2022-03-07

Visualising Population Dynamics to Examine Algorithm Performance

Walter, Mathew

10.1109/TEVC.2022.3157143
IEEE Transactions on Evolutionary Computation
Institute of Electrical and Electronics Engineers

Visualising Population Dynamics to Examine Algorithm Performance

Mathew J. Walter, David J. Walker, and Matthew J. Craven

***Abstract*—This work assesses the efficacy of evolutionary algorithms (EAs) using an intuitive Multi-Dimensional Scaling (MDS) visualisation of the evolution of a population. We propose the use of Landmark Multi-Dimensional Scaling (LMDS) to overcome computational challenges inherent to visualising many-objective and complex problems with MDS. For the benchmark problems we tested, LMDS is akin to MDS visually, whilst requiring less than 1% of the time and memory necessary to produce an MDS visualisation of the same objective space solutions, leading to the possibility of online visualisations for multi- and many-objective optimisation evaluation. Using multi- and many-objective problems from the DTLZ and WFG benchmark test suites, we analyse how Landmark MDS visualisations can offer far greater insight into algorithm performance than using traditional algorithm performance metrics such as hypervolume alone, and can be used to complement explicit performance metrics. Ultimately, this visualisation allows visual identification of problem features and assists the decision maker in making intuitive recommendations for algorithm parameters/operators for creating and testing better EAs to solve multi- and many-objective problems.**

***Index Terms*—Visualisation, Multi- and Many-objective Optimisation, Landmark Multi-Dimensional Scaling**

I. Introduction

EVOLUTIONARY algorithms (EAs) have become an increasingly popular strategy for solving optimisation problems abound in science and industry, providing many benefits over traditional non-evolutionary computation strategies. However, understanding the processes by which EAs generate solutions can be non-trivial for non-expert decision makers (DMs), limiting the control and hence the effectiveness of the EA. Visualisation is a natural approach to address this challenge, although the solution can be high dimensional and comprehending greater than three spatial dimensions yields difficulties. In this work, we visualise the objective space of multi- and many-objective continuous optimisation problems. The visualisation will indirectly show the problem landscape and directly show the performance of the algorithm, inferring the visualisation could be highly efficient for examining algorithm performance. One will be able to observe the population perturbations through the search space, identify where the algorithm encounters difficulty (e.g., at local optima) and how the mutation and crossover parameters are affecting the search.

The visualisation is adapted from the work of [1], which was subsequently extended to multi-objective problems [2]. The initial method was used to visualise simple single-objective problems in which it was effective at showing population dynamics, but could not visualise problems requiring a large number of function evaluations because of memory and computational time requirements. We adapt the visualisation for arbitrary runtime use. We evaluate the performance of the adaptation using both implicit (graphical interpretation) and explicit (statistical ranked RMSE evaluation) representation methods. This visualisation can examine the population at the individual solution level and hence reveal a significant depth of information for examining EA effectiveness - a tool which would be of great use to the EA community.

This paper offers the following novel contributions:

1) We identify and adapt a visualisation for algorithm performance evaluation, understanding problem characteristics and comprehending an EA's performance.
2) We modify the existing visualisation using Landmark Multi-dimensional Scaling [3] to work for complex problems requiring a large number of function evaluations and evaluate its effectiveness. This includes visualising many-objective problems.
3) We developed the aforementioned visualisation as an interactive tool for greater DM control and potential online evaluation of algorithm performance, freely available for application in industry, education and research.

The remainder of this paper is structured as follows. In Section II, we review existing work on visualising many/multi-objective optimisation. Section III contains the methodology implemented for visualisation and the MDS/LMDS process is detailed. The results and the experimental setup, containing details of the parameters used, are highlighted in Section IV. Section V hosts the results and analysis of LMDS for evaluating algorithm performance. We discuss future work and provide a conclusion in Section VI.

II. Existing work

EVOLUTIONARY algorithms can be used to evolve high-dimensional solutions for multi- and many-objective problems. Visualising high-dimensional data is non-trivial. The work of [4] suggests that visualisation techniques can fall into three categories, namely: methods representing the data in their full objectives, visualising data with a reduced number of objectives, or transforming the coordinate system. For a comprehensive taxonomy of EA visualisation, see the work of [5]. We will consider some of the most significant EA visualisation methods in this section.

Initially, EA solutions were predominantly visualised with objective space plots, pairwise coordinate plots [6] and parallel coordinate plots (PCP) [7, 8]. These methods consider visualising all objectives. However, as the number of objectives

M. J. Walter, D. J. Walker and M. J. Craven are with the School of Engineering, Computing and Mathematics, University of Plymouth, U. K., PL4 8AA. e-mail:(mathew.walter,david.walker,matthew.craven)@plymouth.ac.uk

Manuscript received XXXXXXXX; revised XXXXXXXX.

$m > 3$ grows, the number of required scatter plots to visualise all objectives very quickly becomes unmanageable $(m(m-1)/2)$; this phenomenon is known as *the curse of dimensionality*. As the number of objectives in a PCP plot increases, the visualisation becomes cluttered and challenging to interpret. Other methods including radial coordinate visualisation [9] and 3-D radial coordinate visualisation [11] (which added an additional dimension) are used to visualise the approximation front based on representing the objectives as units on a circle. These are examples of methods to represent data in a transformed coordinate system. Radial coordinate visualisation preserves the approximation front distribution, but not the geometry of the front.

Some of the earliest work visualising the population was the work of [12], using predominately 2-D or 3-D line plots and proposed implementing MDS for EA population visualisation. Bubble charts were used in the work of [13]. Bubble charts were simple but could at most only visualise 5-objective data.

Later, visualisations including Self Organising Maps (SOM) [14] and Distance and Distribution Charts [15] were introduced to the EA domain. SOM provided a topology preserving mapping whilst projecting the approximation front into a lower dimension. It did not preserve the dominance relationship or the approximation front's geometry. Although scalable to many objectives, SOMs were challenging to interpret.

Isomap [16], Neuroscale [17, 18] and Sammon mapping [19, 20] all use non-linear mapping to project high dimensional approximation fronts into a lower dimension. Neither of these methods are considered simplistic or preserve the geometry of the solutions. Level diagrams [21] are an alternative to dimension reduction. This consists of a matrix of scatter plots with the x-axis displaying the objective value and the y-axis displaying the distance from the ideal point. However, this suffers from the same limitations as scatter plots in that increasing the number of objectives rapidly increases the number of visualisations.

Other methods that seek to map the original approximation front into a new coordinate system are the prosection method [22] and the polar coordinate system [23]. The prosection method can visualise 4-objective approximation fronts in 3-objectives whilst persevering many of the approximation front characteristics. However, it is difficult to scale for $m > 4$ objectives. More recently, the work of [24] provides a visualisation for understanding high dimensional Pareto optimal datasets.

While many visualisations of the approximation front exist, there is scarce literature for a single intuitive visualisation that can visualise the entire population of a multi- and many-objective EA process. The majority of previously discussed visualisations only consider visualising the approximation front (usually a much smaller subset of the solutions generated during optimisation), revealing only information about the approximation front and neglecting the population dynamics, and thus produce inadequate insight into a *black-box* algorithm.

