Image-based benchmarking and visualization for large-scale global optimization

Kyle Robert Harrison¹,² · Azam Asilian Bidgoli¹ · Shahryar Rahnamayan¹ · Kalyanmoy Deb³

Abstract
In the context of optimization, visualization techniques can be useful for understanding the behaviour of optimization algorithms and can even provide a means to facilitate human interaction with an optimizer. Towards this goal, an image-based visualization framework, without dimension reduction, that visualizes the solutions to large-scale global optimization problems as images is proposed. In the proposed framework, the pixels visualize decision variables while the entire image represents the overall solution quality. This framework affords a number of benefits over existing visualization techniques including enhanced scalability (in terms of the number of decision variables), facilitation of standard image processing techniques, providing nearly infinite benchmark cases, and explicit alignment with human perception. To the best of the authors’ knowledge, this is the first realization of a dimension-preserving, scalable visualization framework that embeds the inherent relationship between decision space and objective space. The proposed framework is utilized with different mapping schemes on an image-reconstruction problem that encompass continuous, discrete, constrained, dynamic, and multi-objective optimization. The proposed framework is then demonstrated on arbitrary benchmark problems with known optima. Experimental results elucidate the flexibility and demonstrate how valuable information about the search process can be gathered via the proposed visualization framework. Results of a user survey strongly support that users perceive a correlation between objective fitness values and the quality of the corresponding images generated by the proposed framework.

Keywords Visualization · Optimization · Large-scale · High-dimensional · Dimensionality-preserving · Image processing

1 Introduction
Real-world optimization problems involve a relatively large number of problem dimensions. However, visualizing solutions with more than three dimensions is problematic and thus extracting useful information is complex. Specifically, high-dimensional data visualization often suffers from the “curse of dimensionality” such that many existing visualization techniques do not scale well with respect to problem dimension. Nonetheless, visualization of high-dimensional data comes with a number of benefits. Firstly, visualization allows for the presentation of data in a much more concise and comprehensible manner. This can allow key patterns to be highlighted, which may otherwise be left unnoticed. Similarly, visualization provides a means for an interactive pipeline, whereby the decision maker can be made aware of underlying characteristics in the data, or the optimization process, and react accordingly. Furthermore, with regards to optimization, visualization of optimization results can give a clearer understanding of the effects of operational decisions and their results.

Visualization techniques generally fall within one of two broad categories, namely those with dimensionality reduction and those without dimensionality reduction. Generally, the techniques that use dimension reduction provide a 2- or 3-dimensional projection of the data with the inherent
goal of preserving spatial distance. While such techniques scale well with regards to the number of problem dimensions, they suffer from the explicit drawback that they do not preserve dimensionality and, thus, the inferences that can be made from them are limited. Furthermore, the insights that can be made from the resulting projection are inherently tied to the quality of the projection; the projection may not convey complex patterns that are present in the data set. In contrast, visualization techniques that do not use dimension reduction are able to visualize problems with arbitrary dimension, in theory, without the use of projection. However, in practice, many of these techniques are limited in their scalability as the visualizations become infeasible when faced with problems that have a very large number of dimensions.

1.1 Dimensionality reduction techniques

One of the simplest visualization techniques for high-dimensional data is a matrix of scatterplots, whereby each scatterplot independently visualizes each of the \( \binom{n}{2} \), or equivalently \( \frac{n^2 - 2}{2} \), distinct pairs of problem dimensions. However, this technique does not scale well in terms of problem dimensions as the space required to visualize all pairs exhibits quadratic growth. While it can be argued that a scatterplot matrix preserves dimensionality, as it provides a visual representation of all \( n \) dimensions, it does so in a pairwise manner whereby only two dimensions are visualized simultaneously. Therefore, the scatterplot matrix is, in the context of this work, considered as a dimensionality-reducing technique. An example scatterplot matrix is shown in Fig. 1.

A self-organizing map (SOM) [21] is a type of artificial neural network that provides a mapping from \( n \)-dimensional vectors to \( m \)-dimensional vectors, where \( m < n \) and commonly \( m = 2 \). A SOM is an unsupervised approach in which the mapping preserves the topological characteristics of the original high-dimensional space. Typically, the output of a SOM can then be used to analyze clusters formed from high-dimensional data [27]. An example of the output from a SOM is given in Fig. 2.

Multidimensional scaling (MDS) [31] refers to a family of data analysis techniques that provide a mapping \( n \)-dimensional data into \( m \)-dimensional points through nonlinear, distance preserving ordination. In the case where \( m < 3 \), MDS can then be used to visualize the spatial relationship among the \( n \)-dimensional points. However, one limitation of MDS is that it does not capture global structure effectively [23]. An example of MDS is shown in Fig. 3.

t-distributed stochastic neighbour embedding (t-SNE) [23] provides a similar mapping capability as MDS, but with an enhanced ability to visualize global structure. t-SNE first defines a conditional probability distribution associated with each pair of \( n \)-dimensional points being selected as neighbours, according to their distance. A similar probability distribution is then formulated for the \( m \)-dimensional projection by minimizing the relative entropy between the two distributions. An example of t-SNE is shown in Fig. 4.

Graph Neural Network Visualization (GNNVis) [12] employs the use of graph-neural networks (GNNs) to learn a dimension-reducing embedding for large-scale, high-dimensional data. First, a GNN is trained with the objective of preserving a similarity metric between the high-dimensional data points when mapped to a 2- or 3-dimensional space. The core feature of GNNVis is the fusion of the features from neighbouring points through the GNN. Subsequently, the learned embedding is used to visualize unseen data points. It is reported that the GNNVis technique can be extended to use any objective function, and thus is a generalizable framework. Note that, GNNVis is similar in principle to t-SNE, but uses a learned embedding to perform the dimension reduction rather than a fixed embedding process.

Radial coordinates visualization (RadVis) [11] provides a nonlinear mapping of \( n \)-dimensional data to two dimensions using the physics of springs. The mapping for each data point is produced by attaching a “spring” to each of the \( n \) equidistant axes arranged on the perimeter of a circle. For each of the visualized dimensions, the spring constant is given by a coefficient that is proportional to the data value in the corresponding dimension. The springs are then allowed to move freely (in two dimensions) until they have reached an equilibrium point. The resulting location provides the two-dimensional location of the original data point. This process is repeated for each data point. Similar to parallel coordinates, RadVis is implicitly limited in terms of scalability as it requires a circle to be divided into \( n \) segments. An example of RadVis is shown in Fig. 5. Recently, RadVis was extended to facilitate mapping to a 3-dimensional space, whereby the third dimension can be used to visualize a quality metric [14].

While not a standalone visualization technique in its own right, a recent study proposed using a sequence of visualizations, each corresponding to the population in one generation, attained via any arbitrary dimensionality reduction technique as a means to ascertain population dynamics over time in evolutionary algorithms [4].

Another, related area of research is on the visualization of optimization results for many-objective optimization, most of which are inherently dimensionality reduction techniques [10, 25, 26].

While the aforementioned visualization techniques are commonly used, they all exhibit dimensionality reduction.
In contrast, there are a number of techniques that are capable of visualizing $n$-dimensional data directly.

### 1.2 Dimensionality preserving techniques

The simplest visualization technique that preserves dimensionality is arguably the star plot, which visualizes each dimension of a candidate solution along a radial axes. The length of the “star” along each axis corresponds to the value in that particular dimension. While the star plot does not reduce the number of dimensions for visualizations, it becomes unwieldy as the number of problem dimensions increases. An example of star plots are given in Fig. 6.

Similar to the star plot, the segment plot segments a circle into equally-spaced wedges, one for each problem dimension, with the relative size of each wedge corresponding to the value in that dimension\(^1\). However, the segment plot also suffers from the same limitation as the star plot – it is inherently limited by the number of divisions of a circle that can be meaningfully perceived. Examples of segment plots are given in Fig. 7.

Another commonly used, dimension-preserving technique is parallel coordinates [15], which visualizes $n$-dimensional data by aligning each of the coordinate axes in parallel, rather than orthogonal. This allows for arbitrary dimensions to be visualized as a series of line segments between successive axes. An example of parallel coordinates in given in Fig. 8. While parallel coordinates are capable of visualizing high-dimensional data, they suffer from an implicit limit on their scalability due to their linear construction. Specifically, visualizing $n$-dimensional data

\(^1\)Not to be confused with the pie chart, where the relative width of the wedges indicates a percentage.
Fig. 3  Multidimensional scaling on the Iris dataset

Fig. 4  t-SNE on the Iris dataset

Fig. 5  RadVis on the Iris dataset
requires \( n \) parallel coordinates, which becomes infeasible as \( n \) grows large. Furthermore, an additional consideration when employing parallel coordinates is the ordering of the axes, which can lead to drastically different observations being highlighted [16]. Recently, the parallel coordinates methodology was extended to include information about solution distribution and convergence in a technique referred to as parallel peaks. Cheng et al. [3]

A more recent visualization approach, designed specifically to visualize genotypes in an evolutionary algorithm, is the diversity and usage (DU) maps [24]. DU maps visualize both the diversity and utility of each problem dimension as a single point in a 2D matrix, where the rows correspond to problem dimensions and the columns correspond to the search iteration. Two functions, which quantify the diversity and usage, must be supplied by the user and are used to select a colour for each dimension of the problem in each iteration of the search process. A diversity function, which produces a value in the range [0,1], quantifies the diversity of a particular problem dimension. The utility function, also with a range of [0,1], measures the degree to which a particular dimension contributes to the overall solution. Note that, for both functions, different implementations can be used for each problem dimension. While the DU map can be used to gather population-level insight about the performance of an evolutionary search algorithm, it does not produce visualizations of individual candidate solutions.

