A Feature Rich Distance-Based Many-Objective Visualisable Test Problem Generator

Jonathan E. Fieldsend*
University of Exeter
Department of Computer Science
EX4 4QF, UK
J.E.Fieldsend@exeter.ac.uk

Tinkle Chugh
University of Exeter
Department of Computer Science
EX4 4QF, UK
T.Chugh@exeter.ac.uk

Richard Allmendinger
The University of Manchester
Alliance Manchester Business School
M15 6PB, UK
richard.allmendinger@manchester.ac.uk

Kaisa Miettinen
University of Jyvaskyla
Faculty of Information Technology, P.O. Box 35 (Agora)
FI-40014 University of Jyvaskyla, Finland
kaisa.miettinen@jyu.fi

ABSTRACT

In optimiser analysis and design it is informative to visualise how a search point/population moves through the design space over time. Visualisable distance-based many-objective optimisation problems have been developed whose design space is in two-dimensions with arbitrarily many objective dimensions. Previous work has shown how disconnected Pareto sets may be formed, how problems can be projected to and from arbitrarily many design dimensions, and how dominance resistant regions of design space may be defined. Most recently, a test suite has been proposed using distances to lines rather than points. However, active use of visualisable problems has been limited. This may be because the type of problem characteristics available has been relatively limited compared to many practical problems (and non-visualisable problem suites). Here we introduce the mechanisms required to embed several widely seen problem characteristics in the existing problem framework. These include variable density of solutions in objective space, landscape discontinuities, varying objective ranges, neutrality, and non-identical disconnected Pareto set regions. Furthermore, we provide an automatic problem generator (as opposed to hand-tuned problem definitions). The flexibility of the problem generator is demonstrated by analysing the performance of popular optimisers on a range of sampled instances.

KEYWORDS

multi-objective test problems; evolutionary optimisation; benchmarking; test suite; visualisation

*Corresponding author

1 INTRODUCTION

The ability to *see* how a multi/many-objective optimisation algorithm is progressing is often a vital aspect of algorithm design and analysis. In terms of progress quality, this may be from a convergence plot to some indicator (e.g. hypervolume [42] or inverted generational distance [4]). However, visualising how the search population moves/converges to the Pareto set and other attractors, in order to understand e.g. search bias, is more difficult.

Parallel coordinate plots and heatmap visualisations coupled with and without dimensionality reduction methods (e.g. principal component analysis) are widely used to show the distribution of solutions, but as the number of dimensions (in either space) increases, picking out relationships quickly is more difficult. The set of alternative solutions to compare also tends to grow with the number of objectives K. Specialised scatterplot visualisation approaches are lossy in general due their data compression from a higher number of dimensions into two or three dimensions used to visualise the data [9, 21, 33]. Alternatively, if pair-wise plots are used, then the number of plots required becomes rapidly overwhelming (as $(K^2 - K)/2$ plots are needed).

The evolutionary multi-objective optimisation (EMO) community has proposed a range of test problems over the years to validate an algorithm's ability to deal with different problem characteristics. For instance, prominent representatives of discrete problems include multi-objective knapsack [42] and NK-landscapes [1], while commonly used permutation problems include the multi-objective travelling salesman problem [5] and flowshop scheduling [18]. Arguably, the largest number of test problems have been proposed for the continuous domain including test suites such as DTLZ [7] and WFG [11], and, more recently, many-objective test problems [3, 30]

and the BBOB problems [32]. Although these multi/many-objective problems allow the user to adjust various problem features, such as the dimension of the decision and/or the objective space and aspects of the Pareto front shape, the issue of being unable to visualise the movement of the search population in its native domain remains, as also pointed out in a recent review of existing scalable multi-objective test suites [38].

Distance-based multi- and many-objective optimisation problems, which were initially popularised in [19, 20] for visualisation, sidestep these issues by creating problems that are *themselves* inherently visualisable. They formulate problems which can have arbitrarily many objectives, but whose design space natively lives in two-dimensions, where the Pareto set is easy to identify by eye. Subsequent work extended these to include (i) arbitrarily large decision spaces that could be projected back to the 2D visualisation space [26], (ii) disconnected Pareto sets of the same [14] or different shapes [12], (iii) non-identical disconnected Pareto sets [12], (iv) dominance resistance regions [8], and (v) local fronts [25]. Distance-based problems have been used in a number of empirical studies (e.g. [13, 14, 24, 31]) in order to visualise the *distribution* of candidate solutions maintained by multi- and many-objective optimisers during their search, and their effectiveness/bias in locating the Pareto set of solutions. A line-based-distance test suite was introduced in [22, 23], though most work remains on point-based formulations, which we are concerned with here (the extension of line-based distance test problems with the problem features proposed here is part of future research).

In the distance-based formulation (also referred to as a Pareto-box formulation), a putative solution is a point in the plane, and its performance on each objective is calculated as its distance to a point in that space. Here we use the acronym DBMOPP as shorthand for distance-based multi/many-objective point problems. Broadly speaking, this work advances the current state-of-the-art in DBMOPP in terms of additional problem characteristics and developing an automatic problem instance generator for creating problem instances with the existing and new characteristics. The specific contributions are:

(1) The ability to vary the density of solutions that lie in different regions of the Pareto set — thus varying the density across the Pareto front.
(2) An alternative approach to create disconnected Pareto sets which map to different regions of the Pareto front.
(3) The ability to have discontinuities in the objective functions.
(4) The ability to have the objectives on markedly different scales, with different minimum values.
(5) A generator to supply well-formed problem instances with arbitrarily many objectives, design variables, local fronts, disconnected fronts, dominance resistance regions and varying projection densities — all visualisable in the plane.
(6) Demonstration of the test problem generator by visualising and analysing the performance of some popular optimisers on a sampled set of problem instances (provided by the proposed generator).

The rest of this work proceeds as follows. The next section provides a formal definition of multi/many-objective optimisation problems and recaps the concepts of Pareto optimality and dominance followed by a description of existing features in the DBMOPP literature. Section 3 extends the existing DBMOPP framework by introducing a set of tunable new problem characteristics. Section 4 then introduces a problem generator capable of automatically creating DBMOPP problem instances that feature desired (existing and newly proposed) characteristics. Results of numerical experiments on a sampled set of problem instances with different characteristics are presented in Section 5. Finally, we conclude and discuss future research directions in Section 6.