Arguably the best method for visualising approximation fronts depends on the approximation solutions and it is unlikely that any one of the visualisations described above can usefully visualise all aspects of all approximation sets, in a similar way that the *No-Free-Lunch theorem* precludes the existence of a single optimiser that can optimise all problems [4]. Many of the visualisation limitations are likely to cause difficulties when trying to visualise an entire population. For example, a method that does not preserve the approximation front geometry is not likely to preserve the population structure's geometry. Additionally, methods that become complex when using large approximation fronts are not likely to be suitable for visualising populations on their own, while storage and computational complexity become of much greater importance for visualising entire populations.

Many of these methods are considered effective for visualising approximation fronts as *a posteriori* methods where no time constraint exists. Many of these approximation front visualisation methods (unadapted) would be unsuitable for visualising large populations, eliminating their ability to visualise whole populations and use in online environments because of computation complexity and large population size.

Whilst scarce, attempts have been made to visualise the evolutionary process. For example, the work visualising the population entirety proposed by [25] visualises the population for examining algorithm parameters by mapping individuals from the high dimensional objective space to a 2-D polar coordinate space. Other work visualising the performance of EAs include [26] - these consider visualising performance indicators of EAs with different parameters. Other studies that illustrate algorithm performance are that of [27] which considers a single-objective cryptography problem. The work of [28] utilises scatter plot animations to study the mechanics of MOEAs to evaluate algorithm performance.

The work of [29] considers visualising the decision space of continuous multi-objective problems (MOPs) proposing a 'PLOT' visualisation comprised of a gradient field heatmap and cost landscapes. Whilst this method does not visualise the entire population, one can observe some problem features such as local optima, basins of attraction and global optima. It notes visualisation techniques for the decision space of continuous MOPs are scarce in research (there being only two existing visualisations that consider the decision space) [29].

Other methods use dimensionality reduction techniques such as MDS and PCA from the machine learning domain [30, 31]. Later work [1] compared the extent to which dimension reduction techniques preserved population movements and the exploration-exploitation trade-off using single-objective problems. It also proposed two compact visualisations, one of which is extended in [2] to visualise the decision space and objective space history of an EA optimising multi- and many-objective problems. However, due to the visualisation method requiring the creation of a single distance matrix of the entire population, the visualisation has limitations for more complex problems. For example, a 3-objective problem requiring 100,000 function evaluations would require approximately 80GB of RAM using Numpy's float64 data type to store a Euclidean distance matrix (or approximately 40GB if stored as an upper triangular matrix, as the Euclidean distance matrix is symmetric). The current paper adapts the visualisation to overcome this limitation and demonstrates the visualisations

possible effectiveness at algorithm performance evaluation, understanding problem characteristics and comprehending the performance of a given EA - we do note that a user study would be required to confirm the benefits of this visualisation and we only make the claim that this may be possible from demonstrations. As with the PLOT visualisation, the LMDS reduced visualisation can also be employed on the decision space, as well as the objective space.

III. Multi-Dimensional Scaling Visualisation Methodology

TO visualise the population, we apply a dimension reduction technique to configure a more visually manageable 2-D representation of the population in the embedded space which can be plotted in a series of scatter plots. There are many different data reduction techniques which may be implemented for the purposes of visualisation. In this work we chose Multi-Dimensional Scaling [3] (MDS), based on the results of [1] which examines the efficacy of preserving the population structure for single-objective problems. MDS is also known to preserve solution structure in multi- and many-objective populations [31].

A. Multi-Dimensional Scaling

The purpose of MDS in this work is to derive a set of vectors which reduces the EA population to a lower dimensional embedding whilst preserving pairwise population distances. 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 $i, j \in \{1, \ldots, N\}$. The notation $|| \cdot ||$ denotes the standard Euclidean norm.

There are three commonly used types of MDS algorithms, namely: classical MDS (also known as Principle Coordinates Analysis (PCoA)), metric MDS and non-metric MDS. All MDS algorithms require a similarity matrix input, though the algorithm employed depends on the input matrix. For non-metric MDS the similarity matrix is qualitative (given by rank only) and therefore we discard the use of non-metric MDS from our visualisations. Thus, we can consider classical MDS and metric MDS for visualising EA populations. Although metric MDS would have produced a very similar visualisation to this work, we arbitrarily implement classical MDS for the visualisations.

Classical MDS uses eigenvalue decomposition to find a lower dimension k embedding matrix $\mathbf{Y}$. We start by creating the $N \times N$ squared proximity matrix $\mathbf{D}^{(2)} = \left(d_{i,j}^2\right)$. Using the centering matrix $\mathbf{C} = \mathbf{I} - \frac{1}{N}\mathbb{O}$ (where $\mathbf{I}$ is the identity matrix and $\mathbb{O}$ is the $N \times N$ matrix filled with ones) double centering is then applied to $\mathbf{D}^{(2)}$ to create matrix $\mathbf{B}$:

$$\mathbf{B} = -\frac{1}{2}\mathbf{C}\mathbf{D}^{(\mathbf{2})}\mathbf{C} \tag{1}$$

With $\mathbf{B} = \mathbf{Y}\mathbf{Y}'$, since $\mathbf{B}$ is symmetric (and now centered) we can use eigenvalue decomposition to decompose $\mathbf{B}$ into $\mathbf{E}\mathbf{\Lambda}\mathbf{E}'$, where $\mathbf{E}$ is a matrix of eignenvectors and $\mathbf{\Lambda}$ is a diagonal matrix of eigenvalues to determine the k largest eigenvalues $\lambda_1, \ldots, \lambda_k$ and their corresponding eigenvectors (non-positive eigenvalues are ignored). Finally, the $k \times N$ matrix $\mathbf{Y}$ can be constructed by

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

B. Landmark Multi-Dimensional Scaling

MDS has its limitations. One major limitation when considering the application of MDS to a large EA population is the substantial amount of RAM required to store a Euclidean distance array (often many gigabytes for the DTLZ [32] test suite requirements). Classical MDS requires $O(N^2)$ storage, where N is the population size. Classical MDS also has a complexity of $O(CN^2 + N^3)$, where C is the cost of computing and accessing each entry of $\mathbf{D}$. The MDS bottleneck is the PCA which for k vectors can be computed in $O(kN^2)$ time using the power method [33], and so as N increases, the computation time becomes impractical for use with EA population visualisations. To overcome this problem we use a modified form of MDS termed Landmark MDS (LMDS) [34].

LMDS is considered an approximation to MDS; however, the results of this work will show this also to be a good visual approximation for EA populations. For landmark MDS the required Euclidean distance matrix storage is $O(nN)$ and the computation requires $O(CnN + knN + n^3)$ time, where n is the number of landmarks and k is the dimensionality of the input matrix. For more on LMDS complexity see [34]. The bottleneck of MDS is in the eigenvalue decomposition of the $N \times N$ proximity matrix. LMDS overcomes this complexity challenge by only utilising a subset of the entire population matrix for eigenvalue decomposition. This subset is referred to as the set of landmark points, and the remaining solutions use a faster computing distance-based triangulation procedure to decide their position.