Another visualization framework that is of particular relevance, but does not pioneer any particular visualization method, is VISPLORE [20]. VISPLORE is a toolkit for visualizing information about the search process of particle swarm optimization at various levels of granularity and includes both dimension-preserving and dimension-reducing visualizations. Most relevant to this study is the visualization of individual particles using density heatmaps, parallel coordinate plots, star plots, and history plots, which correspond to a hierarchical view of information flow between iterations. The particular strength of VISPLORE is that it packages many different visualizations into a coherent, usable toolkit.

The visualization techniques that are most relevant to the context of this paper are pixel-based approaches, whereby decision variables are visualized as coloured pixels. However, the work in this domain is limited and...
often context-specific. Furthermore, these approaches are pixel-oriented and do not necessarily generate a coherent image. Note that, implementations of these techniques are not readily available and, thus, examples are not provided for these techniques.

One example of a pixel-oriented approach is VisDB [17], which provides a pixel-based visualization that maps database items to pixels based on their respective similarity to the terms specified in a query. While VisDB was proposed as the result of a preliminary study on using pixel-oriented
approaches to visualize high-dimensional data, it is inherently tied to visualizing query-related data and thus is not suitable for optimization-oriented visualization tasks.

The recursive pattern visualization [18] provides a generalized recursive process for arranging pixels as means of visualizing high-dimensional sets of data. However, this framework does not actually provide a mapping from the data to pixels. Rather, it provides a generalized process for handling the layout of the pixels such that implicit relationships can be visualized effectively. One inherent issue with this process is that it rearranges the data and thus, in an optimization context, would alter the order of the decision variables in the resulting visualization.

The circle segments technique [1] partitions a circle into $n$ segments, each of which represent a single dimension. Within each segment, data attributes are visualized as a single, coloured pixel. However, this approach also suffers from a similar issue to that of RadVis, where the scalability is inherently related to the segmentation of a circle.

### 1.3 Summary of existing techniques

Table 1 provides a brief summary of the aforementioned visualization techniques, categorized into two groups, namely those with dimensionality reduction and those without dimensionality reduction. Table 1 also provides, for each visualization technique, the number of dimensions used for visualization and a relative ranking in terms of: (1) scalability in the number of data dimensions; (2) scalability in terms of the number of entities; and (3) their preservation of relationships in the data. Note that, the rankings in Table 1 are subjective and correspond to the authors’ opinion on the scalability capabilities of each technique.

It is clear from Table 1 that both dimension-reducing and dimension-preserving visualizations suffer from their own drawbacks. Techniques with dimension reduction suffer from limitations in their visualization power due to the mapping onto lower-dimensional spaces, while dimensionality-preserving techniques suffer from implicit limitations in their scalability with respect to the dimensionality. While the dimensionality-preserving, pixel-based approaches address the dimensional scalability limitations, they suffer from a more nuanced issue – namely that their visualization power is inherently related to the layout of the pixels. Clearly, there is a lack of visualization techniques that strongly address both the dimensionality scaling and relationship preservation aspects of visualization.

### 1.4 Major contributions

To address a key limitation imposed by existing visualization techniques, namely the ability to generate a dimension-preserving visualization for arbitrary solutions to large-scale global optimization problems, this paper proposes an image-based visualization framework. The general goal of the framework is to facilitate the visualization of single solutions to arbitrary problems using images whereby higher quality images represent solutions with better fitness values. For demonstration purposes, a concrete problem type, specifically image reproduction, is considered to elucidate the capabilities of the proposed image-based visualization framework. The use of images as a visualization mechanism introduces a higher-level of visualization capabilities in comparison to pixel-based approaches and allows for semantic embedding of solution quality using standard image-processing techniques. For example, the image itself can represent the overall quality of the solution while the pixels can be used to visualize the optimization variables. To the best of the authors’ knowledge, this is the first time that the inherent relationship between decision space and objective space is realized in a dimensionality-preserving, scalable visualization technique. However, it should be noted that scalability, in our proposed framework, refers to scalability in terms of the number of problem dimensions, not necessarily the number of entities that can be visualized.

The proposed framework is exemplified on a wide variety of optimization problem types in the context of an image reconstruction problem. Furthermore, the usage of the proposed framework on arbitrary benchmark problems, with known optima, is also demonstrated. A user survey is conducted to validate the efficacy of the proposed approach.

### 1.5 Outline

The remainder of this paper is structured as follows. Section 2 provides a brief description of the optimization techniques used in this study. Section 3 introduces the proposed image-based visualization framework and Section 4 describes the experimental design. Section 5 exemplifies the flexibility of image-based visualization via a number of different mapping schemes applied to a wide variety of large-scale global optimization problems. Section 6 then examines the use of image-based visualization on arbitrary benchmark problems with known optima. Section 7 presents the results of a user survey conducted to validate the proposed approach. Finally, Section 8 provides concluding remarks and avenues of future work.

### 2 Background

This section provides a brief introduction to the optimization techniques used in this study. It should be noted that these algorithms are used as a proof-of-concept, rather than to provide an empirical comparison of their performance.
Table 1 Summary of existing visualization techniques

| Type      | Technique              | Dim. | Scal. (Dim.) | Scal. (Ent.) | Rel. Pres. |
|-----------|------------------------|------|--------------|--------------|------------|
| DR        | Scatterplot Matrix     | 2, n | **           | ***          | *          |
| SOM [21]  | ≤ 3                    | ***  | ***          | **           |            |
| MDS [31]  | ≤ 3                    | ***  | ***          | *            |            |
| t-SNE [23]| ≤ 3                    | ***  | ***          | **           |            |
| GNNVis [12]| ≤ 3                  | ***  | ***          | **           |            |
| RadVis [11]| 2                    | ***  | ***          | *            |            |
| 3D RadVis [14]| 3        | ***  | ***          | **           |            |
| DP        | Star Plot              | n    | **           | *            | ***        |
|           | Segment Plot           | n    | **           | *            | ***        |
|           | Parallel Coordinates [15]| n    | **           | *            | ***        |
|           | DU Maps [24]           | n    | **           | **           |           |
|           | VisDB [17]             | n    | **           | **           | *          |
|           | Recursive Pattern [18]  | n    | N/A          | N/A          | N/A        |
|           | Circle Segments [1]    | n    | **           | ***          | **         |
|           | Proposed               | n    | ***          | *            | ***        |

Various characteristics are (subjectively) ranked, where *** indicates the best ranking. Legend: DR – dimensionality reduction, DP – dimensionality preserving, Dim. – dimensionality, Scal. (Dim.) – scalability with respect to dimensionality, Scal. (Ent.) – scalability with respect to the number of entities, Rel. Pres. – relationship preservation.

2.1 Particle swarm optimization (PSO)

PSO [19] is a meta-heuristic technique, inspired by the flocking behaviour of birds, whereby a collection of candidate solutions, referred to as particles, are iteratively improved through a velocity-based movement process. The particles in the PSO algorithm are semi-autonomous and each retain three pieces of information, namely their current position in the search space, their current velocity, and the best position they have found within the search space. Particle positions are then updated, each iteration, by recalculating the velocity vector and performing a movement operation accordingly. The variant of PSO employed in this study is the inertia weight model [29].

Creation of the mutant vector \( v \) for an individual \( x \) in dimension \( i \) using the DE/rand/1/bin strategy [28] is given by

\[
    v_i = x_{r1,i} + F(x_{r2,i} - x_{r3,i})
\]

where \( r1 \neq r2 \neq r3 \) are three randomly selected indices corresponding to distinct members of the population and \( F \in [0, 2] \) is the user-supplied differential weight.

A uniform crossover operator is then used to recombine the mutant vector \( v \) with the current position \( x \) in dimension \( i \), as follows

\[
    u_i = \begin{cases} 
        v_i & \text{if } \text{rand()} < CR \text{ or } i = \text{randi}(D) \\ 
        x_i & \text{otherwise} 
    \end{cases}
\]

where \( \text{rand()} \sim U(0, 1) \), \( \text{randi}(D) \) selects a uniform random integer in the range \([1, D]\), \( D \) is the problem dimensionality, and \( CR \in [0, 1] \) is the user-supplied crossover probability. If the fitness of vector \( u \) is superior to that of \( x \), the individual’s position is updated to \( u \). Otherwise, the individual’s position remains unchanged.

2.2 Differential evolution (DE)

DE [30] is an evolutionary optimization algorithm that iteratively improves a population of candidate solutions, referred to as individuals. Individuals within the population, initially placed at random positions in the feasible search space, are updated using mutation, crossover, and selection operations. In the DE algorithm, mutation and crossover are used to create a trial vector for each individual; a trial vector represents a potential new position for the individual. If the generated trial vector improves the individual’s fitness, the trial vector is accepted and the individual’s position is updated accordingly. Otherwise, the trial vector is discarded and the individual retains its current position.

2.3 Third generalized differential evolution (GDE3)

As will be shown in Section 5, the proposed framework can also be applied in a multi-objective context. Therefore, a multi-objective variant of DE is employed to demonstrate this capability. The third version of Generalized Differential Evolution (GDE3) [22] extends the DE algorithm for
multi-objective optimization problems. The selection mechanism for GDE3 is inspired by the constraint-domination mechanism and a measure of crowding distance.

To perform selection in the GDE3 algorithm, a new trial vector is accepted if the original and trial vectors are both infeasible, or the trial vector dominates, with respect to the problem constraints, the original vector. If only one of the original or trial vectors is feasible, the feasible vector is always selected. If both vectors are feasible, then the vector which dominates the other is selected. Finally, if the vectors are non-dominated with respect to each other, both vectors are selected. As a result, the population size may increase, at which point the population will be pruned using a selection strategy similar to that of NSGA-II [6]. First, the individuals in the population are sorted using non-dominated sorting, then according to the crowding distance measure. A specified number of individuals are then selected to formulate the next generation.

