where the solutions converge along the dual objective axes forming a single positively correlated trajectory of solutions. These solution trajectories appear to converge to the origin from different directions (different objectives). There appears to be only a single non-feasible solution. This could be due to the additional objectives exponentially increasing the search space and constant population size, leading to a larger likelihood that a solution is located in the feasible search space than solutions in dual-objective runs with the same parameters. An infeasible solution can be observed near the origin. Unlike the dual-objective problems, noticing a clear, feasible boundary is more challenging as the axes have been transformed as a prerequisite for LMDS. We also notice how new feasible solutions streams are located in the final generations, suggesting the DM may acquire more optimal solutions with a prolonged run time. It is intriguing to observe the later stages of the search where strong solutions in a single objective are crossed to create solutions that are strong in multiple objectives.

Figure 4 shows a more complex knapsack example than seen in Figures 1, 2 and 3. It is more complex by the requirement of taking many more functions to converge to the optimum. The previous examples we have considered are simpler to create a basis for understanding the visualisation. The next example is more complex and perhaps more representative of real-world problems.

The combination of the figures displays the solution evolution through time, and we can perform a similar analysis to the aforementioned many-objective example. As noticed in the simpler many-objective problem, we see 'stacks' of solutions converging along a few different paths. Some of these near-optimum solutions have



(a) Azimuth and elevation angles are are 35° and −60° respectively.

(b) Azimuth and elevation angles are are 0° and 0° respectively.

(c) Azimuth and elevation angles are 90° and 0° respectively.

(d) Azimuth and elevation angles are are 35° and −60° respectively.

(e) Azimuth and elevation angles are are 0° and 0° respectively.

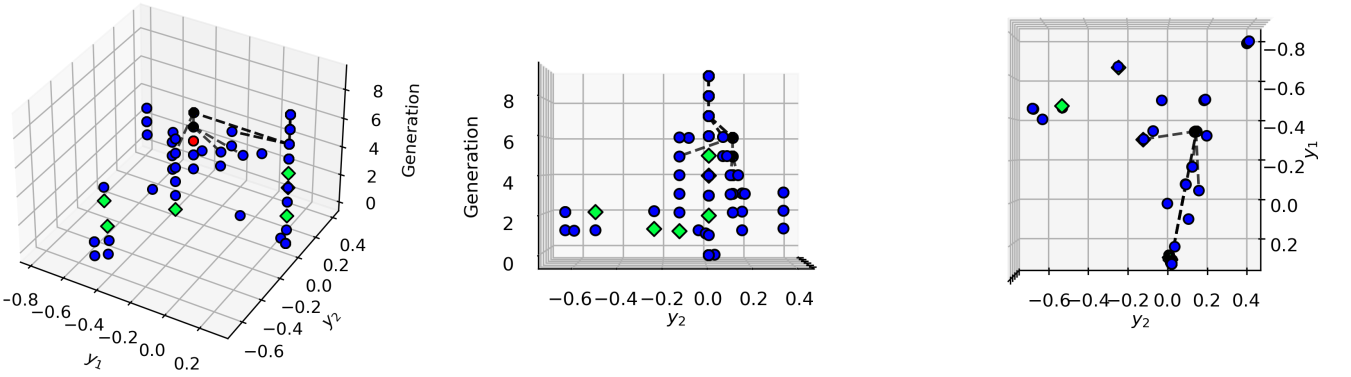
(f) Azimuth and elevation angles are 90° and 0° respectively.

(g) Azimuth and elevation angles are are 35° and −60° respectively.

(h) Azimuth and elevation angles are are 0° and 0° respectively.

(i) Azimuth and elevation angles are 90° and 0° respectively.

Figure 2: The dual-objective knapsack problem solutions. Row 1 has been generated with a mutation probability of 1.0 and crossover probability of 0. Row 2 has been generated with a mutation probability of 0 and crossover probability of 1. Row 3 has been generated with a mutation probability of 0.5 and crossover probability of 0.5. The ◆ solutions were created with mutation. The 🔵 solutions are non perturbed. The + solutions were created with crossover. The 🔴 solutions are infeasible solutions.



(a) Azimuth and elevation angles are are 35° and −60° respectively.

(b) Azimuth and elevation angles are are 0° and 0° respectively.

(c) Azimuth and elevation angles are 90° and 0° respectively.

Figure 3: The many-objective knapsack problem solutions generated with a mutation probability of 0.3 and crossover probability of 0.3. The ◆ solutions were created with mutation. The 🔵 solutions are non perturbed. The + solutions were created with crossover. The 🔴 solutions are infeasible solutions.

been created with crossover from parent solutions located in the far ends of the explored search space. We can see where the strong solutions tend to cluster. Again, we see the key search space areas of convergence, and we can follow the lineage back through time to see the evolution process of these dominant solutions.