To implement LMDS we need to designate a subset of n landmark points from similarity matrix $\mathbf{D}_N$. We create matrix $\mathbf{D}_n$ and apply MDS to this $n \times n$ matrix as we described with classical MDS previously to create a $k \times n$ matrix $\mathbf{Y}_K$. After embedding the landmark points in $\mathbb{R}^k$, we now need to embed the remaining points using the landmark points and an affine linear transformation. We use distance-based triangulation:

1) Let $\mathbf{d}_i$ denote the ith column of the squared proximity matrix $\mathbf{D}_n^{(2)}$ for $i = 1, \ldots, n$.
2) Compute the mean $\mathbf{d}_\mu = (\mathbf{d}_1 + \mathbf{d}_2 + \ldots + \mathbf{d}_n)/n$.
3) With the eigenvalues and eigenvectors obtained in the previous steps calculating $\mathbf{Y}_k$, we then compute the pseudoinverse transpose $\mathbf{Y}_k^{\#}$ of $\mathbf{Y}_k$ with the formula
$$\mathbf{Y}_k^{\#} = \mathbf{e}_i^T / \sqrt{\lambda_i}$$
for all $i = 1, \ldots, k$.
4) To position a remaining point $\mathbf{X}_a$, compute the embedding vector $\mathbf{X}_a = -\frac{1}{2}\mathbf{Y}_k^{\#}(\mathbf{d}_a - \mathbf{d}_\mu)$, where $\mathbf{d}_a$ is the vector of squared distances between the point $\mathbf{X}_a$ and the n landmark points.

It is important to note that for a k-dimensional embedding we require at least $k + 1$ landmarks.

C. Visualisation Methodology

We run an EA for a set number of function evaluations, often until the algorithm has converged to the Pareto set/front or until we reach a pre-defined generation that we are interested in visualising. We save the objective space population coordinate data and the generation in which that solution belongs. Once the EA has reached a terminal state, we can apply MDS/LMDS to the objective space population coordinate data to reduce the population dimensionality to two dimensions. Finally, we project the population in the z-axis based on the generation it belongs to, giving a 3-D plot. We can then colour the plot according to some metric to add additional information.

Visualisations are coloured depending on the state in which the algorithm is operating, either exploring the search space or exploiting an area of high quality in the search space. The exploration/exploitation metric is used in [1], proposed by [35]. The exploration/exploitation metric is calculated as follows: for each solution in each generation g, calculate the Euclidean distance to the solution's closest neighbour and store these values in the similarity to the closest neighbour (SCN) array. This can be any neighbouring solution in the population up to the current generation. We then calculate the median, SCN^*, of SCN. For all $SCN_g > SCN^*$ we add 1 to the exploration/exploitation metric, τ_g, for each generation. Finally, when all τ_g values have been calculated for every generation, we normalise and colour the plots accordingly.

The colouring could also be based on other metrics such as generation number, although we feel exploration/exploitation colouring is more informative for this work. There exist limitations to this metric, such as the raw values themselves not being significantly informative (hence not being included in the visualisations) and the colouring is problem dependant. We colour the non-dominated solutions as white crosses.

We adapt LMDS for EA population visualisation by choosing landmark points that contribute significant value to preserving the MDS solution structure in the LMDS space. We, thus, ensure the landmark points are uniformly randomly chosen solutions from all generations to produce an equal distribution of solutions from across the generations.

For the visualisations in this work, we chose to visualise the objective space population, although the same method would allow one to visualise the decision space in the same way.

The Python implementation of the visualisations is open source at: https://github.com/MathewWalter/LMDS-Evolutionary-Algorithm-Visualisation.

IV. Visualisation Results and Analysis

THE purpose of this section is to analyse the visualisations to identify key characteristics of the optimisation problems. We aim to identify a problem feature mapping from the objective space to the MDS/LMDS reduced objective space visualisation. We start with 3-objective problems to compare the MDS/LMDS reduced plot with the objective space. Once the mappings are understood, we apply the knowledge to analyse many-objective problems.

A. Experimental Parameters

The problems are optimised with NSGA-III [36] unless stated otherwise. The crossover probability is 0.8, and the mutation probability is set to 0.1. The distribution index, controlling the size of the perturbation, in both cases is fixed (15 for SBX, 7 for polynomial mutation). The number of divisions in the boundary layer used in the automatic generation of reference points is 12. Note that since the algorithm termination condition is checked after each iteration, it can be possible for the algorithm to exceed the predefined function evaluations (by a maximum of a single population size). A different number of function evaluations have been used in the different problems. The number of function evaluations for each problem can be identified in the figure caption. To demonstrate certain artefacts of the search, we have terminated some runs early, as indicated in the figure caption.

1) WFG Parameters: The WFG test suite problems [37] used in this work, with the exception of a WFG convex problem (Figure 3), all have a concave-shaped Pareto front. For multi-modal problems, the number of optima is controlled by transformation functions [37] yielding 6 optima (5 local optima and 1 global optimum), and the global optima is located at 0.35 in the first objective, with a magnitude of 2.

2) DTLZ parameters: The experiment comprises of running EAs on four continuous problems from the DTLZ test suite [32], namely DTLZ1, 2, 4 and 7. These problems have real-valued decision variables lying in the region $[0, 1]$. The suggested number of decision variables is $p = \kappa+m-1$, where $m-1$ is the number of position parameters and κ is the number of distance parameters, controlling the distance the solutions are from the Pareto front. In this work, $\kappa = 5$ for DTLZ1, $\kappa = 20$ for DTLZ7, and $\kappa = 10$ for the remaining problems. The problems are scalable in the number of objectives; in this experiment, three- and five-objective problems are utilised.

B. Results for Low Numbers of Function Evaluations

To identify a mapping, one would first need to implement the visualisations on a problem suite in which the quantity and type of features can be controlled by the user. We, therefore, construct 3-objective test problems using the WFG test suite, where all other problem features excluding the features we intend to identify are identical to observe a mapping from objective to manifold embedding space. Using the WFG test suite, we generate a unimodal problem with a concave Pareto shape (Figure 1, left) and a unimodal problem with a convex Pareto shape (Figure 1, right). We can see how the geometry of the non-dominated solutions, coloured white, is conserved in the MDS objective space (Figure 2 and Figure 3 respectively). For both MDS visualisations, we can see the path of the population converging to each non-dominating solution. There are no local optima, and this can be detected by a continuous path of solutions to the global optima. For concave Pareto front geometries, we can see a higher density of solutions on the edges of the MDS population structure and a lower density of solutions in the centre of the MDS population; this effect is reversed for convex shapes. For Figure 2, we can observe the initial solutions quickly converging to the optima by

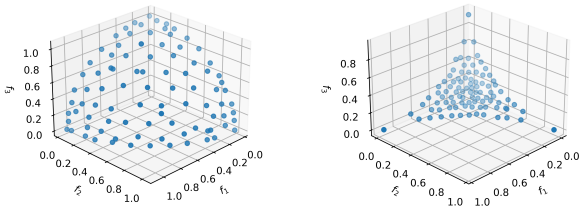


Fig. 1. The Pareto front of a unimodal concave WFG problem in three objectives (left) and the Pareto front of a unimodal convex WFG problem in three objectives (right).