2 EXISTING VISUALISABLE DISTANCE-BASED TEST PROBLEMS

2.1 Problem definition

Before outlining test problem properties, it is useful to formally define Pareto optimality and dominance. For multi/many-objective optimisation problems, without loss of generality, we seek to simultaneously minimise K objectives: $f_k(\mathbf{x})$, $k = 1, \ldots, K$, where each objective depends upon a vector $\mathbf{x} = (x_1, \ldots, x_N)$ of N design or decision variables. The variables may also be subject to equality and inequality constraints. Such constraints define $\mathcal{X} \subseteq \mathbb{R}^N$, a feasible design space. Related to this is $\mathcal{Y}$, the objective space image of $\mathcal{X}$ (the feasible objective space). When there is more than one objective to be minimised, solutions may exist for which performance on one objective cannot be improved without reducing performance on at least one other. Such solutions are said to be *Pareto optimal*. The set of all Pareto optimal solutions is said to form the *Pareto set* $\mathcal{P}$, whose image in the objective space is known as the *Pareto front* $\mathcal{F}$. Identifying such solutions relies on Pareto *dominance*. A feasible decision vector $\mathbf{x}$ is said to dominate another $\mathbf{x}'$ iff

$$f_k(\mathbf{x}) \leq f_k(\mathbf{x}') \quad \text{for all} \quad k = 1, \ldots, K \quad \text{and} \quad \mathbf{f}(\mathbf{x}) \neq \mathbf{f}(\mathbf{x}'). \quad (1)$$

This is often simply denoted as $\mathbf{x} \prec \mathbf{x}'$ rather than $\mathbf{f}(\mathbf{x}) \prec \mathbf{f}(\mathbf{x}')$.

In standard visualisable distance-based test problems, we have $\mathcal{X} \subseteq \mathbb{R}^2$. For point-based formulations in this domain, there are K sets of vectors defined, where the kth set, $V_k = \{\mathbf{v}_1, \ldots, \mathbf{v}_{m_k}\}$, determines the quality of a putative design vector $\mathbf{x} \in \mathcal{X}$, on the kth objective. This is typically calculated as

$$f_k(\mathbf{x}) = \min_{\mathbf{v} \in V_k} (\texttt{dist}(\mathbf{x}, \mathbf{v})).$$

Note as m_k is the number of elements of V_k, which depends on k, it is legal for $|V_i| \neq |V_j|$, but $|V_i| \geq 1$ for all i. The function $\texttt{dist}(\mathbf{x}, \mathbf{v})$ typically returns the Euclidean distance between $\mathbf{x}$ and $\mathbf{v}$. An alternative distance metric, not considered in this paper, is the Manhattan distance [37, 40].

Let us consider the simplest distance-based problem formulation using points, where $|V_k| = 1$ for all k. This means that there is a single connected Pareto set, and no additional locally Pareto optimal regions, as illustrated in Figure 1. We could directly set the elements of V_k, meaning $2 \times K$ parameters to fix to define a problem. A more attractive representation however is to use a centre (2 coordinate values), a circle radius (r), and an angle for each objective minimising vector, making $3 + K$ parameters to fix when initialising a test problem. This has the advantage of having the same or fewer parameters for all problems with $K > 2$ objectives compared to directly choosing the point coordinates. Additionally, the polygon defined by the points generated in this fashion will

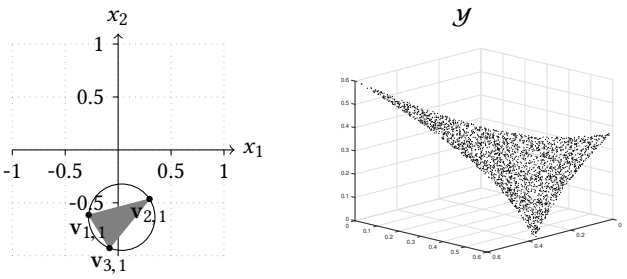

Figure 1: A problem with three objectives, $V_k = \{\mathbf{v}_{k,i}\}$, $|V_k| = 1$. *Left*: The three locations in $\mathcal{X}$, which lie on the circumference of the black circle, determine the objective value minima. They describe a three-sided polygonal Pareto set (coloured grey). *Right*: Samples on the corresponding Pareto front generated by Monte Carlo sampling the Pareto set.

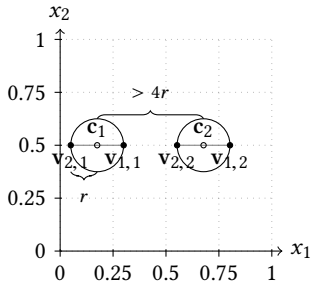

Figure 2: Requirement of $4r$ separation. Two Pareto sets (grey lines) defined by centres $\mathbf{c}_1$ and $\mathbf{c}_2$. If the centres were $\leq 4r$ apart, the Pareto set would be induced between the two regions, as $\mathbf{v}_{1,1}$ would be closer to $\mathbf{v}_{2,2}$ than $\mathbf{v}_{2,1}$.

always result in a well-formed Pareto set (a convex hull formed from them will have every element on its perimeter). We use this convention here in our generator to illustrate how we achieve the various feature additions to the DBMOPP framework.

2.2 Test problem sampling

Work up until now has hand-tuned DBMOPP problems. One of our contributions here is the introduction of a generator to automatically construct DBMOPPs with a range of properties, allowing empirical analysis based on test problem sampling, supporting the assessment of *generalisable* results, rather than those tuned to a particular suite of problems (see e.g. [2]). This is a valuable provision alongside those other generators available for different problem forms (e.g. multi-modal problems [29], multi-objective NK landscapes [36] and discrete optimisation problems [34]).

2.3 Existing features in the DBMOPP literature

Here we briefly describe the existing features enabled in DBMOPP from the literature.

2.3.1 Disconnected Pareto sets. Where $|V_k| > 1$ for all k, one can generate a disconnected Pareto set of solutions (as long as the relative positions of the groups of points defining each Pareto set are kept the same) [14]. We denote the jth region containing Pareto optimal designs as $\mathcal{R}_j$. This is relatively easy to achieve given the proposed representation, as the angles and radii can be replicated across all regions, and only the centres need varying. One must ensure that the distance between the centres is always sufficient to prevent Pareto set locations being formed *between* different point groupings. A minimum centre distance of $> 4r$ will always ensure this, even if the regions $\mathcal{R}_j$ are rotated with respect to each other. This is illustrated in Figure 2. For c disconnected set regions, this results in $1 + K + c \times 2$ parameters to fix. See Figure 3 for an illustrations of problem instances with $K = 3$ and $K = 7$ objectives.

2.3.2 Arbitrarily large design spaces. The original 2D design space can be projected into arbitrarily many dimensions via two orthogonal vectors forming a basis [26], generating a new design space

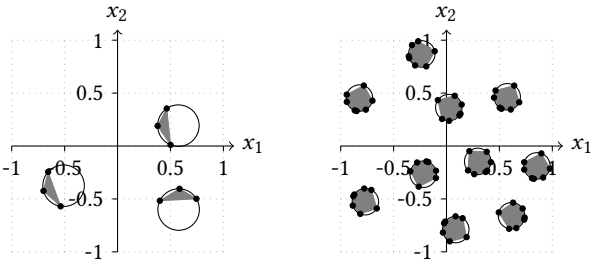

Figure 3: Illustrations of disconnected Pareto sets. *Left*: A $K = 3$ problem with three disconnected regions $\mathcal{R}_j$. *Right*: A $K = 7$ problem with 10 disconnected regions $\mathcal{R}_j$.