### 3 Image-based visualization for large-scale global optimization

A primary motivation for image-based visualization is the inherent scalability with respect to problem dimensionality; it is not unreasonable to have an image with 1,000,000 pixels (i.e., 1 megapixel). In fact, an image of this size would be generally considered to have low resolution. However, many existing visualization techniques, which do not use dimensionality reduction, cannot reasonably visualize dimensions of this magnitude. For example, parallel coordinates would require 1,000,000 parallel axes while RadVis would require dividing the circumference of a circle into 1,000,000 segments. Clearly, even existing dimensionality preserving techniques degrade rapidly with increasing dimensionality. In contrast, visualization of 1,000,000 dimensions using image-based visualization requires only an image with 1,000,000 pixels (e.g., a size of 1000x1000 or 2000x500). Furthermore, the introduction of colour and other image features permits even greater scalability and visualization capabilities. Thus, the problematic “curse of dimensionality”, with respect to visualization, is effectively mitigated by this framework. However, it should be noted that this technique is best equipped to address problems with a composite dimensionality, i.e., those with dimension $D = w \times h, \ w, h > 1$. Nonetheless, by omitting some pixels from the target image (i.e., having an incomplete row/column in the image), this technique can be employed on problems with arbitrary dimensionality. One primary drawback of the framework is that, while it preserves the dimensionality of the problem, identification of the pixel corresponding to a particular decision variable is challenging, though still possible if the user specifies the decision variable(s) of interest. However, in the context of large-scale visualization, with potentially thousands or millions of decision variables, identification of particular variables/pixels would pose a similar challenge for any visualization technique.

In addition to the scalability, image-based visualization affords many other advantages over existing visualization techniques. Notably, one of the fundamental properties that the visualization function $F$ must exhibit is that it must align with human perception. That is, a non-expert practitioner should be able to reasonably perceive the visualization of a superior solution as being a higher-quality image. Image-based visualization can also be trivially extended to produce visualizations of the optimization process over time by encoding the resulting images as frames in a video.

### 3.1 Formal definition

Given an arbitrary optimization problem $\pi$ and candidate solution $s$, the goal of image-based visualization is to devise a function, $F$, which produces an image, $I$, that is representative of the relative quality of $s$ with respect to $\pi$. More concretely, the general goal is to devise a function given by

$$F(\pi, s) \mapsto I. \tag{3}$$

In the case where an optimal solution, $s^\ast$, is known a priori, the objective can be further refined such that the goal is to devise a function given by

$$F(\pi, s, s^\ast) \mapsto I, \tag{4}$$

which produces an image that is representative of the absolute quality of $s$ in comparison to the optimal solution $s^\ast$. Furthermore, $F$ also induces a mapping from decision-space to individual pixels, such that both decision space variables and the overall objective quality of $s$ is embedded within the resulting image, $I$. An example demonstrating how the decision variables are used to formulate an image is provided in Fig. 9. Effectively, a (linear) candidate solution is reformulated as a matrix and the decision variables are then mapped to the corresponding pixels.

Given that the core of image-based visualization lies at the intersection of optimization and image-processing, there are a number of additional benefits that arise from this framework. For example, image-based visualization facilitates the use of image-processing metrics as objective functions to construct new optimization problems. Similarly, it allows for the plethora of readily-available images to be used as benchmark cases. In addition, the flexibility of image-processing capabilities facilitates the use of this framework on many different optimization problem types. Thus, the benefits of image-based visualization are numerous.
3.2 Mapping function for an image reconstruction problem

In this section, a simple mapping function is presented whereby the optimization objective was to replicate a target image according to some image similarity metric. Thus, the objective fitness function corresponds to the minimization of an image comparison metric that would quantify the difference between a candidate solution and a target image. This problem formulation was chosen as it permits a mapping function that is a direct encoding the candidate solution, \( s \), as the corresponding pixel values, formally defined by

\[
F(\pi, s) \mapsto A \in \mathbb{R}^{h \times w} \mid A_{ij} = s[j \times h + i],
\]

where \( A \) is a matrix of pixel greyscale intensity values with height \( h \) and width \( w \), taken as the dimensions of the target image such that \( D = h \times w \). Thus, the pixel values directly visualize the decision variables of the problem while the resulting images visualize the overall solution quality. As such, the complexity of this mapping function is based solely on the dimension of the problem (i.e., \( O(D) \)). Therefore, the use of this mapping function does not pose a significant overhead to the optimizer. However, the overall complexity overhead is dependent on how often the mapping function is employed. For example, employing this mapping on every solution at every iteration would incur significantly more overhead than employing it only on the best known solution at every \( k \)th iteration.

While the mapping function given in (5) is only relevant to a specific, direct application of the proposed visualization framework, more complex mapping functions that handle arbitrary optimization problems are certainly possible. For example, the next section describes a mapping function that is suitable for an arbitrary objective function, with a single known optimum.

3.3 Mapping function for an arbitrary objective function with a known optimum

In this section, an example of a more complex mapping function, applicable to a much wider variety of optimization problems, is presented. This mapping function can be applied to any real-valued, boundary-constrained problem with a single, known optimum. The general concept is similar to that of (5) in that each decision variable will correspond to a single pixel in the output image. However, the decision variable itself no longer directly encodes the pixel’s colour. Rather, the output image will be constructed by normalizing and linearly scaling the error in each dimension of the candidate solution. Thus, decision variables that are less than the value of the optimal solution will be “darkened” (i.e., have a greyscale value closer to 0), while variables that are greater than the value of the optimal solution will be “lightened” (i.e., have a greyscale value closer to 1) as follows

\[
F(\pi, s) \mapsto A \in \mathbb{R}^{h \times w} \mid A_{ij} = \begin{cases} 
T_{ij} + \frac{e_{ij}}{|o_{ij} - T_{ij}|}(1 - T_{ij}) & \text{if } e_{ij} > 0 \\
T_{ij} - \frac{e_{ij}}{|o_{ij} - T_{ij}|}(T_{ij}) & \text{if } e_{ij} < 0 \\
T_{ij} & \text{if } e_{ij} = 0
\end{cases}
\]

where \( T \) is the target image, \( s \) is the candidate solution, \( o \) is the optimal solution, \( e = s - o \) is the error between the candidate solution and the optimal solution, and \( l \) and \( u \) refer to the lower and upper bounds of the problem, respectively. As in (5), the index \( ij \) in the image matrix refers to dimension \( j \times h + i \) in the candidate solution. As with the previous mapping function, the calculation of the pixel values is constant and thus the complexity added through the use of this mapping function is \( O(D) \) for each solution on which it is employed.

Effectively, this mapping scheme linearly scales the error in each dimension (i.e., pixel), based on the distance between the candidate solution and the optimal solution, using the greyscale value of the target pixel as the reference and the problem bounds as the extremes. Note that, the (positive) error term is maximized, for a particular dimension, when the candidate solution lies on the upper boundary of the problem in that dimension – this corresponds to a white pixel. Conversely, the error term for a particular dimension is minimized when the candidate solution lies on the lower boundary of the problem in that dimension – this corresponds to a black pixel. When the candidate solution equals the optimal solution in a given dimension, the corresponding pixel colour matches that of the target image. Any

\[\text{Note that, the subscripts } ij, \text{ corresponding to index } ij \text{ in the image matrix, are removed for brevity}\]
other value for a decision variable is encoded as a shade of grey, depending on its proximity to the optimal value.

To address problems with no known optima, where there exists a best-known solution, the best-known solution can be used as the target. In the event that the best-known solution is improved upon, a special-purpose image modifier, such as a semi-transparent overlay can then be used to convey this information to the user. Similarly, in cases with no known solution, it is conceivable to define a goal solution, which is common in goal-programming methods, that is ideal/acceptable for the problem at hand. The objective then becomes to minimize the distance between the candidate solution(s) and the goal solution, whereby the visualization process can proceed as above by using the goal solution as the target.

This mapping technique is visualized in Fig. 10 and various examples, corresponding to the Spherical function (optimum at 0 with bounds of [−5.12, 5.12])\(^{536}\), are provided in Fig. 11. To produce the visualizations shown in Fig. 11a and b, two solutions were generated using a normal distribution with a mean of 0 and standard deviations of 0.5 and 1.5, respectively. Clearly, the solution with the lower standard deviation (and hence closer proximity to the optimum), has less visual noise in the resulting image. Figure 11c and d depict solutions that are simply shifted from the optimum, thereby evidencing how values above the optimum produce lighter pixels (Fig. 11c) and values below the optimum produce darker pixels (Fig. 11d). Finally, Fig. 11e demonstrates that applying this mapping to the optimum produces the target image. A key observation is that this mapping function preserves the relationship between solution quality and image quality, but is far more broadly applicable than the scheme presented in Section 5. However, there is an implicit assumption that solution quality can be quantified by proximity to the optimal solution, which may not necessarily be true for multi-modal problems. Nonetheless, this mapping function provides evidence that more complex, and broadly applicable, mapping functions can be devised in the proposed framework.

It should be explicitly noted that the mapping function provided by (6) is only one particular example from a much larger study regarding the application of image-based visualization to arbitrary problems and, therefore, the remainder of this study does not consider it as comprehensively as the mapping function given in (5). Rather, experimentation making use of (6) is included largely as a proof-of-concept and to demonstrate the advanced capabilities of the proposed framework.

4 Experimental design

This section provides background information on the image evaluation metrics and optimizers used in this study.

4.1 Fitness evaluation metrics

This section briefly describes the image comparison metrics that were used as objective fitness measures, assuming that the mapping function provided in (5) was employed. For all examined fitness measures, it is assumed that A and B are \(m \times n\) matrices representing two images, respectively, such that the entries \(A_{ij}\) and \(B_{ij}\) correspond to the pixels at location \((i, j)\) in images A and B. Each metric imposes different characteristics on the landscape of the search process, as exemplified in Figs. 12 and 13, which presents the 2D fitness landscape for each measure with a target solution of \(B = (0.25, 0.75)\). Note that, the landscape visualizations correspond to a target image with two pixels, specifically with greyscale values of 0.25 and 0.75, respectively and corresponds to a purely synthetic scenario constructed only to assist with visualizing the different landscapes induced by the various image comparison metrics. Each point in the landscape corresponds to some other two pixel image, while the height corresponds to the image comparison metric between that point (i.e., image) and the target image B. It should also be noted that the landscapes in Figs. 12 and 13 are for the 2D formulations only, and that the landscape characteristics may change with an increase in dimensionality.