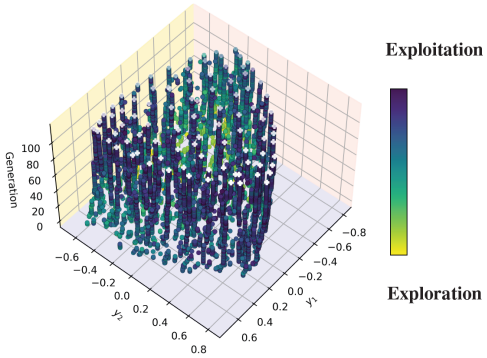


Fig. 2. The MDS reduced objective space of a unimodal concave WFG problem in three objectives produced with 10,000 function evaluations.

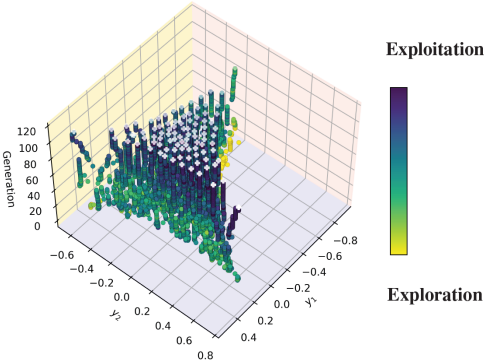


Fig. 3. The MDS reduced objective space of a unimodal convex WFG problem in three objectives produced with 10,000 function evaluations.

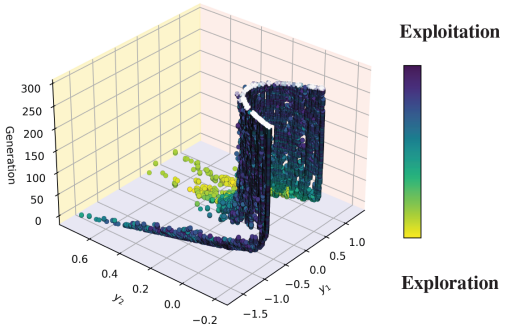


Fig. 4. The MDS reduced objective space of DTLZ4 in three objectives produced with 25,000 function evaluations. The EA process has been terminated early to demonstrate population movements.

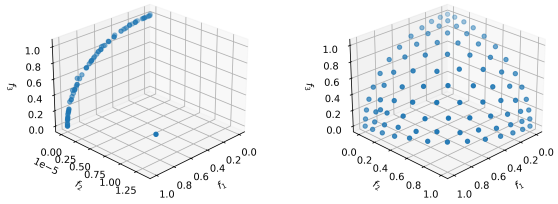


Fig. 5. The left visualisation shows the current approximated front of DTLZ4. The EA process has been terminated early to demonstrate population movements, for which we can see the bias. The right visualisation shows the Pareto front of DTLZ4 in three objectives. EA process complete.

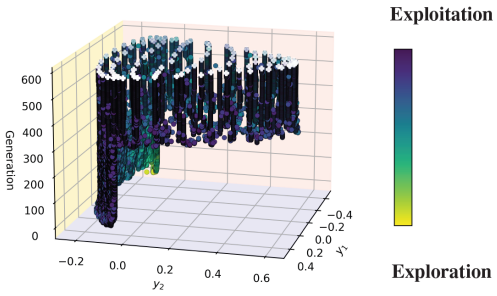


Fig. 6. The LMDS reduced objective space of DTLZ4 in three objectives, with 5000 landmarks and 50,000 function evaluations. EA process complete.

looking at the relatively small difference in the initial solution (y_1, y_2) positions to the final solutions (y_1, y_2) positions. This lack of solution exploration indicates that the initial solutions generated are close to the Pareto front positions.

In Figure 4, we visualise DTLZ4 in three objectives, terminating the evolutionary run before the solutions have converged to the Pareto front. DTLZ4 solutions encounter bias which can be seen in the objective space (Figure 5) by solutions only initially forming on the f_1 and f_3 plane. The MDS accurately represents the geometry of the solutions when compared to the objective space in Figure 5 (left). We can also interpret how NSGA-III's reference plane selection operation is affecting the choice of population solutions, and we can see from the MDS these solutions converging in lines towards the Pareto front. We could manipulate the reference plane points distribution to create a population search weighting, allowing the EA to search different areas of the search space, which could be useful for an online visualisation tool. Subsequently, we continue the full evolutionary run to visualise how the population overcomes the bias of the DTLZ4 problem. The large number of function evaluations required to overcome the bias would necessitate the use of LMDS to visualise the solutions. This can be seen in Figure 6, in which we see the population exhibit greater exploration as the solutions spread across the entire Pareto front. This can be seen in the figure, as generation 300 (approximately) contains solutions of a slightly lighter green colour than the previous generation, indicating a temporary phase of greater exploration. This demonstrates how the visualisation can be used to create a deeper insight into the optimisation process.

Finally, for the remaining low function evaluation problems, we consider DLTZ7 in three objectives. DTLZ7 (Figure 7)

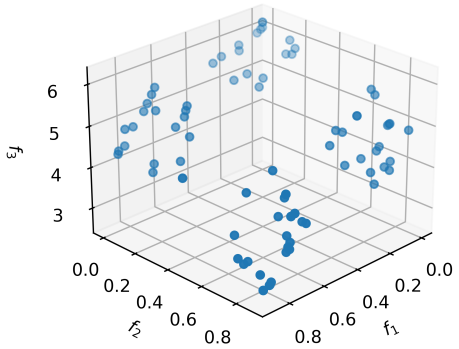


Fig. 7. The Pareto front of DTLZ7 in three objectives produced with 19,000 function evaluations.

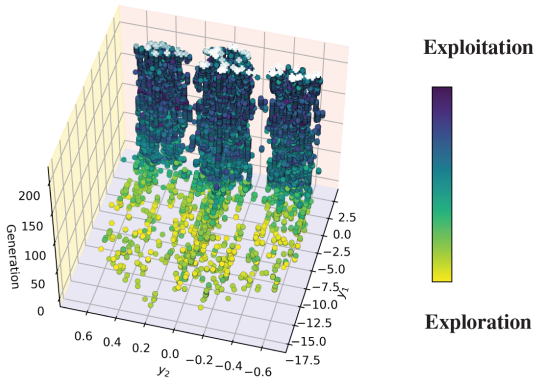


Fig. 8. The MDS reduced objective space of DTLZ7 in three objectives produced with 19,000 function evaluations.

contains a disconnected Pareto front resulting in four clusters of solutions. In the MDS plot (Figure 8), we can see how an almost even distribution of initial solutions evolves quickly to four population clusters. The exploration/exploitation colouring identifies the initial solutions in an exploring state and once the clusters of solutions begin to form the colouring becomes much darker to signify exploitation to the Pareto front. We can see the path the solutions are taking and determine which generation the solutions categorise into their cluster and begin to yield an exploitative state.

C. Results for High Numbers of Function Evaluations

As exposed in the previous results, MDS produces intuitive visualisations of the evolution process and the non-dominating solutions. However, as the number of function evaluations required to produce the visualisations increases, the memory and time required to produce the Euclidean distance matrix required for MDS also significantly increases.