$\mathcal{Z} \in \mathbb{R}^N$, $N > 2$. Designs $\mathbf{z}$ from this larger space can be mapped to a corresponding $\mathbf{x}$ using the orthogonal projection vectors, the basis $(\boldsymbol{\pi}_1, \boldsymbol{\pi}_2)$. Subsequently, $\mathbf{x}$ can be evaluated and visualised:

$$\mathbf{x} = \frac{(\mathbf{z} \cdot \boldsymbol{\pi}_1)}{||\boldsymbol{\pi}_1||} \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \frac{(\mathbf{z} \cdot \boldsymbol{\pi}_2)}{||\boldsymbol{\pi}_2||} \begin{pmatrix} 0 \\ 1 \end{pmatrix}.$$

It is possible to have a single 2D space with multiple regions $\mathcal{R}_j$ projected via two orthogonal vectors, but it is also possible to have multiple *different* 2D spaces, projected with *different* orthogonal vector pairs of the same dimension and evaluate $\mathbf{z}$ using each of these projections. This allows the different regions $\mathcal{R}_j$ to be oriented differently in $\mathcal{Z}$ (and be more distant than in the single projection case) [26].

2.3.3 Non-identical disconnected Pareto sets. It is illustrated in [12] how non-identical Pareto set regions may be formed via positioning points to describe identical convex polygons, but swapping positions of points minimising each objective in each. This does, however, have the effect that the Pareto set is potentially a non-convex sub-region of the polygon. Another disconnected Pareto set is illustrated via a map based problem in [12], with multiple locations (railway stations, schools, etc.) defining the minimising locations. This is an excellent example of a real-world problem of the same form, but for arbitrary test problem designs it is less advantageous. Here we would like to control a number of other problem

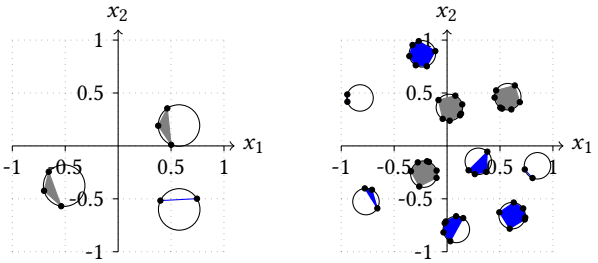

Figure 4: Illustrations of dominance resistance regions (coloured blue) in DBMOPP instances. *Left*: A problem with $K = 3$ and two disconnected regions $\mathcal{R}_j$ and one dominance resistance region. *Right*: A problem with $K = 7$, three disconnected regions $\mathcal{R}_j$ and seven dominance resistance regions.

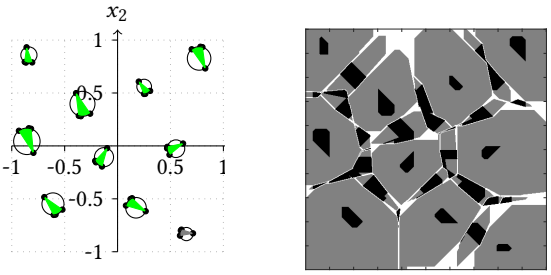

Figure 5: Illustration of local fronts. *Left*: A problem with $K = 4$, one global Pareto set and nine local Pareto fronts (in green). *Right*: Local dominance landscape approximated by sampling $\mathcal{X}$ on a 500×500 grid.

properties when automatically generating problem instances, ensure that instances are viable, and Pareto sets are easy to identify *a priori*. We detail the approach we use in our generator in Section 3.2.

2.3.4 Dominance resistance regions. The usual generation of a DBMOPP results in all solutions which minimise any *individual* objective f_k also being Pareto optimal. In [8], region constructions were introduced which could overcome this limitation and supply designs which were dominance resistant [10] (i.e. dominated but weakly Pareto optimal [27] when compared to Pareto set members). These regions had points whose relative positions matched those in the Pareto set, but which are described by at most $K-1$ of the points used to define a region $\mathcal{R}_j$, meaning each solution in a dominance resistance region is dominated by at least one member of the Pareto set. Illustrations are provided in Figure 4. Such a formulation means there are (many) locations in $\mathcal{X}$ whose quality may be optimal under one or more f_k, but when evaluated under $\mathbf{f}$ are still located very far from $\mathcal{F}$ — unlike in the standard formulation where all $\mathbf{x}^* = \texttt{argmin}_{\mathbf{x}\in\mathcal{X}} f_k(\mathbf{x})$ are in $\mathcal{P}$, $\forall k$.

2.3.5 Local fronts. Local fronts in multi-objective problems act much like local optima in uni-objective problems — generating basins of attraction which compete with the Pareto set. These may be easily generated in our framework by using the angles selected for the placement of the objective minima points around the centre in the Pareto set, but applying a larger radius when distributing attractor points for local regions[1]. An illustration is provided in the left panel of Figure 5, with the corresponding *local dominance landscape* shown in the right panel (generated through sampling on a 500×500 grid).

The black regions in the local dominance landscape are comprised of cells in the discretised space, where all eight immediate neighbouring locations (the Moore neighbourhood) are mutually non-dominating with the centre cell (denoting dominance-neutral local optima regions). These may be identified by point-based Pareto hill-climbing [35], but note that a contiguous region of such local optima is not guaranteed to be composed entirely of members that are mutually non-dominating (a local Pareto set), as construction of these relies on a set-based rather than point-based hill-climb (see e.g. [28]). Instead, the black regions describe a locally dominance-neutral region, where all local moves are incomparable from a dominance perspective. Grey regions in the plot are made up of cells which have at least one dominating neighbour (i.e. lie on a dominance hill-climb path, rather than the end of a path), and all dominating movement paths from neighbours in grey regions lead to the same local optima region. As such, the grey regions denote those basin components which lead to the same dominance-neutral attractor. White regions are comprised of cells whose neighbours lead to multiple different attractor regions (and therefore denote boundary regions/saddle-points).

Note the complex interactions in the landscape in the right panel of Figure 5. The local dominance-neutral regions include the Pareto set and the regions denoting specified local fronts from the left panel, but also additional dominance-neutral regions lying between these have been induced by the attractor points. These generally have much smaller basins (and in some cases no basin at all). As noted, the dominance-neutral regions may be larger than the corresponding region illustrated in the left panel of Figure 5 — this is because the dominance-neutrality is *local* to the neighbourhood of each cell, rather than calculated with respect to *every* member of the region (denoting the landscape observed by a local greedy dominance-based hill-climber).