4.4 Discussion on PopDP Explainability

As well as describing PopDP in terms of principles used within evolutionary computation, it is also important to consider the method through the lens of explainable AI (XAI). We, therefore, consider the nascent literature of XAI to benchmark the degree of explainability produced by PopDP.

Many taxonomies classify models into global, local or introspective methods, we would classify PopDP as a global method, given its ability to interpret the EA process at a holistic level.

We first recognise that not all AI requires a level of explainability [23]. Algorithm development/enhancement and showing important aspects of the solution journey would be the likely purpose of PopDP. For all these cases, explainability is necessary.

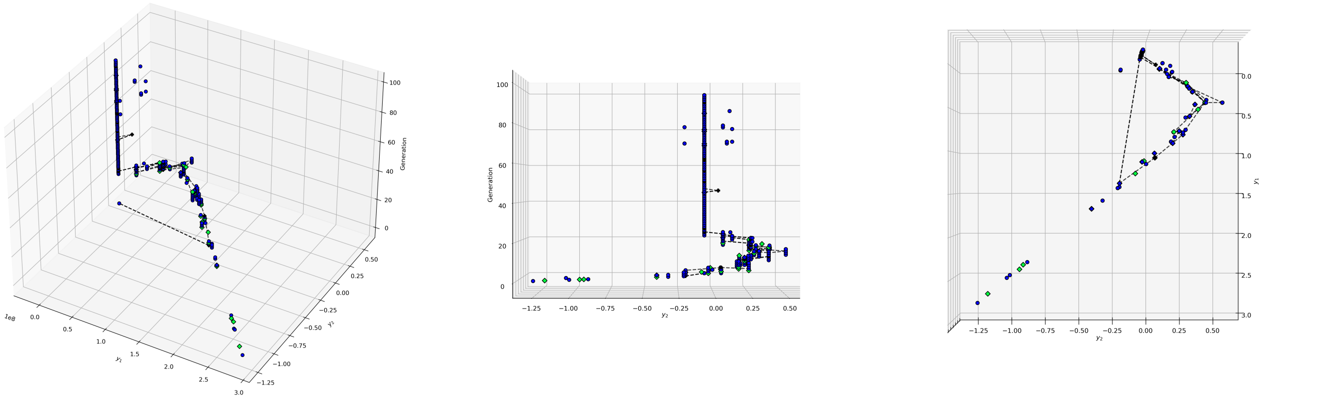
In XAI, we note that a one-size-fits-all explanation is non-existent. Therefore, it is important to consider the use case of PopDP and its likely user when analysing its explainability [1]. Given the use case of PopDP, we could expect the likely users to be either experts developing or enhancing EAs, or non-experts with some understanding of the general framework (i.e., an understanding of crossover and mutation). With the users' requirements and abilities understood, we can now contrast PopDP with the existing XAI literature benchmarks.

The work of [23] divides XAI into four principles: explanation, meaningful explanation, explanation accuracy and knowledge limits. The explanation is the evidence or reason(s) for the output decision. In the case of PopDP, this would be obtainable from the objective space, lineage and reproduction operators in the visualisation and, thus, this principle would be upheld. The last three principles are properties of the quality of explainability.

The second principle is the meaningfulness principle which is fulfilled if the intended recipient understands the system's explanation(s). In the case of PopDP, one can follow the evolutionary algorithm search through time to observe the evolution of the solutions. As the intended recipients are expected to have a basic understanding of the EA framework, combined with the information in PopDP it is assumed the user would be able to follow the evolution process of the solutions logically. The level of meaningfulness would vary depending on numerous factors including experience using the visualisation and the psychology and the cognitive ability of the user. Generally, someone operating with EAs would likely have some technical knowledge, making this a suitable visualisation. A usability study must be undertaken before we can confirm these conclusions.

The explanation accuracy ensures that the PopDP accurately displays the decision-making process. As we are directly projecting the solution operators, fitness and lineage, accurate information is obtainable from the PopDP, and this principle is thus upheld. We also note that, as the amount of information displayed increases, the fidelity of the explanation can be increased but the accessibility of the explanation for a lay audience could be reduced.

Finally, the work suggests that the extent to which a visualisation is explainable might be limited by issues relating to the visualisation technique in use. A relevant example in this work is the information loss inevitably incurred when using a dimension reduction technique such as LMDS. While we have not observed any issues in our work so far, we are actively investigating the extent to which scalability – for example, in terms of the number of generations an optimiser is run for or the number of problem objectives – has an effect on the clarity of the visualisation and the non-expert user's ability to glean useful explanations about the evolutionary process.



(a) Azimuth and elevation angles are are 35° and −60° respectively.

(b) Azimuth and elevation angles are are 0° and 0° respectively.

(c) Azimuth and elevation angles are 90° and 0° respectively.