One method to overcome the difficulty in computing the MDS projection of a single large array is to use the method proposed in the work of [2]. This method produces MDS visualisations at each generation, requiring additional but far smaller (population size) matrices. This overcomes the large array computation challenges to develop faster visualisations for online visualisations and visualising problems which require large populations or many generations to optimise. In contrast, the method has its limitations - the MDS plots are not aligned, and so population movements between generations are not as clearly mapped as the method proposed in [1].

A further technique to overcome computational difficulties encountered with large Euclidean distance matrices involves adapting the method proposed in [1], making this method applicable to more complex problems than the single objective problems applied in the original work, but maintaining a standardised orientation of the MDS visualisation. To attain these requirements, we propose using Landmark MDS for visualising the evolution process for complex or many-objective problems requiring a greater number of function evaluations. In this section, we demonstrate how LMDS is akin to MDS; however, LMDS can significantly reduce the computational cost of visualising, as detailed in Section III.

To show the similarity between MDS and LMDS, we employ the RMSE metric to a ranked Euclidean distance matrix. We use the word 'similarity' to describe the ranked error between MDS and LMDS visualisations. We implement the use of ranked distances rather than the Euclidean distances to identify the closeness ordering of solutions. Strikingly, one such effect of using the ranked RMSE is obtaining large numerical values - this is owing to the large population size (approximately 10,000 solutions to be ordered). We vary the size of the landmark points matrix and consider the similarity with MDS and its computation time. For each problem and number of landmarks, we compute thirty RMSE measurements and the mean and standard deviation of the collection of measurements. Equation 3 is implemented to calculate the RMSE and the results are shown in Table I.

$$RMSE = \left(\frac{1}{N_g} \sum_{h=1}^{N_g} \left(\sum_{i=1}^{P} \sum_{j=1, j \neq i}^{P} \left(d_{i,j}^{M} - d_{i,j}^{L} \right)^2 \right) \right)^{\frac{1}{2}} \quad (3)$$

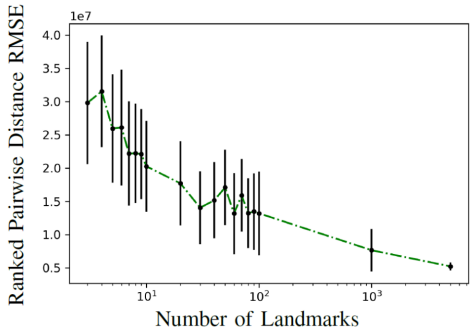
Here, N_g is the number of generations, P is the population size, $d_{i,j}^{M}$ and $d_{i,j}^{L}$ are the MDS distance matrix and LMDS distance matrix, respectively. Table I and the graphs in Figure 9 show decreasing error as the number of landmark points increases. Notably, the number of landmarks and RMSE relationship is not linear. One can therefore determine an effective (problem dependent) optimal number of landmarks. For DTLZ1 and DTLZ2 we can approximate the knee point lying between 10 and 100 landmarks, allowing one to effectively approximate MDS visualisations with just 1% of the total population. This results in a drastic reduction in the required memory and time for computation; for instance, for DTLZ1 the computation time of LMDS is less than 0.02% of the time required to complete the MDS. For these visualisations, fewer than 1% of the population was required to produce LMDS that was visually indistinguishable to MDS when observed. When using greater than 1% of the population, the improvement in the LMDS was small and difficult to detect by eye. DTLZ2 portrays a similar story, although we note fewer landmarks are required to produce a similar error as seen in DTLZ1.

D. Further Analysis with MDS/LMDS

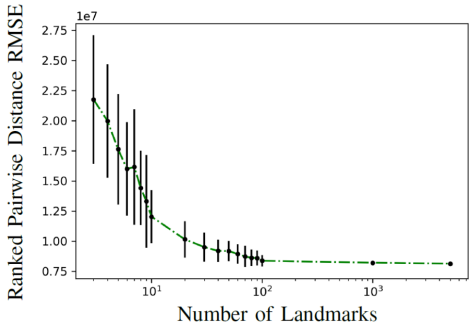
Unimodal problems, as shown in Figure 2, often contain a continuous central mass of solutions (or n-masses for prob-

TABLE I
COMPUTATION TIME, MEAN μ AND STANDARD DEVIATION σ OF RMSE RANKED PAIRWISE DISTANCES ACROSS 30 RUNS ON DTLZ1 AND DTLZ2 IN THREE OBJECTIVES, 10,000 FUNCTION EVALUATIONS. MDS COMPUTATION TIMES WERE 326.37 S (DTLZ1) AND 286.07 S (DTLZ2).

Problem	Landmarks	RMSE ($\mu \pm \sigma$)	LMDS Time (s)
DTLZ1	3	$3.07 \times 10^7 \pm 9.29 \times 10^6$	0.0112
	4	$3.24 \times 10^7 \pm 8.51 \times 10^6$	0.0018
	5	$2.68 \times 10^7 \pm 8.28 \times 10^6$	0.0021
	6	$2.70 \times 10^7 \pm 8.81 \times 10^6$	0.0022
	7	$2.30 \times 10^7 \pm 7.99 \times 10^6$	0.0026
	8	$2.30 \times 10^7 \pm 7.62 \times 10^6$	0.0027
	9	$2.30 \times 10^7 \pm 6.93 \times 10^6$	0.0031
	10	$2.12 \times 10^7 \pm 6.98 \times 10^6$	0.0037
	50	$1.49 \times 10^7 \pm 5.85 \times 10^6$	0.0202
	100	$1.46 \times 10^7 \pm 5.85 \times 10^6$	0.0366
	1000	$8.20 \times 10^6 \pm 3.75 \times 10^6$	0.6459
	5000	$5.77 \times 10^6 \pm 5.79 \times 10^5$	31.9253
DTLZ2	3	$2.26 \times 10^7 \pm 5.31 \times 10^6$	0.0051
	4	$2.08 \times 10^7 \pm 4.76 * 10^6$	0.0018
	5	$1.84 \times 10^7 \pm 4.65 \times 10^6$	0.0022
	6	$1.68 \times 10^7 \pm 3.97 \times 10^6$	0.0020
	7	$1.69 \times 10^7 \pm 4.89 \times 10^6$	0.0023
	8	$1.52 \times 10^7 \pm 3.17 \times 10^6$	0.0025
	9	$1.40 \times 10^7 \pm 3.89 \times 10^6$	0.0024
	10	$1.28 \times 10^7 \pm 2.22 \times 10^6$	0.0028
	50	$9.31 \times 10^6 \pm 8.86 \times 10^5$	0.0186
	100	$9.33 \times 10^6 \pm 6.63 \times 10^5$	0.0357
	1000	$8.81 \times 10^6 \pm 1.57 \times 10^5$	0.6330
	5000	$8.86 \times 10^6 \pm 6.73 \times 10^4$	32.3680



(a) DTLZ1



(b) DTLZ2

Fig. 9. The MDS and LMDS ranked RMSE results for DTLZ1 and DTLZ2 in three objectives for 10,000 function evaluations.

lems with n-disconnected fronts) with an occasional solution lying outside these columns. These solutions are solutions that have been mutated to a new area of the search space, but in the next generation selection pressure does not include this solution in the following child population, and hence these solutions do not appear in columns but are single solutions appearing for a single generation. In multi-modal problems such as the one in Figure 10, columns of solutions form when a subset of the population explores more regions of the search space, and become trapped in local optima (where the objective value is not significant enough to be removed from the population at subsequent generations by current selection pressure). This can be seen as a column of solutions forming, separated from the global optima/main mass of solutions (converging to local optima) and disappearing at subsequent generations (escaping local optima). When solutions are diversifying across the Pareto front, this can also be seen as a column of solutions joining to the Pareto front in the final generations. For the same multimodal problem (Figure 10), we expect many columns of solutions to form in various areas of the search space, where they become trapped for a period of time and break free from local optima. The MDS/LMDS visualisations for these more difficult problems do reflect a more complex search.