2.3.6 Pareto front shape. It is worth noting that due to their construction, the front shape of DBMOPP problems result in a $\mathcal{F}$ whose members at most minimise a *single* objective. For example, in $k = 3$ problems $\mathcal{F}$ has an 'inverted triangle' shape. Recent work has highlighted that such shapes can cause particular problems for both decomposition-based algorithms [17] and hypervolume-based algorithms [15], and can also cause issues selecting an reference point for the hypervolume calculation [16].

3 NEW AND ENHANCED DBMOPP FEATURES

We now describe new features (in the case of non-identical disconnected Pareto sets, an enhanced feature) we have added to the existing DBMOPP framework and which we have implemented as a problem instance generator alongside those previously described.

[1]Note, for computational reasons a problem instance generator must pre-calculate the maximum local front radius.

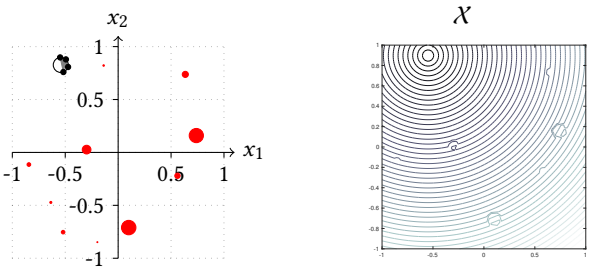

Figure 6: Penalty locations and their effect on an objective landscape. *Left*: A problem with $K = 4$ and 10 penalty regions (shown with filled red circles). *Right*: f_1 quality landscape.

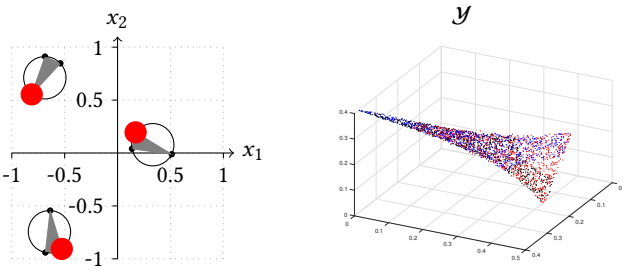

Figure 7: Illustrations of penalty locations generating non-identical disconnected Pareto sets. *Left*: Penalty regions intersect the three regions $\mathcal{R}_i$ in different areas. *Right*: $\mathcal{X}$ is sampled and evaluated under $\mathbf{f}$, the non-dominated subset is shown, with the three colours represented in the three different regions $\mathcal{R}_j$ responsible for the front members.

3.1 Discontinuous objective surfaces

The use of sets V in DBMOPP construction results in smooth objective landscapes. We propose here the introduction of discontinuities. They can be introduced via *penalty regions* centred on a penalty location $\mathbf{p}$. These may be used to apply a fixed or varying non-zero penalty to one or more objective values for all locations within the region. This induces a discontinuity in the landscape of those objectives affected by the penalty at all locations that lie on the perimeter of the penalty region, illustrated in Figure 6.

Here we use circular penalty regions defined by a centre (location) and radius. However, arbitrary polygonal shapes to define the penalty regions could also be used (at the cost of additional parameters and interior checking). Where a penalty region intersects a region $\mathcal{R}_j$ or lies entirely within one, additional features are induced, which we now detail.

3.2 Non-identical disconnected Pareto sets

Under most current DBMOPP formulations, the image of each region $\mathcal{R}_j$ in $\mathcal{X}$ under $\mathbf{f}$ describes the entire Pareto front. However, if we place penalty regions which intersect with a region $\mathcal{R}_j$ (or which lie entirely within $\mathcal{R}_j$), whose penalty is sufficient to make points within the penalty region dominated by elements of $\mathcal{X}$, we can effectively 'cut-out' a chunk of that region $\mathcal{R}_j$.

Furthermore, if penalty regions are placed in different $\mathcal{R}_j$ asymmetrically, then each $\mathcal{R}_j$ will map to different parts of the Pareto front (depending on construction, these may be partially overlapping, or non-intersecting). An illustration if this is provided in Figure 7. Given the penalty locations, some objective combinations are only available in one of the regions $\mathcal{R}_j$ (e.g. the right-hand edge of $\mathcal{F}$ in the middle — as this area is removed from two of the three regions $\mathcal{R}_j$) and some in different pairs of $\mathcal{R}_j$ (i.e. the corner regions, where one of the three regions $\mathcal{R}_j$ each have a penalty centred). Some optimal objective combinations reside in all three disconnected sets (i.e. the central portion of the front).

3.3 Varying solution density in Pareto sets

Varying the relative lengths of the orthogonal projection vectors used to generate arbitrarily large design spaces allows us to vary the density of the solutions mapped back to the 2D representation in $\mathcal{X}$. This can in turn make some regions $\mathcal{R}_j$ and regions of the Pareto front more difficult to attain than others.[2]

3.4 Varying objective scales

In standard formulations of DBMOPP, the range of each objective does not vary greatly, and the minimum of all objectives is 0. We can, however, shift the objective ranges to be arbitrarily wide/narrow, with arbitrary maxima and minima via a multiplication and shift term $f_k^{rescaled}(\mathbf{x}) = a_k + b_k \times f_k(\mathbf{x})$.

3.5 Neutrality

Neutral (flat) regions of the objective/domination landscape can be generated using the penalty region approach detailed in Section 3.1, where instead of an additive/multiplicative penalty on the objective(s) associated with designs in the region, a constant value is used to replace objective values. This has the effect of making all design vectors in the region express identical objective values for the set of objectives affected. Neutrality is common in combinatorial spaces, but can also exist in continuous spaces, for instance the labour cost/time of manufacture may not change *at all* between similar engineering designs.

4 PROBLEM INSTANCE GENERATOR

Given the wealth of features described above extending the existing DBMOPP framework, which can be incorporated in a DBMOPP test problem instance, the question is how to generate a problem automatically and correctly, ensuring the desired properties are all present (and to the correct degree). We solve this here by observing that $\mathcal{X}$ may be partitioned into areas concerned with providing examples of each of the various properties desired. These are largely determined by sets of points defining the different region types (Pareto sets, dominance resistance regions, penalty regions, non-identical disconnected Pareto sets including penalty regions, local fronts, etc.). Algorithm 1 outlines the procedure at a high-level.[3]

[2] A non-linear transform to the f_k value could also be applied (i.e. taking the natural logarithm) to vary the density of mapping from the design space to the objective space.
[3] A MATLAB implementation of the generator and supporting functions to plot the regions in 2D, plot dominance landscape, and create the set of TikZ commands in