4.1.1 Sum of absolute error

Given two images, the sum of absolute error (SAE) \([8]\) is calculated as

\[
\arg\min_A SAE(A, B) = \sum_{i=1}^m \sum_{j=1}^n |A_{ij} - B_{ij}|, \tag{7}
\]

whereby the SAE corresponds to sum of absolute differences between each pixel in the candidate solution and the corresponding pixel in the target image. A value of 0 for the SAE indicates an exact replica of the target image. The 2D landscape for the SAE metric is provided in Fig. 12a.
The SAE metric is both unimodal and fully separable. As the dimensionality increases, the basic characteristics of the landscape for the SAE metric will remain fixed.

4.1.2 Mean squared error

The mean squared error (MSE) \cite{9} is the mean of the squared error of each pixel, given by

\[
\arg \min_A MSE(A, B) = \frac{1}{mn} \sum_{i=1}^{m} \sum_{j=1}^{n} (A_{ij} - B_{ij})^2, \tag{8}
\]

where an MSE value of 0 indicates an exact replica of the target image. Compared to the SAE measure, MSE is more susceptible to outliers, and thus provides a heavier penalty for large errors, due to the square term in its calculation.

The 2D landscape for the MSE metric is provided in Fig. 12b. The MSE measure is also both fully separable and unimodal, and the characteristics will be similar as dimensionality increases.

4.1.3 Peak signal-to-noise ratio

The peak signal-to-noise ratio (PSNR) \cite{13} is used as a measure of quality between two images, typically an original and compressed image. This can be calculated according to

\[
\arg \max_A PSNR(A, B) = 10 \log_{10} \left( \frac{R^2}{MSE(A, B)} \right), \tag{9}
\]

where $R$ is the maximum possible pixel value in the image, according to the data type. In the context of this work, $R = 1$ for continuous images and $R = 255$ for discrete images. Alternatively, one can specify a value for $R$. Note that, in the case of continuous images, the PSNR is simply a logarithmic function of the inverse of the MSE, which provides a smoothing effect. For the PSNR metric, larger values indicate a higher degree of similarity. The 2D landscape for the PSNR metric is provided in Fig. 12c and shows a relatively neutral landscape, with a peak near the optimal solution. The PSNR metric is also fully separable and depicts unimodality in two dimensions. Based on the definition in (9), these characteristics are expected to hold in higher dimensions.

4.1.4 2D correlation coefficient

The 2D (Pearson) correlation coefficient (PCC) \cite{7} measures the correlation between two matrices and can be used as an image similarity measure. The 2D correlation coefficient is given by

\[
\arg \max_A r(A, B) \tag{10a}
\]

where

\[
r(A, B) = \frac{\sum_{i=1}^{m} \sum_{j=1}^{n} (A_{ij} - \bar{A})(B_{ij} - \bar{B})}{\sqrt{\left( \sum_{i=1}^{m} \sum_{j=1}^{n} (A_{ij} - \bar{A})^2 \right) \left( \sum_{i=1}^{m} \sum_{j=1}^{n} (B_{ij} - \bar{B})^2 \right)}}, \tag{10b}
\]

with a range of values between -1 and 1. Positive values for the correlation coefficient indicate a positive correlation among the values of the matrices whereas negative
Fig. 12 Visualization of the 2D fitness landscape for the SAE, MSE, PSNR, and PCC fitness functions. The target solution for each landscape, denoted by \( B \), is located at \((0.25, 0.75)\). Thus, the fitness correlation coefficients indicate a negative correlation was identified. The 2D landscape for the PCC metric is provided in Fig. 12d, which shows perfect positive correlation for solutions with \( x_2 > x_1 \) and perfect negative correlation for solutions with \( x_1 > x_2 \). In only two dimensions, there are no other correlations visible in the landscape, which would not be the case for higher dimensions. Therefore, the landscape characteristics are expected to change in higher dimensions. In contrast to the previous metrics, the PCC measure is non-separable.

4.1.5 Structural similarity index

The structural similarity index (SSIM) \[32\] is a metric for comparing the similarity of two images. In contrast to other measures, the SSIM quantifies perceived changes rather than absolute changes. The SSIM can be calculated according to

\[
\arg \max_A SSIM(A, B) = [l(A, B)^\alpha \cdot c(A, B)^\beta \cdot s(A, B)^\gamma],
\]

where \( l(A, B) \) is a measure of luminance, given by

\[
l(A, B) = \frac{2\mu_A\mu_B + c_1}{\mu_A^2 + \mu_B^2 + c_1},
\]

c(A, B) is a measure of contrast, given by

\[
c(A, B) = \frac{2\sigma_A\sigma_B + c_2}{\sigma_A^2 + \sigma_B^2 + c_2},
\]

and \( s(A, B) \) is a measure of the structure given by

\[
s(A, B) = \frac{\sigma_{AB} + c_3}{\sigma_A\sigma_B + c_3},
\]

assuming the following definitions:
$\mu_A, \mu_B$ are the averages of $A$ and $B$
$\sigma_A, \sigma_B$ are the standard deviations of $A$ and $B$
$\sigma_{AB}$ is the covariance of $A$ and $B$
$L$ is the range of the pixel values
$c_1 = (k_1 L)^2$, with $k_1 = 0.01$ by default
$c_2 = (k_2 L)^2$, with $k_2 = 0.03$ by default
$c_3 = \frac{c_2}{2}$

If the weights $\alpha, \beta, \gamma$ are set to 1, as is the case with this study, the SSIM calculation simplifies to

$$SSIM(A, B) = \frac{(2\mu_A\mu_B + c_1)(2\sigma_A\sigma_B + c_2)}{(\mu_A^2 + \mu_B^2 + c_1)(\sigma_A^2 + \sigma_B^2 + c_2)}.$$  \hfill (12)

With regards to the SSIM measure, larger values indicate a higher degree of similarity. The 2D landscape for the SSIM metric is provided in Fig. 13, which shows that the landscape characteristics vary with the target image. The SSIM metric is non-separable and the characteristics are expected to change with an increase in problem dimensionality.

Fig. 13 Visualization of the 2D fitness landscape for SSIM with various target solutions, denoted by $B$. The fitness value (z-axis) at each point represents the value of the SSIM calculated using the points $(x_1, x_2)$ and $B$. The mappingschemes for image-based benchmarking

In this section, a variety of mapping schemes which demonstrate the efficacy and flexibility of the proposed image-based visualization framework, in the context of image replication using the mapping function described in Section 3.2, are presented.

After each iteration in the optimization process, the mapping function given in (5) was applied to the best solution in the population, thereby providing a visualization of the best fitness over time. While any image, assuming the
correct dimensions, can be used in this framework, the well-known “Lena” image from the standard image-processing benchmarks was employed as an arbitrary example. The generated image is then displayed to the user, thereby giving an indication of the search characteristics. For example, an improvement in the overall image quality between two subsequent iterations indicated that the best fitness had improved while an image that does not change for a number of iterations would indicate that the global best has stagnated. Unless otherwise noted, the results presented are taken from a single execution of each algorithm. It should be explicitly noted that the purpose of these experiments was not to perform a comparative empirical study of the examined algorithms. Rather, the purpose of these experiments is to elucidate the efficacy and flexibility of the proposed image-based visualization framework. Therefore, the experimental results are meant to demonstrate that meaningful observations can be made regarding the respective optimization processes using image-based visualization.

Unless otherwise stated, control parameter values were set as follows. For the PSO algorithm, a global-best (star) topology was employed with an asynchronous iteration strategy. Control parameter values were set as \( \omega = 0.729844, c_1 = c_2 = 1.49618 \). Velocity clamping \([19]\) was employed with \( v_{\text{max}} = 0.1 \ast (x_{\text{max}} - x_{\text{min}}) \) and \( v_{\text{min}} = -v_{\text{max}} \). Velocities were initialized to 0. The personal best position of a particle was only updated if the candidate position was both feasible and had a better fitness than the current personal best position. For the DE and GDE3 algorithms, the DE/rand/1/bin strategy was employed with \( F = 0.5 \) and \( CR = 0.9 \). To prevent infeasible solutions, a clamping boundary strategy was employed by all algorithms. Furthermore, all algorithms had a population size of 100 and were executed for 10,000 \( D \) function evaluations, where \( D \) is the problem dimension. Unless otherwise stated, it was assumed that the optimization objective was to be minimized.

**Mapping Scheme 1** (Continuous, unconstrained, single-objective) Figure 14 shows a comparison of two algorithms, namely PSO and DE using SAE, given in (7), as the fitness function. Immediately, the positive correlation between the objective fitness and the visual quality was observed, which is a critical aspect of the image-based visualization that differentiates it from previous approaches. Interestingly, at iteration 7990, both PSO and DE had the same fitness value, yet produced two distinct images, indicating that they did not correspond to the same location in the search space. Without the visualization, one would not be able to differentiate two distinct solutions with equal fitness values. As the optimization progresses, both optimizers depicted very similar fitness values and both were able to replicate the target image very accurately.

One drawback that can be seen from the image-based visualizations in Fig. 14 is that once the solution quality reaches a certain threshold, the resulting images become too similar to discern much valuable information regarding the respective optimization processes. To address this concern, another visualization technique, which highlights the dissimilarities between the resulting image and the target image, was also produced and is shown to the right of the corresponding image. In these visualizations, referred to as error heatmaps, each pixel was colorized according to the difference between its corresponding decision variable value and the value of the target decision variable. Note that, the error heatmaps correspond to a special case of image-based visualization whereby the target image is white, negative error values are mapped to blue pixels, and positive error values are mapped to red pixels. Thus, the overall solution quality can be inferred by the amount of white space in the image such that an optimal solution would be a completely white image while sub-optimal decision variables are colorized according to the magnitude of their respective error. However, the intended purpose of the error heatmap is to highlight the dissimilarities, which is somewhat difficult to discern in the original images once the fitness is sufficiently close to the optimal, and therefore they serve as a complementary visualization technique.