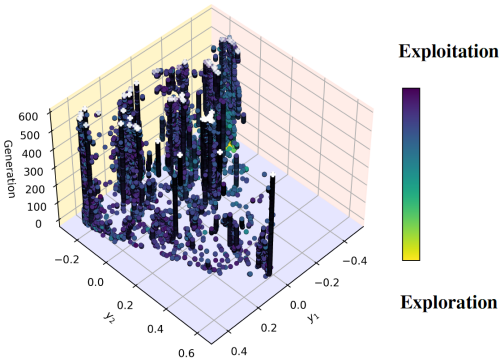


Fig. 10. LMDS reduced objective space of a WFG multi-modal problem in three objectives produced with 5000 landmarks and 50,000 function evaluations. Solutions not yet converged.

E. Many-objective Problems

Considering the many-objective problems (problems with greater than three objectives), we can observe from Figure 11 all the same features as from the multi-objective MDS/LMDS problems described previously. We can see the Pareto front geometry is preserved. A smooth and continuous path of solutions can be seen for the easier unimodal problems such as Figure 12, whilst the more challenging multimodal problems cause the solutions to take a more sporadic path to the optima, indicating the presence of local optima, as well as identifying which solutions encounter difficulty. We can identify at which generations the local optima trap the solutions and the duration of the event, demonstrating how the same mapping principles are carried through to many-objective problems. The same mappings for unimodal problems in the multi- and many-objective DTLZ problems (Figures 12 and 13) subsist.

V. ALGORITHM PERFORMANCE TESTING

This section demonstrates LMDS visualisations to be an effective tool for examining algorithm performance. We have

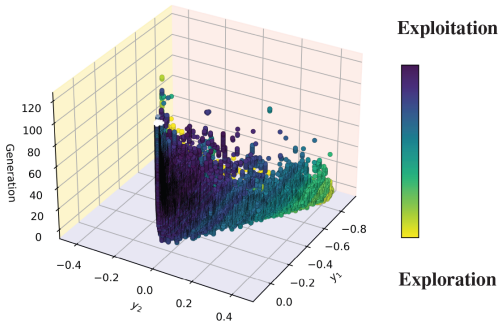


Fig. 11. LMDS reduced objective space of DTLZ1 in five objectives produced with 5000 landmarks and 200,000 function evaluations.

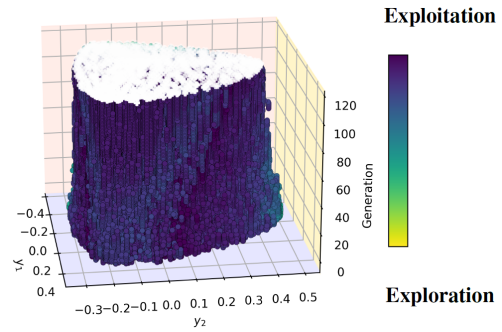


Fig. 12. LMDS reduced objective space of DTLZ2 in five objectives produced with 5000 landmarks and 200,000 function evaluations.

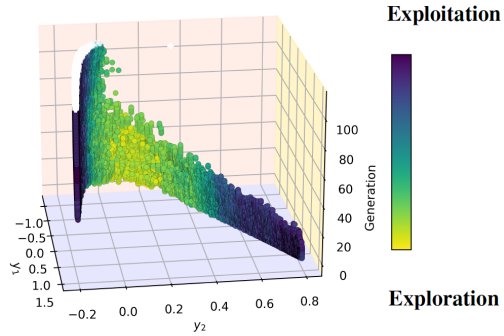


Fig. 13. LMDS reduced objective space of DTLZ7 in five objectives produced with 5000 landmarks and 200,000 function evaluations.

chosen to visualise a 3-objective DTLZ1 problem with 5000 landmarks and 30,000 function evaluations. We have chosen four different algorithms, namely NSGA-II [38], NSGA-III [36], MOEA/D [39] and SPEA2 [40], with different parameters and mutation operators to evaluate and detail some of the comprehension that can be obtained using this visualisation for examining EAs, using the mappings from Section IV.

From Figure 14, we can identify NSGA-III as being the most effective algorithm as it more rapidly converges to the Pareto front. It encounters only a few local optima up to generation 50 - consequently, we could tune more suitable parameters for this algorithm run, although in comparison with the other tested algorithms it is the most effective algorithm with its default parameters. The LMDS forms almost a single column of solutions at generation 50, demonstrating the EA's ability to overcome local optima in short periods of time (and in a shorter time than SPEA2 and NSGA-II). We can also identify a large subset of the population converge along unique paths (these being the reference plane or crowding distance boundaries); this is an artefact of the algorithm selection operator. We can compare the NSGA-III algorithm reference plane operator effect to the effect that crowding distance has on NSGA-II. This level of algorithm examination provides an extra level of insight to merely using the hypervolume or IGD.

NSGA-II is the second most effective algorithm tested. This is again intuitive from the LMDS visualisation. A larger subset of the population encounters difficulty than with the NSGA-III algorithm. However, the algorithm converges to the global optima in fewer generations, and with a smaller population subset becoming trapped in local optima, than SPEA2.

We next consider SPEA2, placing it in a similar category to NSGA-II in terms of algorithm performance. SPEA2 population solutions that become trapped in local optima become trapped in optima further away from the global optima than the positions of the NSGA-II trapped solutions. SPEA2 solutions trapped in local optima become trapped for longer periods of time (more generations) before escaping the optima, as seen in the visualisation. So from the visualisation, the performance of SPEA2 is slightly worse than NSGA-II. MOEA/D solutions show very little exploration as the initial solutions positions are similar to the final solutions, and the colouring metric suggests exploitation dominated the search. This visualisation shows how the algorithm is working by creating sub-problems as a reference for the population to enhance convergence.

We then evaluate NSGA-II with the mutation and crossover parameters $(1, 1)$. We consider these bad parameters for this problem and the resulting visualisation demonstrates this by taking more generations than NSGA-II with the standard parameters to converge and containing more solutions converging away from the global optima. Some solutions form much further away from the global optima (again, represented by the main column of solutions) in comparison with NSGA-II (with standard parameters) and NSGA-III.

Finally, we produced the visualisation of NSGA-III with a mutation operator that discards a current solution and samples a new individual randomly. We would expect this algorithm, with this mutation operator, to be the least effective algorithm than all other six algorithm and parameter combinations that we considered. The visualisation shows poor convergence and reflects the expected poor performing result.

VI. Conclusion

In this work, we have proposed the use of LMDS for visualising EA populations, demonstrating how this may produce intuitive visualisations for examining algorithm performance. The visualisation indirectly maps problem features and directly maps algorithm performance. Whilst we have identified some mappings of problem features to the visualisation, further work will investigate other problems, including discrete and real-world problems to find additional mappings. Further work could also consider an investigation into whether other methods in the Nyström algorithms class can produce faster and more accurate results than LMDS.