Algorithm 1 DPMOPP-generator

function GENERATOR($K, n_{\mathcal{R}}, n_l, n_d, n_{rr}$, *non_identical*, *vary_scales*, *random_seed*)
  set_seed(*random_seed*) ▷ Ensure instance reproducibility
  $V_k := \emptyset \; \forall k$ ▷ Empty set of objective minima coordinates
  $(P_k, R_k) := (\emptyset, \emptyset) \; \forall k$ ▷ Empty penalty locations and radii
  $C := \emptyset$ ▷ Empty set of region centres
  $n := n_{\mathcal{R}} + n_l + n_d$ ▷ Total number of centred regions
  $r^{max} := \mathcal{U}(0, 1/(1 + \lceil 2\sqrt{(n)} \rceil))$ ▷ Draw r^{max}
  $r^{Pareto} := r^{max}$ ▷ Default radius of Pareto set regions
  $\mathbf{a} := [\mathbf{0}]_{K \times 1}$ ▷ Additive constants for objective rescaling
  $\mathbf{b} := [\mathbf{1}]_{K \times 1}$ ▷ Multiplicative constants for rescaling
  if $n_l > 0$ **then** ▷ If local fronts are generated
    $(V, C, r^{Pareto}) :=$ place_local_fronts(V, C, n_l, r^{max})
  end if
  $(V, C) :=$ place_Pareto_set$(V, C, n_{\mathcal{R}}, r^{Pareto})$
  if $n_r r > 0$ **then** ▷ If dominance resistance regions needed
    $(V, C) :=$ place_dom_resistance$(V, C, n_{rr}, r^{Pareto})$
  end if
  if *non_identical* = true **then** ▷ $|\mathcal{R}_i|$ vary
    $(P, R) :=$ modfiy_regions_with_assym_pen(V, C, P, R)
  end if
  if $n_d > 0$ **then** ▷ If discontinuities outside of $\mathcal{R}_i$
    $(P, R, C) :=$ place_discontinuities(V, C, n_l, r^{max})
  end if
  if *vary_scales* = true **then** ▷ Modify objectives ranges
    $(\mathbf{a}, \mathbf{b}) :=$ sample_scaling_constants(K)
  end if
  return $(V, P, R, \mathbf{a}, \mathbf{b})$ ▷ (Points, penalties, penalty radii, objectives rescaling)
end function

The properties of problems generated by Algorithm 1 are directly verifiable. For instance, finely discretising the space and commencing a Pareto Local Search [28] from a single location in each of the local front sets will verify their existence and location matches that described in the problem state variables (i.e. $V, P, R, \mathbf{a}$ and $\mathbf{b}$). The existence of dominance resistance regions can be verified by simply sampling from each corresponding region and observing there are solutions with minimal $f_k(\mathbf{x})$ which are dominated, etc.

4.1 Randomly placing region centres

We allocate the centres defining each of the regions at random, but subject to lying at least $4r$ from the closest next region for all attractor regions. Here, r is the *largest* radius employed by any individual region. Additionally, all region centres must be at least r from the domain boundary. We employ a Monte Carlo circle placement with rejection sampling for this. For non-attractor regions, i.e. penalty regions forming discontinuities or neutral regions in the objective landscapes, these may be placed immediately adjacent to attractor regions, as they cannot induce Pareto optimal regions if placed too close (unlike the other region types).

LaTeX to generate illustrations (as in many of the subplots here) are available at https://github.com/fieldsend.

The region radius r cannot be set arbitrarily as, depending on the number of circles being fit into a bounded $\mathcal{X}$, legal placement for all may be impossible. Given n attractor regions and n' non-attractor regions to be placed, and our domain boundaries $(-1, +1)$, we can calculate the maximum possible value this could take, r^{max}, *a priori*. This corresponds to packing in all $n + n'$ regions of the two distinct types in the bounded area in a regular grid (with four non-attractor regions having the same minimum area requirements as one attractor region). As such, we have $r^{max} = 1/(2 + \lceil \sqrt{(n + \frac{n'}{4})} \rceil)$, and for a particular problem instance $r \sim \mathcal{U}(0, r^{max})$.

In reality, the legal Monte Carlo allocation of all $n + n'$ centres with $r = r^{max}$ is vanishingly small, as it essentially requires the random generation of $n + n'$ points on a regular grid. Subsequently, for each instance, r is drawn from the uniform distribution $\mathcal{U}(0, r^{max})$, but if a legal set of centres is not drawn via Monte Carlo sampling sufficiently quickly, a shrinkage factor of 0.95 is recursively multiplied to r until a legal set can be generated.[4]

5 ILLUSTRATION ON SOME POPULAR OPTIMISERS

In this section, we illustrate the search behaviour of three different optimisers on problems with different features from our DBMOPP generator. We used NSGA-II [6], MOEA/D [39] and IBEA [41] as a dominance, decomposition and indicator based EMO algorithm, respectively. In the experiments, a broad range of dimensions and other features of the problems tested were covered:

Problem characteristics:

(1) Number of objectives: 2, 4 and 10.
(2) Number of design variables: 2, 5 and 100.[5]
(3) Number of disconnected Pareto sets: 1 and 5.
(4) Number of local fronts: 0, 5 and 10.
(5) Number of dominance resistance regions: 0, 5 and 10.
(6) Varying objectives scales.[6]

Optimiser characteristics:

(1) Population size: 100 for two objectives, 120 for four objectives and 275 for 10 objectives.
(2) Crossover: Simulated binary with distribution index of 20 and probability of 0.8.
(3) Mutation: Polynomial with distribution index of 20 and probability of 1/number of design variables.
(4) Number of independent runs: 21.
(5) Maximum number of generations: 500.

We also used a random search algorithm as a baseline to compare against the three optimisers. Trace generation plots (or evolution of solutions with generations) of different optimisers from the run with median hypervolume values are shown in Figure 8. In the figure, rows represent the features of different problems from the DBMOPP generator, which are detailed in parentheses on the left in the figure in the order [number of objectives, number of design

[4] In practise, we found that for all $r < 0.7r^{max}$ we could find a legal set immediately, without recourse to shrinkage.
[5] Here we generate orthogonal vectors with elements exclusively ones or zeros, and use the same orthogonal pair for all region mappings.
[6] Here an shift changing each minima for each objective function generated is draw randomly from the range [−100, 100], and the objective range itself is rescaled by a multiplier randomly drawn from the range [1, 1000].