Figure 4: The many-objective 'complex' knapsack problem solutions generated with a mutation probability of 0.2 and crossover probability of 0.6. The ◆ solutions were created with mutation. The ● solutions are non perturbed. The + solutions were created with crossover. The ● solutions are infeasible solutions.

Below are some other metrics used for benchmarking explainability in the existing literature. AI needs to demonstrate trustworthiness through explainability [1, 18, 24]. The objective of PopDP is to visualise the search. Hence, the additional level of insight obtained by PopDP could provide better understanding, confidence and trustworthiness through transparency of the inner workings of the search.

The work of [1] recognises the importance of transferability in AI. PopDP could highlight some of the EA's weaknesses (as we have done in the results section) and help define the EA boundaries of EA transferability within different problems. Interaction can enhance interpretability [1]. The PopDP is capable of being interactive. This level of interaction allows the recipient to engage the process with additional dimensions.

Other additional principles can be found in the literature of [1, 5, 14, 23, 27]. However, many other principles overlap with the principles we discussed or are not applicable to PopDP for optimisation (e.g., privacy) and so we have chosen to omit these.

5 CONCLUSION

We conclude the work by examining prospective avenues of research, limitations and the benefits of this research.

With regards to explainability, we have demonstrated how algorithm practitioners can use the PopDP visualisation to gain better insight into the algorithm process. In order to obtain a global explainability for Evolutionary Computing, we must first start by producing methods that are explainable to algorithm practitioners before developing methods for the layman. Moreover, visualisation is an effective foundation to develop XAI; it is a preferred method within the existing works of XAI, given the natural human ability to recognise visual patterns quickly [1, 27].

Visualisation intends to be informative, but not all visualisation is explainable. For a visualisation to be classed as XAI it should ultimately reveal insight into the black box optimiser, such as the inner mechanics or solution generation mechanisms (as illustrated with PopDP), not just illustrate solutions. Therefore most existing visualisations, i.e., a heatmap of the solutions, could not be classed as XAI methods.

The benefits of this work have been demonstrated and provide a framework to decode the black-box nature of evolutionary algorithms. However, the research also creates questions for future research avenues, such as considering the level of usability of PopDP by non-experts. Moreover, igniting questions relating to optimising the type and quantity of information displayed, although this is likely to be problem-dependent. And the extension of this work to continuous problems.

Furthermore, we can also envision how a large population size combined with a large crossover value could provide an overwhelming quantity of information that would be unbeneficial to the DM. However, as an interactive visualisation tool, this could be mitigated by focusing on specific regions of interest. We also note that most EA visualisations suffer from scalability issues, but we have demonstrated the effectiveness of PopDP for 100 generations. Other limitations include: for multi- and many-objective problems; some spatial information is lost during the dimension reduction process.

In this work, we have demonstrated how PopDP can visualise dual- and many-objective discrete knapsack problems. We have seen how PopDP can exhibit solution feasibility, solution generation mechanisms, solution child-parent lineage, potential algorithm artefacts and a visualisation of the objective space. Furthermore, we have seen how the visualisation provides a wealth of information to elucidate or ameliorate the difficulty of comprehending the process behind a black box nature evolutionary algorithm. This ultimately provides some level of transparency, interpretability and explainability of the DM's solutions and the mechanisms for which the solutions are generated.

REFERENCES

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

[2] Anastasia Bezerianos, Pierre Dragicevic, Jean-Daniel Fekete, Juhee Bae, and Ben Watson. 2010. Geneaquilts: A system for exploring large genealogies. *IEEE Transactions on Visualization and Computer Graphics* 16, 6 (2010), 1073–1081.

[3] Bogdan Burlacu, Michael Affenzeller, Michael Kommenda, Stephan Winkler, and Gabriel Kronberger. 2013. Visualization of genetic lineages and inheritance information in genetic programming. In *Proceedings of the 15th annual conference companion on Genetic and evolutionary computation*. 1351–1358.

[4] Angelos Chatzimparmpas, Rafael M Martins, Kostiantyn Kucher, and Andreas Kerren. 2021. VisEvol: Visual analytics to support hyperparameter search through evolutionary optimization. In *Computer Graphics Forum*, Vol. 40. Wiley Online Library, 201–214.

[5] Roberto Confalonieri, Ludovik Coba, Benedikt Wagner, and Tarek R Besold. 2021. A historical perspective of explainable Artificial Intelligence. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 11, 1 (2021), e1391.

[6] Matej Črepinšek, Marjan Mernik, and Shih-HsiLiu Liu. 2011. Analysis of exploration and exploitation in evolutionary algorithms by ancestry trees. *International Journal of Innovative Computing and Applications* 3, 1 (2011), 11–19.