To ascertain the effect of increasing the problem dimensionality, Fig. 15 shows the performance of PSO on a 900D (i.e., 30x30), 1600D (i.e., 40x40), and 2500D (i.e., 50x50) problem. These experiments also highlight the ability of the visualization technique to trivially scale with problem dimensionality. Figure 15 clearly depicts the increased difficulty as dimensionality increased. After only 10,000 iterations, the visual disparity between the images produced on each of the problems is readily apparent. The increased difficulty is also clearly visible in the error heatmaps – after 50,000 iterations, there was nearly no errors visible on the error heatmap for the 900D problem, while the error heatmap for the 2500D problem depicted a significant number of sub-optimal decision variables.

**Mapping Scheme 2** (Alternative fitness functions for continuous, unconstrained, single-objective) In this scheme, alternative fitness functions for continuous, unconstrained, single-objective problems are investigated for the 30x30 Lena image using both the PSO and DE algorithm. The overall goal is to investigate various landscape characteristics and ascertain the visual similarity induced by alternative fitness functions. Note that, the results for the SAE measure from Mapping Scheme 1 are not repeated here.
5.1 Mean squared error

Figure 16 presents the results when the MSE, as given in (8), was employed as the fitness function. As can be expected, the general observations made when MSE was used as a fitness function are largely the same as when SAE was used. Specifically, the MSE is an effective measure of similarity that appears to match well with human perception. However, the MSE produced images of similar quality quicker than those produced when SAE was employed; the image quality after 5000 iterations using MSE is roughly the same as the quality of images after 7990 iterations when SAE was used. This result is expected given that MSE more harshly penalizes large errors, thereby causing a more rapid initial improvement.

5.2 2D correlation coefficient

Figure 17 presents the results when the 2D correlation coefficient, as given in (8), was employed as the fitness function. To employ the PCC measure in a minimization context, objective values were negated. The images produced using PCC depict an inherent property of the PCC measure, namely that it measures correlation, rather than direct similarity. This correlation is most clearly depicted by the PSO results, whereby the resulting images are very highly correlated with the target image, which results in a near perfect fitness value but an image that is not identical to the target. Interestingly, this also leads to an error heatmap whereby the overall structure of the target image is visible, further highlighting the correlation between the candidate solution and the target image. In contrast, the DE algorithm produced a target image that had much greater visual similarity to the target, despite having a nearly identical fitness as the PSO algorithm.

5.3 Peak signal-to-noise ratio

Figure 18 presents the results when the PSNR measure, as given in (9), was employed as the fitness function. To employ the PSNR measure in a minimization context, objective values were negated. Examining the results using
PSNR, notably different fitness profiles were observed, attributed in part to the logarithmic nature of the PSNR measure. Specifically, the fitness plots depict numerous relatively flat regions, followed by sharp decreases in fitness. Nonetheless, the PSNR measure depicted enhanced visual quality as the fitness improved, thereby indicating it was a suitable metric for image-based visualization.

5.4 Structural similarity index

The final measure employed in this study was the SSIM, as given by (11). Again, the values of SSIM were negated to be employed in a minimization context. Note that, using SSIM as a fitness measure induces non-separability because (11) makes use of properties dependent upon the entire image, such as averages and standard deviations, in contrast to using the fully-separable SAE measure, which is dependent only upon individual pixel values. Therefore, SSIM is expected to be a more challenging metric to optimize. As can be seen in Fig. 19, neither optimizer produced an image that was visually close to the target. An interesting observation was that the non-separability can be readily observed in the error heatmaps for both optimizers. This correlation is observed via the vertical segments in the error heatmaps, which alternate in colour across the horizontal axis. In contrast, the other, fully-separable measures, depicted scattered (i.e., independent) pixel-wise errors. Therefore, the difficulty introduced by the non-separability of SSIM was readily apparent in the results. Furthermore, an interesting observation was that the image-based visualization provided directly identifiable information about the underlying fitness function.

5.5 Summary

To provide an overall summary of the results using various fitness functions, Figs. 20 and 21 present the resulting images at various iterations for PSO and DE, respectively. After 90000 iterations, the images produced when SAE, MSE, and PSNR were used as fitness functions were nearly identical to the target for both algorithms. In contrast, the worst visual quality was attained when SSIM was employed.
Fig. 18 Image-based visualization applied to various solutions attained by PSO and DE on a 900D (30x30) continuous image reconstruction problem using PSNR as the fitness function. Images scaled by 400%.

Fig. 19 Image-based visualization applied to various solutions attained by PSO and DE on a 900D (30x30) continuous image reconstruction problem using SSIM as the fitness function. Images scaled by 400%.

Fig. 20 Comparison of image-based visualizations produced by PSO solutions on a continuous image reconstruction problem using various fitness functions. Images scaled by 400%.
as the fitness function. The images produced when PCC was used as the fitness function were structurally similar to the target, yet the colour was primarily correlated, rather than identical. A further interesting observation is that the images produced by PSO after 1000 iterations had much higher visual quality than those produced by DE. This implies that the PSO demonstrated superior performance in the early stages of the search. However, after 10000 iterations, the images produced by DE were generally of a higher quality than those produced by PSO.

**Mapping Scheme 3** (Discrete, unconstrained, single-objective) In the image-based visualization framework, a discrete optimization problem can be easily defined by considering a target image with a discrete intensity value for each pixel. As such, the pixel values are taken as (discrete) integers in the range $[0..255]$, which correspond to the 8-bit greyscale intensity. Given the larger domain, the corresponding fitness values are expected to be higher than the previous results using a continuous domain. Similar to the other case studies, both the DE and PSO algorithms were applied on the 30x30 Lena image.

Given that both algorithms are inherently continuous meta-heuristics, they must be adapted to handle discrete problems. A simple method to utilize continuous algorithms for discrete problems is to convert the variable values to discrete values before evaluating their objective fitness. In this case, the variable values (i.e., the pixel grey levels) are rounded to closest integer number in the range $[0..255]$ immediately before the objective function evaluation. The remaining elements of the algorithms have not been modified; the update phase, operators, and the selection schemes are unaltered.

**Figure 22** presents the results on a discrete image reconstruction problem. The image resulting from employing DE does not depict good visual quality after 9,000 iterations, while PSO was able to attain an image with much better quality. Contrasting the quality of images produced by the continuous and discrete problems indicates that the image is improved at a much slower rate for discrete problems, which provides direct evidence of their increased difficulty for the inherently continuous optimizers. Continuous optimization problems are, in general, easier to solve than discrete problems given that the smoothness of the landscape facilitates deduction of information about solutions in the neighbourhood. Furthermore, the many-to-one mapping imposed by the discretization process causes a reduction in exploration power for the optimizers. These observations are directly indicated through the image-based visualization framework.

**Mapping Scheme 4** (Partially-separable, unconstrained, single-objective) As discussed in Mapping Scheme 2, the SSIM is calculated using characteristics of the entire image, not independent pixels. Thus, the SSIM introduces a non-separable fitness function. To highlight the difficulty of non-separability in an optimization problem, both SAE and SSIM were evaluated on distinct regions (i.e., sub-images) of the target image. Therefore, a single image can be used to visually ascertain the effect of separability. Specifically, the target image was divided into two regions, where one region...
Fig. 22 Image-based visualization applied to various solutions attained by PSO and DE on a 900D (30x30) discrete image reconstruction problem using SAE as the fitness function. Images scaled by 500%.

was evaluated using SAE and the other was evaluated using SSIM according to

\[
f(x, t, p) = SAE(A_1, B_1) + \frac{1}{SSIM(A_2, B_2)},
\]

(13a)

\[x[1 : \lfloor pD \rfloor] \mapsto A_1\]

(13b)

\[x[\lceil pD \rceil : D] \mapsto A_2\]

(13c)

\[t[1 : \lfloor pD \rfloor] \mapsto B_1\]

(13d)

\[t[\lceil pD \rceil : D] \mapsto B_2\]

(13e)

where \( x \) is the candidate solution, \( t \) is the (flattened) target image, and \( p \in [0, 1] \) is a control parameter that quantifies the level of separability. Using the parameter \( p \), the level of separability can be explicitly controlled; setting \( p = 1 \) results in the SAE measure being used for the entire image (i.e., a fully separable problem), \( p = 0 \) results in the SSIM measure being used for the entire image (i.e., a non-separable problem), and \( 0 < p < 1 \) results in exactly \((1 - p)\)% of the image being evaluated using the fully-separable SAE metric. In this study, \( p \) is set to 0.5, thus introducing a problem that is 50% separable and 50% non-separable.

Figure 23 presents the results on a partially-separable problem using DE and PSO algorithms, where the left half was evaluated using SAE as the fitness function and the right half of the image was evaluated using SSIM. The performance of two algorithms, DE and PSO were comparable and reached the same fitness value after approximately 4,000 iterations. Despite this, it was observed that the optimizers were able to produce images with reasonable visual quality on the half evaluated using SAE after only 10,000 iterations, while the SSIM half of the image had much lower quality. The quality of the images produced after 30,000 iterations was much better with regards to both metrics. Nonetheless, the region optimized with SAE still depicted much better quality. This result highlights that the increased difficulty associated with non-separable problems can be easily observed through the quality of images produced, thereby providing further evidence of the merits associated with image-based visualization.