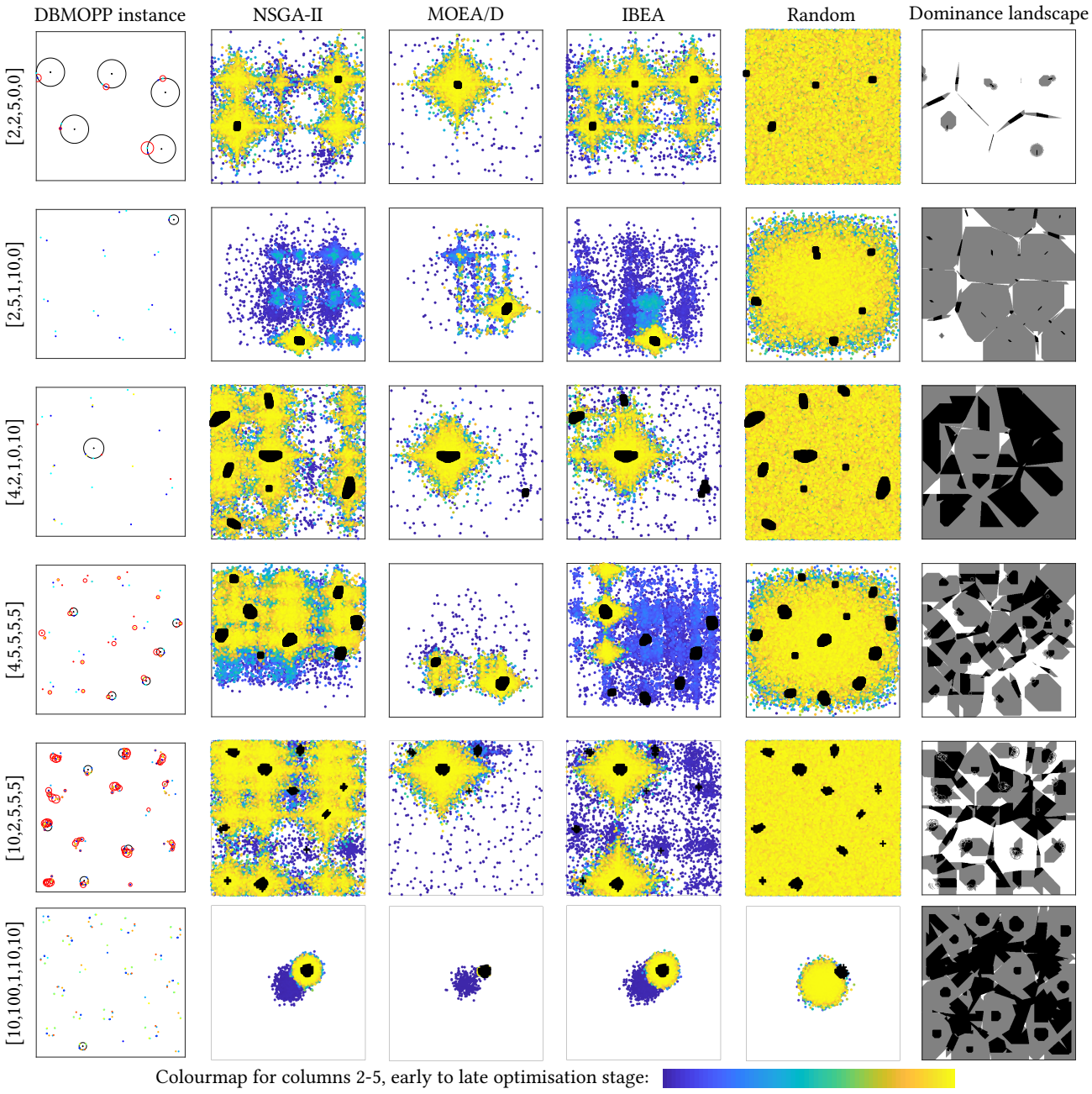

Figure 8: Trace generation plots of solutions with different optimisers of the run with median hypervolume value. The features of different problems are mentioned in parentheses in the order [number of objectives, number of design variables, number of disconnected Pareto sets, number of local fronts and number of dominance resistance regions].

variables, number of disconnected Pareto sets, number of local fronts and number of dominance resistance regions].

In Figure 8, for each row, the first column represents the problem features. The second, third and fourth columns represent the trace generation plots of the three EMO algorithms: NSGA-II, MOEA/D and IBEA, respectively. The fifth column corresponds to random search, and the sixth is the *local dominance landscape*. In the first column, attractor locations are represented by points, and coloured according to the objective they minimise. Points associated with a Pareto optimal region are plotted on the circumference of a black circle with a black symbol in the centre, allowing the region R_j to be identified. Where there is no black circle, points are associated

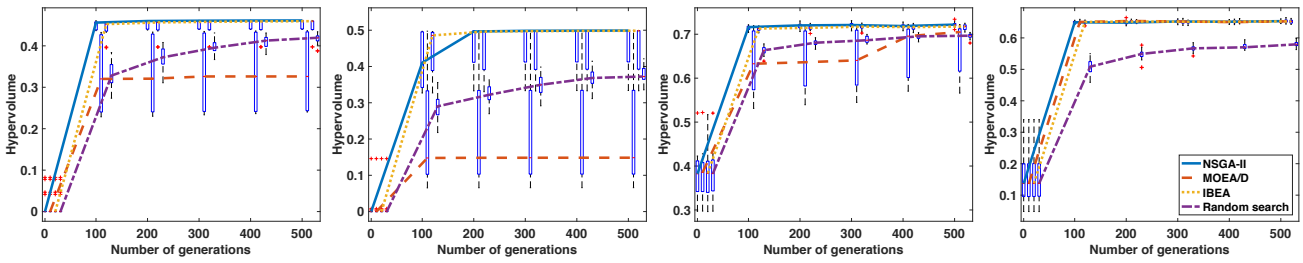

Figure 9: Hypervolume values with number of generations of problems corresponding to rows 1, 2, 5 and 6 in Figure 8.

with local fronts or dominance resistance regions. Red circles (e.g. in the top left panel of Figure 8) denote penalty regions.

In the trace generation plots, the shading transitions from blue to yellow as the optimisation advances. The black regions (in columns 2-5) represent the final non-dominated solutions obtained after 500 generations. The axes are the first and the second element of the design vector, respectively. A design vector with more than two elements (rows 3-6) is projected to two dimensions using formula in Section 2.3.2. The corresponding normalized hypervolume plots as a function of the generation counter (abscissa) are shown in Figure 9. In the figure, the box plots are shown at different numbers of generations (0, 100, 200, 300, 400 and 500). For ease of readability, a small gap is added between the boxplots of the different optimisers in each of the hypervolume plots.

The trace generation and hypervolume plots clearly show the effect of different problems features. For a low dimensional problem in the design space with many disconnected Pareto sets as in row 1, all three optimisers failed to find all disconnected Pareto sets. On the other hand, random search found four out of five disconnected Pareto sets. This is because all three optimisers could not preserve diversity in the design space and converged to few disconnected Pareto sets (while random search inherently generates diverse solutions). In the second row, where the problem has many local Pareto fronts and no disconnected Pareto sets, all three optimisers (and random search) failed to converge to global solutions (located in the top right corner of the design space), getting stuck at a local optima. In row 3, the problem has many dominance resistance regions and four objectives, all optimisers including random search converged to the global optima. This is because the problem is relatively easy possessing a fully connected Pareto set and no local fronts.