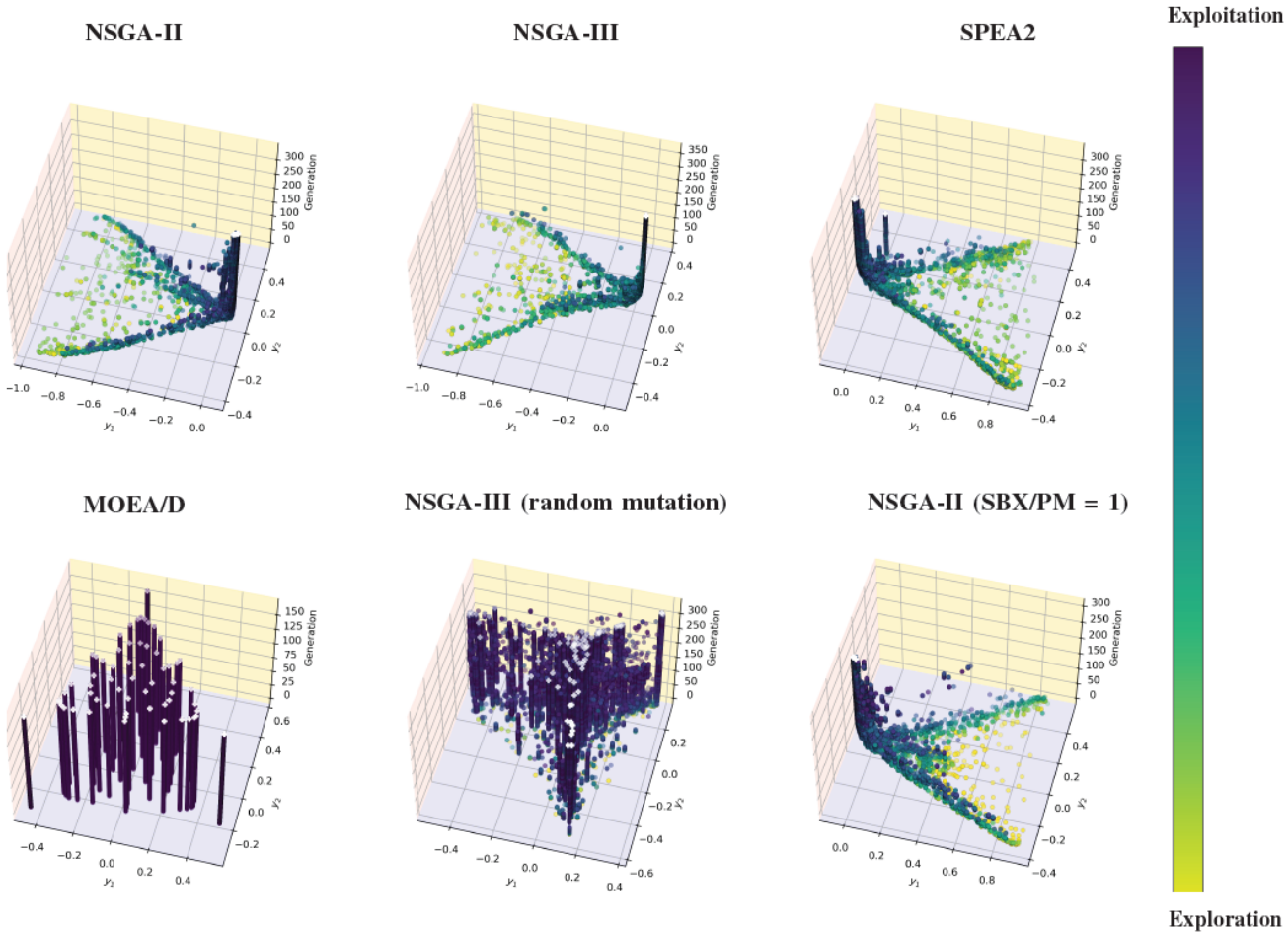


Fig. 14. LMDS reduced objective space of DTLZ1 in three objectives. Top left algorithm is NSGA-II, with hypervolume 0.8202. Top middle algorithm is NSGA-III, with hypervolume 0.8171. Top right algorithm is SPEA2, with hypervolume 0.7117. The bottom left algorithm is MOEA/D, with hypervolume 0.7541. The bottom middle algorithm is NSGA-III with a 'ruin and recreate' mutation operator and hypervolume 0. The bottom right algorithm is NSGA-II with mutation (PM) and crossover (SBX) parameters of 1, and hypervolume 0.6748.

From understanding the mappings identified in Section IV-B, we suggest LMDS visualisations can be used to visually rank and improve algorithm performance in an online environment. LMDS visualisations show a wealth of information; showing solution distributions, one can observe how an algorithm is performing, when and where the population is exploring/exploiting and which areas of the search space the population is perturbing through. Additionally, LMDS can visualise the locations the EA solutions encounter local optima and some features of the problem. We can visualise the non-dominating solutions and the 'journey' the solutions took to get there. Ultimately, LMDS can be an effective tool for assisting the DM in recommending algorithms, their parameters and operators for creating and testing better EAs to solve multi- and many-objective problems.

REFERENCES

[1] A. De Lorenzo, E. Medvet, T. Tušar, and A. Bartoli, "An analysis of dimensionality reduction techniques for visualizing evolution," in *Proceedings of GECCO 2019*, 2019, pp. 1864–1872.

[2] M. J. Walter, D. J. Walker, and M. J. Craven, "Visualising evolution history in multi-and many-objective optimisation," in *International Conference on Parallel Problem Solving from Nature*. Springer, 2020, pp. 299–312.

[3] W. S. Torgerson, "Multidimensional scaling: I. theory and method," *Psychometrika*, vol. 17, no. 4, pp. 401–419, 1952.

[4] H. Gao, H. Nie, and K. Li, "Visualisation of Pareto front approximation: A short survey and empirical comparisons," in *2019 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2019, pp. 1750–1757.

[5] B. Filipič and T. Tušar, "A taxonomy of methods for visualizing Pareto front approximations," in *Proceedings of the Genetic and Evolutionary Computation Conference*, 2018, pp. 649–656.

[6] W. S. Cleveland and R. McGill, "The many faces of a scatterplot," *Journal of the American Statistical Association*, vol. 79, no. 388, pp. 807–822, 1984.

[7] A. Inselberg and B. Dimsdale, "Parallel coordinates: a tool for visualizing multi-dimensional geometry," in *Proceedings of the First IEEE Conference on Visualization: Visualization90*. IEEE, 1990, pp. 361–378.

[8] C. M. Fonseca and P. J. Fleming, "Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization," in *ICGA 93*, vol. 93, no. July. Morgan Kaufmann, 1993, pp. 416–423.

[9] P. Hoffman, G. Grinstein, K. Marx, I. Grosse, and E. Stanley, "DNA visual and analytic data mining," in *Proceedings. Visualization'97 (Cat. No. 97CB36155)*. IEEE, 1997, pp. 437–441.