Mapping Scheme 5 (Continuous, constrained, single-objective) To produce a constrained continuous problem, the search domain, i.e., \( \text{dom}(x_i) \), was widened to a range of \([-1, 2]\), while the feasible domain remained at \([0, 1]\). This introduces an inequality constraint that feasible candidate solutions must adhere to. Thus, decision variables that were outside of the feasible domain were considered to be constraint violations. Two variants of PSO were employed to handle the constraints. The PSO with no constraint handling (PSO-NCH) variant employed no constraint handling, i.e., was a vanilla PSO, and thus was purely focused on reducing the objective function. The PSO with scaled constraint handling, no penalty (PSO-SCHNP) variant employed Deb’s constraint handling technique [5], whereby infeasible solutions were always inferior to feasible solutions and two infeasible solutions were compared according to the sum of the magnitudes of their violations. However, the fitness function did not account for the violations (i.e., there was no penalty term) and thus was simply the sum of absolute errors. Thus, the PSO-SCHNP variant first optimizes the constraints, then the fitness, which explains the increasing fitness value near the beginning of the optimization process in Fig. 24.

Figure 24 presents the results of employing image-based visualization on a constrained optimization problem. In
Fig. 23 Image-based visualization applied to various solutions attained by PSO and DE on a 900D (30x30) continuous image reconstruction problem using 50% SAE and 50% SSIM as the fitness function. Images scaled by 500%.

Fig. 24, the individual decision variable violations are visualized via the pixel colour, while the colour of the border represents the absolute sum of violation amounts across the entire solution. This highlights another strength of image-based visualization, namely that additional features can also be visualized through the use of colour and, for example, adding borders. Interestingly, the raw fitness of PSO-NCH was superior to PSO-SCHNP. However, this is a result of its inherent focus on purely minimizing the fitness, without regard for the constraints, as can been seen by its inability to attain a feasible solution. In contrast, PSO-SCHNP was able to rapidly attain a feasible solution, after which the solution remained feasible, then slowly improved the fitness. Importantly, such observations could not be made without the use of this visualization framework, which inherently embeds important information about the search process in the visualization.

Mapping Scheme 6 (Dynamic, continuous, unconstrained, single-objective) To simulate a continuous dynamic environment in the context of this study, an animated GIF image was used as the target with \( \text{dom}(x_i) = [0, 1] \). The target GIF consisted of 21 frames, which were taken as successive environments for the optimizer. Specifically, the total number of iterations was divided equally into 21 environments, where each environment used the subsequent frame from the animation as the target image. Therefore, each environmental change introduced a new optimal solution and, as a byproduct, a different fitness value associated with each solution.

To mitigate the effect of change detection mechanisms on the results, the optimizers were made aware of the frequency of environmental changes. A re-evaluating change response mechanism was employed whereby individuals re-evaluate their fitness (as well as the personal bests and global best for PSO) when an environmental change has occurred. The results presented in Fig. 25 depict the performance of PSO and DE on the dynamic problem. While PSO depicted superior performance on the initial environment, it is evident that the PSO algorithm struggled to effectively recover from the environmental changes, which is to be expected given that PSO is known to suffer from obsolete...
memory in dynamic environments [2]. The visual quality of the subsequent images produced by the DE optimizer depicted better visual quality, thereby indicating superior overall performance by DE on this dynamic problem.

**Mapping Scheme 7** (Multi-objective, unconstrained) In order to produce a multi-objective environment in the context of this study, two target images (i.e., sub-objectives) were considered simultaneously. To produce a bi-objective optimization problem, two target images were taken as an original target image and its inverse. The primary image was taken as a black and white image (corresponding to a binary optimization problem), and its inverse was created by inverting the pixel values (i.e., setting values of 0 to 1 and vice versa). As such, the second target image is in direct conflict with the original image. GDE3 was utilized to optimize both images in a multi-objective context using PSNR as the evaluation metric. Formally, the problem can be defined by

Minimize: \( f(x) = (PSNR(x, T), PSNR(x, T')) \) (14)

where \( T \) is the target image and \( T' \) is the inverse of \( T \).

Figure 26 presents the results of the bi-objective optimization problem. It was observed that GDE3 was able to locate the extreme points on the Pareto front. Moving along the Pareto front, the quality of images were degraded because the optimizer is forced to make a trade-off between the two conflicting objectives. Thus, solutions towards the middle segment of the Pareto front don’t necessarily correspond to meaningful images but provide visual depictions of the various trade-offs in each sub-objective.

### 5.6 Summary

In this section, a variety of mapping schemes were exemplified to demonstrate the efficacy and flexibility of the proposed image-based visualization framework in the context of image replication. These case studies encompassed a wide variety of optimization environments, thereby elucidating the ability of the proposed framework to be applied in any given optimization problem type. A variety of image quality metrics were examined as fitness functions to ascertain their alignment with human perception and to induce different landscape characteristics for the optimization process. A number of critical insights about the optimization process and, in some cases, the fitness function itself were made through inspection of the generated images.

### 6 Image-based visualization for arbitrary single-objective benchmark functions with known optima

This section examines the image-based visualization framework applied to well-known single-objective (minimization) benchmark problems with known optima using the mapping function given in (6). All problems were optimized in 900 dimensions, corresponding to a 30x30 target image and were optimized with both the DE and PSO optimizers. The experimental procedures, specifically the algorithmic control parameters and image generation process, were the same as in Section 5, with the exception of (6) used as the mapping function. Table 2 gives a brief overview of the benchmark functions examined in this study, presented in alphabetical order. The equations for the functions can be found in (15) to (21). These functions were chosen to represent a wide variety of characteristics, such as modality, separability, and location of the optimum.

\[
f_1, \quad \text{the Qing function, defined as}
\]

\[
f_1(x) = \sum_{j=1}^{D} (x_j^2 - j)^2
\]
Fig. 26 Image-based visualization applied to various solutions attained by GDE3 on a 900D (30x30) bi-objective, binary image reconstruction problem using PSNR as the fitness function. The two objectives correspond to reconstructing the standard and inverted images, respectively. Images scaled by 500%

with each \( x_j \in [-500, 500] \).

\[ f_2(x) = 10D + \sum_{j=1}^{D} (x_j^2 - 10 \cos(2\pi x_j)) \]  

(16)

with each \( x_j \in [-5.12, 5.12] \).

\[ f_3(x) = \sum_{j=1}^{D-1} \left(100(x_{j+1} - x_j^2) + (x_j - 1)^2\right) \]  

(17)

with each \( x_j \in [-30, 30] \).

\[ f_4(x) = -\cos \left(2\pi \sum_{j=1}^{D} x_j^2 \right) + 0.1 \sqrt{\sum_{j=1}^{D} x_j^2 + 1} \]  

(18)

with each \( x_j \in [-100, 100] \).

\[ f_5(x) = \sum_{j=1}^{D} x_j^2 \]  

(19)

with each \( x_j \in [-5.12, 5.12] \).

\[ f_6(x) = \frac{1}{2} \sum_{j=1}^{D} (x_j^4 - 16x_j^2 + 5x_j) \]  

(20)

with each \( x_j \in [-5.00, 5.00] \).

\[ f_7(x) = 1 - \frac{1}{D} \sum_{j=1}^{D} \cos(kx_j) \exp\left\{-\frac{x_j^2}{2}\right\} \]  

(21)

with each \( x_j \in [-\pi, \pi] \). In our study, \( k \) is taken to be 10.

Table 2 Benchmark functions with known optima

| Problem         | Domain          | Optimum |
|-----------------|-----------------|---------|
| Qing            | \([-500, 500]^D\) | 0       |
| Rastrigin       | \([-5.12, 5.12]^D\) | 0       |
| Rosenbrock      | \([-30, 30]^D\)  | 1       |
| Salomon         | \([-100, 100]^D\) | 0       |
| Spherical       | \([-5.12, 5.12]^D\) | 0       |
| Styblinski-Tang | \([-5.5]^D\)     | \(-2.90354\) |
| Wavy            | \([-\pi, \pi]^D\) | 0       |

Figure 27 presents a comparison of PSO and DE on the Qing function. It is observed that relative to the initial solutions, the fitness of PSO more rapidly improves. This improvement in fitness is directly correlated with an improvement in the image quality. However, after 1000 iterations, both optimizers arrive at solutions with similar fitness, thereby producing images that are of similar visual quality. Nonetheless, the pixel-level differences of the resulting images indicate that despite the optimizers attaining a similar fitness value, the solutions correspond to different areas of the search space. Similarly, the relatively high-quality images indicate that the fitness associated with both optimizers are near-optimal.

Figure 28 presents a comparison of PSO and DE on the Rastrigin function. As with the Qing function, it was observed that PSO led to a more rapid improvement in fitness. However, the performance of the optimizers was nearly identical at 2625 iterations, at which point the differences in the images clearly indicates that different solutions were attained. From iteration 2625 onwards, the performance of DE, and hence the resulting image quality, was better than PSO. Despite the improved performance, it is evident from the quality of image produced that even after 20000 iterations, the solution produced by DE is still not sufficiently close to optimal.
Figure 27 presents a comparison of PSO and DE on the Rosenbrock function, where it was observed that PSO led to a much more rapid improvement in fitness and resulting image quality. Similar to the Qing function, both optimizers arrived at similar objective fitnesses after 1000 iterations, but at noticeably different locations in the search space as evidenced by the differences in the resulting images.

Figure 28 presents a comparison of PSO and DE on the Salomon function. Initially, PSO demonstrates superior performance and leads to a noticeably clearer image after 2000 iterations. However, after 2000 iterations, DE also produced a relatively high-quality image, thereby indicating that both optimizers found well-fit solutions at this point in the search. After 11747 iterations, the fitness of the optimizers is approximately equal, after which point DE outperforms PSO. Nonetheless, the high quality of the images produced by both optimizers demonstrates that both optimizers performed relatively well on the Salomon function.

Figure 29 presents a comparison of PSO and DE on the Spherical function. Note that the Spherical function is considered to be relatively simple, and thus after 1000 iterations, both PSO and DE found high-quality solutions, thereby resulting in high quality images. Despite the similar fitness and image quality, the image produced by PSO is noticeably better, corresponding to its superior performance.