In row 4, where the problem has many disconnected Pareto sets, local fronts and dominance resistance regions, all optimisers except MOEA/D performed well and found all disconnected Pareto subsets (NSGA-II found four out of five). MOEA/D relies on the reference vectors (or weight vectors) to find a good distribution of solutions in the objective space and decomposes the problem into sub-problems. In DBMOPP problems as in row 4, the global front is deceptively multi-modal [11], which poses a difficulty for MOEA/D in escaping local optima when solving sub-problems. Compared to row 4, in row 5 we have more objectives but fewer design variables. Here, all optimisers (including random search) except MOEA/D found the global optima. This is because there were few design variables and NSGA-II and IBEA were able to generate a diverse set of solutions. In row 6, with 10 objectives and 100 design variables, no disconnected Pareto subsets, 10 local fronts and 10 dominance resistance regions, none of the optimisers nor random search found the global optimum. This was the most difficult problem used in this test setting with all of the optimisers failing to generate sufficiently diverse solutions; this pattern for large scale problems was also observed in [26]. Note that the solutions in row 6 are not distributed well in the design space (even for random search) because of the projection of the originally 100-dimensional design space to 2 dimensions.

The behaviour of different optimisers can also be seen in the hypervolume plots in Figure 9. As we observed from the trace generation plots, most of the optimisers could not find the global optima in many instances and in the hypervolume plots, most of the optimisers converged very fast. This implies that in many of the cases, the optimisers converged to local optima and failed to get out of them. Another interesting observation is that random search performed better or equivalent to the optimisers on the low dimensional problems (first two plots from the top) and worse in high dimensional problems (plots three and four from the top). The rather poor performance of MOEA/D, which we observed and discussed above, is confirmed in the hypervolume plots.[7]

6 CONCLUSIONS

We have presented a number of extensions to the existing DBMOPP framework, providing a rich feature set comparable in range to existing (non-visualisable) problem suites. We have developed a problem instance generator for this class of problem, and illustrated the performance of a few widely used optimisers on some example instances generated by it.

In future work, we intend to develop and integrate additional features into the generator, including (i) more complex penalty geometries, allowing more complicated front surfaces (arbitrary polygons); (ii) disconnected Pareto fronts and (iii) dynamic problem variants. We also intend to undertake a rigorous testing and analysis of popular optimisers on a range of problem instances.

ACKNOWLEDGMENTS

This work was supported by the Engineering and Physical Sciences Research Council [grant number EP/N017846/1] and the Natural Environment Research Council [grant number NE/P017436/1]. This research is related to the thematic research area DEMO (jyu.fi/demo) of the University of Jyväskylä.

[7] The maximum of non-dominated objective function values of all 21 runs from all optimisers including random search in solving a problem instance was used as the reference point in calculating the hypervolume.

REFERENCES

[1] H. E. Aguirre and K. Tanaka. 2007. Working principles, behavior, and performance of MOEAs on MNK-landscapes. *European Journal of Operational Research* 181, 3 (2007), 1670–1690.

[2] T. Bartz-Beielstein. 2015. How to create generalizable results. In *Springer Handbook of Computational Intelligence*, J. Kacprzyk and W. Pedrycz (Eds.). Springer, Berlin, Heidelberg, 1127–1142.

[3] Y-M. Cheung, F. Gu, and H-L. Liu. 2016. Objective extraction for many-objective optimization problems: Algorithm and test problems. *IEEE Transactions on Evolutionary Computation* 20, 5 (2016), 755–772.

[4] C. A. C. Coello and M. R. Sierra. 2004. A study of the parallelization of a coevolutionary multi-objective evolutionary algorithm. In *MICAI 2004: Advances in Artificial Intelligence: Third Mexican International Conference on Artificial Intelligence, Proceedings*. Springer, Berlin, Heidelberg, 688–697.

[5] D. W. Corne and J. D. Knowles. 2007. Techniques for highly multiobjective optimisation: some nondominated points are better than others. In *Genetic and Evolutionary Computation Conference, GECCO2007, Proceedings*. ACM, 773–780.

[6] K. Deb, A. Prarap, S. Agarwal, and T. Meyarivan. 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation* 6 (2002), 182–197.

[7] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler. 2005. Scalable test problems for evolutionary multiobjective optimization. In *Evolutionary Multiobjective Optimization: Theoretical Advances and Applications*, A. Abraham and R. Goldberg (Eds.). Springer, 105–145.

[8] J. E. Fieldsend. 2016. Enabling dominance resistance in visualisable distance-based many-objective problems. In *Proceedings of the 2016 on Genetic and Evolutionary Computation Conference Companion*. ACM, 1429–1436.

[9] J. E. Fieldsend and R. M. Everson. 2013. Visualising high-dimensional Pareto relationships in two-dimensional scatterplots. In *Evolutionary Multi-Criterion Optimization, 7th International Conference, Proceedings*, R. Purshouse, P. Fleming, C. Fonseca, S. Greco, and J. Shaw (Eds.). 558–572.

[10] T. Hanne. 1999. On the convergence of multi objective evolutionary algorithms. *European Journal of Operational Research* 117 (1999), 553–564.

[11] S. Huband, P. Hingston, L. Barone, and L. While. 2006. A review of multiobjective test problems and a scalable test problem toolkit. *IEEE Transactions on Evolutionary Computation* 10, 5 (2006), 477–506.

[12] H. Ishibuchi, N. Akedo, and Y. Nojima. 2011. A many-objective test problem for visually examining diversity maintenance behavior in a decision space. In *Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation*. ACM, 649–656.

[13] H. Ishibuchi, N. Akedo, H. Ohyanagi, and Y. Nojima. 2011. Behavior of EMO algorithms on many-objective optimization problems with correlated objectives. In *2011 IEEE Congress on Evolutionary Computation (CEC), Proceedings*. IEEE, 1465–1472.

[14] H. Ishibuchi, Y. Hitotsuyanagi, N. Tsukamoto, and Y. Nojima. 2010. Many-objective test problems to visually examine the behavior of multiobjective evolution in a decision space. In *Parallel Problem Solving from Nature, PPSN XI, 11th International Conference, Proceedings, Part II*, R. Schaefer, C. Cotta, J. Kolodziej, and G. Rudolph (Eds.). Springer, Berlin, Heidelberg, 91–100.

[15] H. Ishibuchi, R. Imada, N. Masuyama, and Y. Nojima. 2018. Use of two reference points in hypervolume-based evolutionary multiobjective optimization algorithms. In *International Conference on Parallel Problem Solving from Nature*. Springer, 384–396.

[16] H. Ishibuchi, R. Imada, Y. Setoguchi, and Y. Nojima. 2018. How to specify a reference point in hypervolume calculation for fair performance comparison. *Evolutionary Computation* 26, 3 (2018), 411–440.