[7] António Cruz, Penousal Machado, Filipe Assunção, and António Leitão. 2015. Elicit: Evolutionary computation visualization. In *Proceedings of the Companion Publication of the 2015 Annual Conference on Genetic and Evolutionary Computation*. 949–956.

[8] Jason M Daida, David J Ward, Adam M Hilss, Stephen L Long, Mark R Hodges, and Jason T Kriesel. 2004. Visualizing the loss of diversity in genetic programming. In *Proceedings of the 2004 Congress on Evolutionary Computation (IEEE Cat. No. 04TH8753)*, Vol. 2. IEEE, 1225–1232.

[9] Andrea De Lorenzo, Eric Medvet, Tea Tušar, and Alberto Bartoli. 2019. An analysis of dimensionality reduction techniques for visualizing evolution. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*. 1864–1872.

[10] Vin De Silva and Joshua B Tenenbaum. 2004. *Sparse multidimensional scaling using landmark points*. Technical Report. technical report, Stanford University.

[11] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and TAMT Meyarivan. 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE transactions on evolutionary computation* 6, 2 (2002), 182–197.

[12] Bogdan Filipič and Tea Tušar. 2018. A taxonomy of methods for visualizing Pareto front approximations. In *Proceedings of the Genetic and Evolutionary Computation Conference*. 649–656.

[13] Martin Fyvie, John AW McCall, and Lee A Christie. 2021. Towards Explainable Metaheuristics: PCA for Trajectory Mining in Evolutionary Algorithms. In *International Conference on Innovative Techniques and Applications of Artificial Intelligence*. Springer, 89–102.

[14] 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.

[15] Emma Hart and Peter Ross. 2001. Gavel-a new tool for genetic algorithm visualization. *IEEE Transactions on Evolutionary Computation* 5, 4 (2001), 335–348.

[16] Andreas Kerren and Thomas Egger. 2005. Eavis: A visualization tool for evolutionary algorithms. In *2005 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC’05)*. IEEE, 299–301.

[17] Marwa Keshk. 2017. Semantic Evolutionary Visualization. In *International Conference on Swarm Intelligence*. Springer, 624–635.

[18] Been Kim, Elena Glassman, Brittney Johnson, and Julie Shah. 2015. iBCM: Interactive Bayesian case model empowering humans via intuitive interaction. (2015).

[19] Nicholas Freitag McPhee, Maggie M Casale, Mitchell Finzel, Thomas Helmuth, and Lee Spector. 2016. Visualizing genetic programming ancestries. In *Proceedings of the 2016 on Genetic and Evolutionary Computation Conference Companion*. 1419–1426.

[20] Nicholas Freitag McPhee, Maggie M Casale, Mitchell Finzel, Thomas Helmuth, and Lee Spector. 2017. Visualizing genetic programming ancestries using graph databases. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*. 245–246.

[21] Nicholas Freitag McPhee, Mitchell D Finzel, Maggie M Casale, Thomas Helmuth, and Lee Spector. 2018. A detailed analysis of a PushGP run. In *Genetic Programming Theory and Practice XIV*. Springer, 65–83.

[22] 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.

[23] P Jonathon Phillips, Carina A Hahn, Peter C Fontana, David A Broniatowski, and Mark A Przybocki. 2020. Four principles of explainable artificial intelligence. *Gaithersburg, Maryland* (2020).

[24] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. " Why should I trust you?" Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. 1135–1144.

[25] Lukas Sekanina and Vlastimil Kapusta. 2016. Visualisation and analysis of genetic records produced by Cartesian genetic programming. In *Proceedings of the 2016 on Genetic and Evolutionary Computation Conference Companion*. 1411–1418.

[26] William B Shine and Christoph F Eick. 1997. Visualizing the evolution of genetic algorithm search processes. In *Proceedings of 1997 IEEE International Conference on Evolutionary Computation (ICEC’97)*. IEEE, 367–372.

[27] Giulia Vilone and Luca Longo. 2020. Explainable artificial intelligence: a systematic review. *arXiv preprint arXiv:2006.00093* (2020).

[28] David J Walker, Richard M Everson, and Jonathan E Fieldsend. 2012. Visualizing mutually nondominating solution sets in many-objective optimization. *IEEE Transactions on Evolutionary Computation* 17, 2 (2012), 165–184.

[29] Mathew J Walter, David J Walker, and Matthew J Craven. 2022. Visualising Population Dynamics to Examine Algorithm Performance. *IEEE Transactions on Evolutionary Computation* in press (2022).

[30] Annie S Wu, Kenneth A De Jong, Donald S Burke, John J Grefenstette, and C Loggia Ramsey. 1999. Visual analysis of evolutionary algorithms. In *Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406)*, Vol. 2. IEEE, 1419–1425.

[31] Eckart Zitzler and Lothar Thiele. 1999. Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach. *IEEE transactions on Evolutionary Computation* 3, 4 (1999), 257–271.