Figure 30 presents a comparison of PSO and DE on the Styblinski-Tang function. The poor quality of images, even after 20000 iterations, indicates that neither optimizer was able to achieve a reasonably-fit solution. Moreover, the relatively low quality images facilitates concluding that the Styblinski-Tang function was the most challenging problem for both optimizers among those considered. Furthermore, the black pixels in the resulting images indicate that both optimizers were deceived into searching near the lower end of the search domain in many dimensions. Based on this observation, it was hypothesized that a basin of attraction exists in the vicinity of the lower bound of the search space. Visual inspection of the 2D problem landscape confirmed this hypothesis and exemplified a striking benefit to the usage of image-based visualization – namely, that an intrinsic property of the fitness landscape associated with
Fig. 29 Image-based visualization applied to various solutions attained by PSO and DE on the 900D Rosenbrock function using Lena (30x30) as the target image. Images scaled by 500%.

Fig. 30 Image-based visualization applied to various solutions attained by PSO and DE on the 900D Salomon function using Lena (30x30) as the target image. Images scaled by 500%.

Fig. 31 Image-based visualization applied to various solutions attained by PSO and DE on the 900D Spherical function using Lena (30x30) as the target image. Images scaled by 500%.
benchmark problem was made clearly visible through the use of the image-based visualization framework.

Finally, Fig. 33 presents a comparison of PSO and DE on the Wavy function. Again, the low quality of solutions indicated that neither optimizer was able to effectively optimize this problem, even after 50000 iterations. As with most problems, PSO demonstrated superior initial performance. After 29481 iterations, the fitness of both optimizers was approximately equal, after which DE demonstrated superior performance. Regardless of its superior performance, DE did not arrive at a high-quality solution overall and thus produced an image that did not reasonably replicate the target.

6.1 Gradient target image

This subsection replicates the previous experiments using a simple gradient of grayscale, referred to as Gradient, as the target image to ascertain the visualization capabilities given a less complex target image, for the Rastrigin and Spherical functions. The primary purpose of these experiments is to provide further evidence in support of using a non-simple image as the target image, given that it provides better alignment with human perception; it is expected that the use of the grayscale image will provide less insight about the optimization process relative to the use of a more complex image with semantic meaning, e.g., the Lena image.

Figure 34 presents a comparison of PSO and DE on the Rastrigin function using the grayscale Gradient image. In contrast to Fig. 28, a much clearer improvement in the image quality can be ascertained over time when using the Lena image. Notably, it is clear from Fig. 28 (i.e., Lena target image) that DE exhibited a (slightly) better fitness than PSO after 20000 iterations whereas this is much less clear in Fig. 34 (i.e., grayscale Gradient target image). While it is fair to point out that these results correspond to different algorithm executions, such a comparison is still reasonable given the similarity in fitness profiles.

As an additional example, Fig. 35 presents a comparison of PSO and DE on the Spherical function using the grayscale Gradient image. Note that, the Spherical function presents a much simpler optimization problem and, as seen
In Fig. 31, a high-quality image is produced after only 1000 iterations. However, in comparison to Fig. 31, the images produced using the grayscale Gradient as target image do not have the same degree of visual similarity to their respective target. This provides further evidence that having a target image with semantic meaning is beneficial to the user; visual similarity is easier to perceive when the images have semantic meaning.

6.2 Summary

In summary, this section exemplified the usage of the image-based visualization framework on arbitrary benchmark problems with known optima. The results depict how the quality of the resulting images aligns with the observations made using traditional performance plots but additionally facilitates critical insights into both the optimizers’ performance and the intrinsic characteristics of the benchmark problems themselves. Furthermore, a brief experiment using a simple gradient image as the target revealed that the usage of (non-simple) images with semantic meaning, e.g., Lena, provide more a greater level of information to the user given their inherently ability to more clearly align with the global fitness via visual similarity.

7 Validating the image-based visualization framework

This section presents the results of a user survey that was conducted to validate the merits of the proposed image-based visualization framework.

7.1 Survey methodology

Participants were asked to complete a survey consisting of 10 questions. The first 9 questions asked the users “Rank each of the following images based on their similarity to the target image shown. When considering similarity, consider both the colour/shade and brightness of each pixel, with respect to the target image. Note: Images may be scaled” and presented an image such as the one in Fig. 36. Images provided in each of the questions were taken from experimental results, such as those presented
in Sections 5 and 6. The difference in relative magnitudes among the presented images varied for each question. Seven of the questions used the Lena image, while the remaining two used the gradient image. These questions were used to assess the correlation between the objective fitness of solutions and the subjective assessment of image quality by the participants.

For each of the 9 ranking questions, users were tasked with ranking each of the images (denoted as A, B, C, and D), based on their similarity to the target image, as one of “Most Similar”, “Fairly Similar”, “Less Similar”, or “Least Similar”. The true rankings were determined based on the objective fitness of the corresponding solutions; the image corresponding to the best fitness was regarded as “Most Similar”, while the image corresponding to the worst fit solution was regarded as “Least Similar”.

To assign a score to each response for questions 1 through 9, numerical values were assigned to each rank as follows: “Most Similar” = 4, “Fairly Similar” = 3, “Less Similar” = 2, “Least Similar” = 1. The score for each response was calculated as

$$\max(3 - |R - T|, 0),$$

where $R$ is the numerical value corresponding to the user’s response and $T$ is the value associated with the true rank of the image. Note that, this means a correct ranking was awarded a score of 3, while a score of 2 indicates the user’s rank was off by one position in either direction. For example, assume the true rank of an image is “Fairly Similar”, i.e., the image is the second best fit image presented. A response of “Fairly Similar” was awarded 3 points, a response of either “Most Similar”, or “Less Similar” was awarded 2 points, while a response of “Least Similar” was awarded 1 point.

The final question presented the users with visualizations of two solutions to a 900D optimization problem. The visualization techniques consisted of image-based visualization, parallel coordinates, t-SNE, and RadViz. The wording of the question was as follows: “This question compares various different visualization techniques on a 900D problem. In each of the following, the same two solutions to a minimization problem are visualized. One solution has a fitness of $\sim63,567,000$ while the other has a fitness of $\sim302$. Which of the following most clearly represents the difference in magnitude between these solutions? Note: only the first 200 dimensions are shown for parallel coordinates while only the first 20 dimensions are shown for RadViz due to limitations in the visualization software.” This question was meant to determine the visualization technique that users perceived to most accurately portray the magnitude of change between solutions.

The survey participants consisted of 42 students, both undergraduate and postgraduate, from Ontario Tech University. The survey was conducted using Google Forms, with results collected between 5 and 7 April 2021.

### 7.2 Validation results

First, it should be noted that a “perfect” score for the ranking questions, whereby the user correctly identified the rank associated with each of the 4 images for all 9 questions, was 108 (i.e., 3x4x9). The average participant score was 96.8 while the median score was 98. A histogram of participant scores is provided in Fig. 37. Over all 1512 responses (42 participants over 9 questions with 4 images each), 73.4% of responses received a score of 3 (i.e., the image was ranked correctly), 22.8% scored 2 points (i.e., the rank was off by 1 in either direction), 3.0% scored 1 point (i.e., the rank was off by 2), and 0.7% scored 0 points (i.e., the best image was ranked as the least similar or vice versa). These results strongly confirm that there is a correlation between the objective fitness value of candidate solutions and the

![Fig. 36 Sample ranking question presented to users during the validation study](image)

![Fig. 37 Histogram of participant scores on survey questions 1 through 9. A “perfect” score is 108](image)
perceived quality of the resulting images produced by the proposed image-based visualization.

With respect to the final question, 76.2% of users indicated that image-based visualization most clearly represented the difference in magnitude between the two solutions, 14.3% selected parallel coordinates, while both t-SNE and RadViz were each selected by 4.8% of participants. This result provides strong evidence that users are able to more accurately perceive differences in solution quality using image-based visualization compared to other visualization techniques.

8 Conclusions and future work

The primary aim of this study was to propose a novel image-based visualization framework for large-scale global optimization. The proposed framework is the first instance of a visualization technique that explicitly visualizes both decision-space and objective-space without dimensionality reduction. The proposed framework offers a number of advantages over existing visualization techniques, such as dimensionality preservation and scalability, alignment with human perception, and the flexibility to visualize a wide variety of different problem types. However, it should be noted that the scalability is with regards to the number of dimensions, not necessarily the number of entities. In contrast, many of the existing techniques offer strong scalability in terms of the number of entities, while sacrificing the ability to maintain visualization of complex relationships between variables.

The flexibility and robustness of the proposed framework were first demonstrated on a suite of different optimization types and mapping schemes using image replication as an example optimization problem. The experiments encompassed mapping schemes for continuous, discrete, constrained, dynamic, and multi-objective optimization, thereby demonstrating the suitability of the approach for a wide variety of optimization environments. A number of different image quality metrics were also examined to ascertain their alignment with human perception and to induce different landscape characteristics. The critical insights facilitate by the image-based visualization framework were further exemplified by its application to seven arbitrary benchmark functions. Through these examples, the benefits of the proposed framework were elucidated, and the ability for the user to derive critical information about the underlying optimization process was highlighted.

Finally, a user survey was conducted to validate the efficacy of the proposed visualization framework. The results strongly supported the hypothesis that users were able to perceive a correlation between solution quality and the quality of the generated images. Furthermore, when asked to select the most effective visualization among four options, 76.2% of users selected image-based visualization.

It should be noted that the proposed framework is easily extensible and can, for example, address mixed-type optimization by introducing a mapping function that explicitly accounts for differing variable types. Moreover, many of the examined mapping schemes can be composed, such that more complex types of optimization problems (such as dynamic, constrained, multi-objective optimization), can also be readily visualized.