[17] H. Ishibuchi, Y. Setoguchi, H. Masuda, and Y. Nojima. 2017. Performance of decomposition-based many-objective algorithms strongly depends on Pareto front shapes. *IEEE Transactions on Evolutionary Computation* 21, 2 (2017), 169–190.

[18] H. Ishibuchi, T. Yoshida, and T. Murata. 2003. Balance between genetic search and local search in memetic algorithms for multiobjective permutation flowshop scheduling. *IEEE Transactions on Evolutionary Computation* 7, 2 (2003), 204–223.

[19] M. Köppen, R. Vicente-Garcia, and B. Nickolay. 2005. Fuzzy-Pareto-dominance and its application in evolutionary multi-objective optimization. In *Evolutionary Multi-Criterion Optimization, Third International Conference, Proceedings*, C. A. Coello Coello, A. H. Aguirre, and E. Zitzler (Eds.). Springer, Berlin, Heidelberg, 399–412.

[20] M. Köppen and K. Yoshida. 2007. Substitute distance assignments in NSGA-II for handling many-objective optimization problems. In *Evolutionary Multi-Criterion Optimization 4th International Conference, Proceedings*, S. Obayashi, K. Deb, K. Poloni, T. Hiroyasu, and T. Murata (Eds.). Springer, Berlin, Heidelberg, 727–741.

[21] M. Köppen and K. Yoshida. 2007. Visualization of Pareto-sets in evolutionary multi-objective optimization. In *Proceedings of the 7th International Conference on Hybrid Intelligent Systems*. 156–161.

[22] M. Li, C. Grosan, S. Yang, X. Liu, and X. Yao. 2018. Multiline distance minimization: A visualized many-objective test problem suite. *IEEE Transactions on Evolutionary Computation* 22, 1 (2018), 61–78.

[23] M. Li, S. Yang, and X. Liu. 2014. A test problem for visual investigation of high-dimensional multi-objective search. In *2014 IEEE Congress on Evolutionary Computation (CEC), Proceedings*. IEEE, 2140–2147.

[24] M. Li, S. Yang, X. Liu, and R. Shen. 2013. A comparative study on evolutionary algorithms for many-objective optimization. In *Evolutionary Multi-Criterion Optimization, 7th International Conference, Proceedings*, R.C. Purshouse, P.J. Fleming, C.M. Fonseca, S. Greco, and J. Shaw (Eds.). Springer, Berlin, Heidelberg, 216–275.

[25] Y. Liu, H. Ishibuchi, Y. Nojima, N. Masuyama, and K. Shang. 2018. A Double-niched evolutionary algorithm and its behavior on polygon-based problems. In *International Conference on Parallel Problem Solving from Nature*. Springer, 262–273.

[26] H. Masuda, Y. Nojima, and H. Ishibuchi. 2014. Visual examination of the behavior of EMO algorithms for many-objective optimization with many decision variables. In *IEEE Congress on Evolutionary Computation (CEC), Proceedings*. IEEE, 2633–2640.

[27] K. Miettinen. 1999. *Nonlinear Multiobjective Optimization*. Kluwer Academic Publishers, Boston.

[28] L. Paquete, T. Schiavinotto, and T. Stützle. 2007. On local optima in multiobjective combinatorial optimization problems. *Annals of Operations Research* 156, 1 (2007), 83–97. article 83.

[29] J. Rönkkönen, X. Li, V. Kyrki, and J. Lampinen. 2008. A generator for multimodal test functions with multiple global optima. In *Simulated Evolution and Learning, 7th International Conference, Proceedings*. Springer, Berlin, Heidelberg, 239–248.

[30] D. K. Saxena, Q. Zhang, J. A. Duro, and A. Tiwari. 2011. Framework for many-objective test problems with both simple and complicated Pareto-set shapes. In *Evolutionary Multi-Criterion Optimization, 6th International Conference, Proceedings*, R.H.C. Takahashi, K. Deb, E.F. Wanner, and S. Greco (Eds.). Springer, Berlin, Heidelberg, 197–211.

[31] H. K. Singh, A. Isaacs, T. Ray, and W. Smith. 2008. A study on the performance of substitute distance based approaches for evolutionary many objective optimization. In *Simulated Evolution and Learning, 7th International Conference, Proceedings*. Springer, Berlin, Heidelberg, 401–410.

[32] T. Tusar, D. Brockhoff, N. Hansen, and A. Auger. 2016. COCO: The bi-objective black box optimization benchmarking (bbob-biobj) Test Suite. *CoRR* abs/1604.00359 (2016). http://arxiv.org/abs/1604.00359

[33] T. Tušar and B. Filipič. 2015. Visualization of Pareto front approximations in evolutionary multiobjective optimization: A critical review and the prosection method. *IEEE Transactions on Evolutionary Computation* 19, 2 (2015), 225–245.

[34] M. Ullrich, T. Weise, A. Awasthi, and J. Lässig. 2018. A generic problem instance generator for discrete optimization problems. In *Proceedings of the 2018 Genetic and Evolutionary Computation Conference Companion*. ACM, 1761–1768.

[35] S. Verel, A. Liefooghe, L. Jourdan, and C. Dhaenens. 2011. Pareto local optima of multiobjective NK-landscapes with correlated objectives. In *Evolutionary Computation in Combinatorial Optimization, 11th European Conference, Proceedings*, P. Merz and J.-K. Hao (Eds.). Springer, Berlin, Heidelberg, 226–237.

[36] T. Weise, S. Niemczyk, H. Skubch, R. Reichle, and K. Geihs. 2008. A tunable model for multi-objective, epistatic, rugged, and neutral fitness landscapes. In *Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation*. ACM, 795–802.

[37] J. Xu, K. Deb, and A. Gaur. *Identifying the Pareto-optimal solutions for multi-point distance minimization problem in Manhattan space*. Technical Report COIN Report Number 2015018. Michigan State University.

[38] S. Zapotecas-Martínez, C. A. Coello Coello, H. E Aguirre, and K. Tanaka. 2019. A review of features and limitations of existing scalable multi-objective test suites. *IEEE Transactions on Evolutionary Computation* 23, 1 (2019), 130–142.

[39] Q. Zhang and H. Li. 2007. MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on Evolutionary Computation* 11 (2007), 712–731.

[40] H. Zille and S. Mostaghim. 2015. Properties of scalable distance minimization problems using the Manhattan metric. In *2015 IEEE Congress on Evolutionary Computation (CEC), Proceedings*. IEEE, 2875–2882.

[41] E. Zitzler and S. Kunzli. 2004. Indicator-based selection in multiobjective search. In *Proceedings of the Parallel Problem Solving from Nature-PPSN VIII*, X. Yao et al. (Ed.). Springer, 832–842.

[42] E. Zitzler and L. Thiele. 1999. Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach. *IEEE Transactions on Evolutionary Computation* 3, 4 (1999), 257–271.