[10] P. Chiu and C. L. Bloebaum, "Hyper-radial visualization (HRV) method with range-based preferences for multi-objective decision making," *Structural and Multidisciplinary Optimization*, vol. 40, no. 1-6, p. 97, 2010.

[11] A. Ibrahim, S. Rahnamayan, M. V. Martin, and K. Deb, "3D-RadVis: Visualization of Pareto front in many-objective optimization," in *2016 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2016, pp. 736–745.

[12] H. Pohlheim, "Visualization of evolutionary algorithms-set of standard techniques and multidimensional visualization," in *Proceedings of the Genetic and Evolutionary Computation Conference*, vol. 1. San Francisco, CA., 1999, pp. 533–540.
[13] M. Ashby, "Multi-objective optimization in material design and selection," *Acta materialia*, vol. 48, no. 1, pp. 359–369, 2000.
[14] S. Obayashi and D. Sasaki, "Visualization and data mining of Pareto solutions using self-organizing map," in *International Conference on Evolutionary Multi-Criterion Optimization*. Springer, 2003, pp. 796–809.
[15] K. H. Ang, G. Chong, and Y. Li, "Visualization technique for analyzing non-dominated set comparison," in *Proceedings of the 4th Asia-Pacific Conference on Simulated Evolution and Learning (SEAL'02)*, vol. 1, 2002, p. 36.
[16] J. B. Tenenbaum, V. De Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *science*, vol. 290, no. 5500, pp. 2319–2323, 2000.
[17] D. Lowe and M. E. Tipping, "Neuroscale: Novel topographic feature extraction using RBF networks," in *Advances in neural information processing systems*, 1997, pp. 543–549.
[18] R. M. Everson and J. E. Fieldsend, "Multi-class ROC analysis from a multi-objective optimisation perspective," *Pattern Recognition Letters*, vol. 27, no. 8, pp. 918–927, 2006.
[19] J. W. Sammon, "A nonlinear mapping for data structure analysis," *IEEE Transactions on computers*, vol. 100, no. 5, pp. 401–409, 1969.
[20] J. J. Valdés and A. J. Barton, "Visualizing high dimensional objective spaces for multi-objective optimization: A virtual reality approach," in *2007 IEEE Congress on Evolutionary Computation*. IEEE, 2007, pp. 4199–4206.
[21] X. Blasco, J. M. Herrero, J. Sanchis, and M. Martínez, "A new graphical visualization of n-dimensional Pareto front for decision-making in multiobjective optimization," *Information Sciences*, vol. 178, no. 20, pp. 3908–3924, 2008.
[22] T. Tušar and B. Filipič, "Visualization of Pareto front approximations in evolutionary multiobjective optimization: A critical review and the prosection method," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 2, pp. 225–245, 2014.
[23] Z. He and G. G. Yen, "Visualization and performance metric in many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 3, pp. 386–402, 2015.
[24] A. K. A. Talukder and K. Deb, "Paletteviz: A visualization method for functional understanding of high-dimensional pareto-optimal data-sets to aid multi-criteria decision making," *IEEE Computational Intelligence Magazine*, vol. 15, no. 2, pp. 36–48, 2020.
[25] D. J. Walker and M. J. Craven, "Toward the online visualisation of algorithm performance for parameter selection," in *International Conference on the Applications of Evolutionary Computation*. Springer, 2018, pp. 547–560.
[26] D. Walker and M. J. Craven, "Visualising the operation of evolutionary algorithms optimising water distribution network design problems," in *13th International Conference on Hydroinformatics*, 2018.
[27] M. J. Craven and H. C. Jimbo, "EA stability visualization: perturbations, metrics and performance," in *Proceedings of the Companion Publication of GECCO 2014*, 2014, pp. 1083–1090.
[28] B. Chakuma and M. Helbig, "Visualizing the optimization process for multi-objective optimization problems," in *International Conference on Artificial Intelligence and Soft Computing*. Springer, 2018, pp. 333–344.
[29] L. Schäpermeier, C. Grimme, and P. Kerschke, "One PLOT to show them all: Visualization of efficient sets in multi-objective landscapes," in *International Conference on Parallel Problem Solving from Nature*. Springer, 2020, pp. 154–167.
[30] Y. Masafumi, Y. Tomohiro, and F. Takeshi, "Study on effect of MOGA with interactive island model using visualization," in *IEEE Congress on Evolutionary Computation*. IEEE, 2010, pp. 1–6.
[31] D. J. Walker, R. M. Everson, and J. E. Fieldsend, "Visualizing mutually nondominating solution sets in many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 17, no. 2, pp. 165–184, 2012.
[32] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable multi-objective optimization test problems," in *Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No. 02TH8600)*, vol. 1. IEEE, 2002, pp. 825–830.
[33] G. H. Golub and C. F. Van Loan, "Matrix computations," *Johns Hopkins Universtiy Press, 3rd edtion*, 1996.
[34] V. De Silva and J. B. Tenenbaum, "Sparse multidimensional scaling using landmark points," Stanford University, Tech. Rep., 2004.
[35] M. Črepinšek, S.-H. Liu, and M. Mernik, "Exploration and exploitation in evolutionary algorithms: A survey," *ACM computing surveys (CSUR)*, vol. 45, no. 3, pp. 1–33, 2013.
[36] K. Deb and H. Jain, "An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: solving problems with box constraints," *IEEE transactions on evolutionary computation*, vol. 18, no. 4, pp. 577–601, 2013.
[37] S. Huband, P. Hingston, L. Barone, and L. While, "A review of multiobjective test problems and a scalable test problem toolkit," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 5, pp. 477–506, 2006.
[38] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE transactions on evolutionary computation*, vol. 6, no. 2, pp. 182–197, 2002.
[39] Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE Transactions on evolutionary computation*, vol. 11, no. 6, pp. 712–731, 2007.
[40] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength pareto evolutionary algorithm," *TIK-report*, vol. 103, 2001.

Mathew Walter received a degree in Mathematics with Finance from the University of Plymouth in 2019, and is working toward a PhD in Computing. He is currently a Member of the Plymouth Optimisation Group, the Centre for Robotics and Neural Systems (CRNS), and the Big Data Group. His current research interests include evolutionary computation, optimisation, visualisation, machine learning and novel applications of evolutionary algorithms.

David Walker is a Lecturer in Computer Science at the University of Plymouth. He has a PhD in Computer Science from the University of Exeter, where he studied the visualisation of the mutually non-dominating sets generated by many-objective evolutionary algorithms. His work focuses on human-in-the-loop nature-inspired computing through visualisation and interactive optimisation, as well as the development of novel hyper-heuristics. He joined the University of Plymouth in 2018, where he is co-lead of the Plymouth Optimisation Group, and a member of the Centre for Robotic and Neural Systems.

Matthew Craven is a Lecturer in Applied Mathematics at the University of Plymouth. His research interests and publications are interdisciplinary and centred around AI, including Visualisation, Mathematical Modelling and Cryptography. He is a member of the IMA and LMS, and reviews for international journals such as the Journal of the Operational Research Society and Applied Soft Computing. He co-leads the Big Data Research Group at Plymouth, is co-lead of the Plymouth Optimisation Group and is a member of the Centre for Mathematical Sciences.