This study constitutes only the first phase of a much larger study on image-based visualization. An immediate avenue of future work is to gather insight from various types of users, e.g., via a survey, to clearly highlight the effectiveness of our approach in comparison to other visualization techniques. Additionally, subsequent studies that examine more complex mapping functions are also an immediate avenue of future work. For example, the next phase involves further work on devising mapping functions that can visualize solutions to arbitrary optimization problem without a known optimal solution. Future work will also examine how dimensionality in terms of the number of entities can be addressed more effectively. Furthermore, the use of fitness-based mapping, where the overall quality of the resulting image is degraded based on a function of the objective fitness, will also be examined. Finally, an additional avenue of future work involves using multi-criteria and/or multi-objective approaches to simultaneously consider various measures of similarity.

Declarations

Conflict of Interests The authors declare that they have no conflict of interest.

References

1. Ankerst M, Keim DA, Kriegel HP (1996) Circle segments: a technique for visually exploring large multidimensional data sets. In: Visualization
2. Blackwell T, Branke J (2006) Multiswarms, exclusion, and anti-convergence in dynamic environments. IEEE Trans Evol Comput 10(4):459–472
3. Cheng R, Li M, Yao X (2017) Parallel peaks: A visualization method for benchmark studies of multimodal optimization. In: 2017 IEEE congress on evolutionary computation (CEC), pp 263–270. https://doi.org/10.1109/CEC.2017.7969322
4. De Lorenzo A, Medvet E, Tušar T, Bartoli A (2019) An analysis of dimensionality reduction techniques for visualizing evolution. In: Proceedings of the genetic and evolutionary computation conference companion, association for computing machinery, New York, NY, USA, GECCO ’19, pp 1864–1872. https://doi.org/10.1145/3319619.3326868
5. Deb K (2000) An efficient constraint handling method for genetic algorithms. Comput Methods Appl Mech Eng 186(2-4):311–338
K. R. Harrison et al.

6. Deb K, Pratap A, Agarwal S, Meyarivan T (2002) A fast and elitist multimultiobjective genetic algorithm: NSGA-II. IEEE Trans Evol Comput 6(2):182–197
7. Dodge Y (2008a) Correlation coefficient. Springer, New York, pp 115–119
8. Dodge Y (2008b) Least absolute deviation regression. Springer, New York, pp 299–302
9. Dodge Y (2008c) Mean squared error. Springer, New York, pp 337–339
10. He Z, Yen GG, Ding J (2021) Knee-based decision making and visualization in many-objective optimization. IEEE Trans Evol Comput 25(2):292–306. https://doi.org/10.1109/TEVC.2020.3027620
11. Hoffman P, Grinstein G, Marx K, Grosse I, Stanley E (1997) Dna visual and analytic data mining. In: Proceedings. Visualization ’97 (Cat. No. 97CB36155). IEEE, pp 437–441
12. Huang Y, Zhang J, Yang Y, Gong Z, Hao Z (2020) GNNVis: Visualize large-scale data by learning a graph neural network representation, Association for Computing Machinery, New York, NY USA
13. Huynh-Thu Q, Ghanbari M (2008) Scope of validity of psnr in image/video quality assessment. Electron Lett 44(13):800–801
14. Ibrahim A, Rahnamayan S, Martin MV, Deb K (2016) 3D-RadVis: Visualization Of pareto front in many-objective optimization. In: 2016 IEEE Congress on evolutionary computation (CEC). IEEE, pp 736–745
15. Inselberg A (1985) The plane with parallel coordinates. Vis Comput 1(2):69–91
16. Johansson J, Forsell C (2016) Evaluation of parallel coordinates: overview, categorization and guidelines for future research. IEEE Trans Vis Comput Graph 22(1):579–588
17. Keim DA, Kriegel H (1994) VisDB: database exploration using multidimensional visualization. IEEE Comput Graph Appl 14(5):40–49
18. Keim DA, Ankerst M, Kriegel HP (1995) Recursive pattern: A technique for visualizing very large amounts of data. In: Proceedings of the 6th conference on visualization’95. IEEE Computer Society, p 279
19. Kennedy J, Eberhart R (1995) Particle swarm optimization. In: Proceedings of the international conference on neural networks, vol 4. IEEE, pp 1942–1948
20. Khemka N, Jacob C (2009) Visplore: A toolkit to explore particle swarms by visual inspection. In: Proceedings of the 11th annual conference on genetic and evolutionary computation, ACM, GECCO’09, pp 41–48
21. Kohonen T (1990) The self-organizing map. Proc IEEE 78(9):1464–1480
22. Kukkonen S, Lampinen J (2005) GDE3: The Third evolution step of generalized differential evolution. In: Evolutionary computation, 2005. The 2005 IEEE congress on, vol 1. IEEE, pp 443–450
23. Lvd Maaten, Hinton G (2008) Visualizing data using t-SNE. J Mach Learn Res 9(Nov):2579–2605
24. Medvet E, Virgolin M, Castelli M, Bosman PAN, Gonçalves I, Tušar T (2018) Unveiling evolutionary algorithm representation with du maps. Genet Program Evolvable Mach 19(3):351–389
25. Meneghini IR, Koochaksaraei RH, Guimarães FG, Gaspar-Cunha A (2018) Information to the eye of the beholder: Data visualization for many-objective optimization. In: 2018 IEEE congress on evolutionary computation (CEC), pp 1–8. https://doi.org/10.1109/CEC.2018.8477889
26. Nasrolahzadeh M, Ibrahim A, Rahnamayan S, Haddadnia J (2020) Pareto-radvis: A novel visualization scheme for many-objective optimization. In: 2020 IEEE international conference on systems, man, and cybernetics (SMC), pp 3868–3873. https://doi.org/10.1109/S MC42975.2020.9283308
27. Obayashi S, Sasaki D (2003) Visualization and data mining of pareto solutions using self-organizing map. In: International conference on evolutionary multi-criterion optimization. Springer, pp 796–809
28. Price KV (1999) New ideas in optimization. In: Corne D, Dorigo M, Glover F, Dorigo M, Moscato P, Poli R, Price KV (eds) New ideas in optimization. McGraw-Hill Ltd. UK, chap An Introduction to Differential Evolution, pp 79–108
29. Shi Y, Eberhart R (1998) A modified particle swarm optimizer. In: Proceedings of the 1998 IEEE International Conference on Evolutionary Computation. IEEE, pp 69–73
30. Storn R, Price K (1997) Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces. J Glob Optim 11(4):341–359
31. Torgerson WS (1952) Multidimensional scaling: i. theory and method. Psychometrika 17(4):401–419
32. Wang Z, Bovik AC, Sheikh HR, Simoncelli EP et al (2004) Image quality assessment: from error visibility to structural similarity. IEEE Trans Image Process 13(4):600–612

Publisher’s note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Kyle Robert Harrison received the BSc and MSc degrees in computer science from Brock University, St. Catharines, Canada in 2012 and 2014 and the PhD degree in computer science from the University of Pretoria, Pretoria, South Africa in 2018. He is currently a Research Associate in the School of Engineering and Information Technology at the University of New South Wales Canberra at the Australian Defence Force Academy, Canberra, Australia. From Jan. 2019 to Aug. 2019, he was a Postdoctoral Fellow in the Department of Electrical, Computer, and Software Engineering at Ontario Tech University, Oshawa, Canada. His research interests include computational intelligence, optimization, operations research, complex networks, and fitness landscape analysis.

Azam Asilian Bidgoli is a Postdoctoral Fellow at Ontario Tech University (Oshawa, Canada) collaborating with a research team at University of Waterloo for Ontario Research Fund (ORF) project. She has received her PhD degree in Artificial Intelligence from University of Kashan (Kashan, Iran). She conducted research as a visiting scholar in Ontario Tech University for one year (2018-2019). She is also a faculty member of Electrical and Computer Engineering at Pooyesh institute of higher education (Qom, Iran). Her research is mainly focused on Machine Learning, Evolutionary Computation, and Multi-objective Optimization. She has more than 30 peer-reviewed publications in this direction. Dr. Asilian is an active reviewer for several international conferences and journals.
Dr. Shahryar Rahnamayan has received his PhD in Systems Design Engineering from the University of Waterloo. His research area includes Evolutionary Computation, Evolving Deep Neural Networks, Optimization, Big Data Analytics, and Medical Image Processing, all with focus on both theoretical contributions and tackling real-world complex problems. His around 200 publications are distributed over the mentioned research directions. He has received university-wide research excellence award and also Faculty of Engineering and Applied Science Teaching Excellence Award. Dr. Rahnamayan is recently listed in top 2% of world-wide Artificial Intelligence (AI) researchers. He has more than 20 years research experience at University of Waterloo, Michigan State University (MSU, USA), Robarts Research Institute, Simon Fraser University, and Ontario Tech University.

Kalyanmoy Deb received the bachelors degree in mechanical engineering from IIT Kharagpur, India, and the masters and Ph.D. degrees from the University of Alabama, Tuscaloosa, AL, USA, in 1989 and 1991, respectively. He is currently the Koenig Endowed Chair Professor with the Department of Electrical and Computer Engineering, Michigan State University, East Lansing, MI, USA. He is also largely known for his seminal research in evolutionary multi-criterion optimization. He has authored two textbooks on optimization and published more than 535 international journal and conference research papers to date. His current research interests include evolutionary optimization and its application in design, AI, and machine learning. He is one of the top cited EC researchers with more than 140000 Google Scholar citations. Dr. Deb was a recipient of the Shanti Swarup Bhatnagar Prize in Engineering Sciences, in 2005, the Edgeworth-Pareto Award, in 2008, the CajAstur Mamdani Prize, in 2011, the Distinguished Alumni Award from IIT Kharagpur, in 2011, the Infosys Prize, in 2012, the TWAS Prize in Engineering Sciences, in 2012, the Lifetime Achievement Award from Clarivate Analytics, in 2017, the IEEE CIS EC Pioneer Award, in 2018, and the Thomson Citation Laureate Award from Thompson Reuters. In 2021, Kalyanmoy Deb was named University Distinguished Professor by the MSU Board of Trustees. The prestigious title is in recognition of their exceptional teaching abilities, prominent record of public service, and scholarly, creative, and artistic achievements.