



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

DOI:

[10.1145/3321707.3321727](https://doi.org/10.1145/3321707.3321727)

Document Version

Accepted author manuscript

[Link to publication record in Manchester Research Explorer](#)

Citation for published version (APA):

Fieldsend, J., Chugh, T., Allmendinger, R., & Miettinen, K. (2019). A Feature Rich Distance-Based Many-Objective Visualisable Test Problem Generator. In *GECCO '19 Proceedings of the Genetic and Evolutionary Computation Conference* <https://doi.org/10.1145/3321707.3321727>

Published in:

GECCO '19 Proceedings of the Genetic and Evolutionary Computation Conference

Citing this paper

Please note that where the full-text provided on Manchester Research Explorer is the Author Accepted Manuscript or Proof version this may differ from the final Published version. If citing, it is advised that you check and use the publisher's definitive version.

General rights

Copyright and moral rights for the publications made accessible in the Research Explorer are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

Takedown policy

If you believe that this document breaches copyright please refer to the University of Manchester's Takedown Procedures [<http://man.ac.uk/04Y6Bo>] or contact uml.scholarlycommunications@manchester.ac.uk providing relevant details, so we can investigate your claim.



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

X*	X
X	X
X	X
XX	XX
X	X
X	X
X	X
XX	XX
X	X

ABSTRACT

In optimiser analysis and design it is **informative** useful to be able to visualise how a search point/population moves through the design space over time. Visualisable distance-based many-objective **optimization** problems have been developed over the last decade to aid this—whose design space is in two-dimensions, but which can have **with** arbitrarily many objective dimensions. Previous work has shown how disconnected Pareto sets may be formed in this framework, how these 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 for this problem class when using distances to lines rather than points. However, uptake in the optimisation community of such **attention** to visualisable problems has been limited when compared to other test problem frameworks/suites. One of the likely reasons for this is **This may be because** the type of problem characteristics available in the distance-based framework has been relatively limited compared to the wide range seen in many practical problems (and non-visualisable problem suites). Here we introduce the mechanisms required to embed a number of **several** widely seen problem characteristics in this **a distance-based** problem framework, which we hope will make the problem framework **much more attractive to the community**. These include local fronts, variable density of solutions in objective space, landscape discontinuities, varying objective ranges, neutrality in objective, space, and non-identical disconnected Pareto set regions. Furthermore we also provide an automatic **problem** generator for this problem class (work until now has been restricted **opposed** to hand-tuned problem definitions). Additionally, example performance results

are provided on some popular optimisers on sampled problem instances.

CCS CONCEPTS

•**Mathematics of computing** → **Evolutionary algorithms**; •**Human-centered computing** → **Visualization**; **Visualization design and evaluation methods**;

KEYWORDS

Multi-objective test problems; evolutionary optimisation

ACM Reference format:

X, X, X, and X. 2019. A Feature Rich Distance-Based Many-Objective Visualisable **Test** Problem Generator. In *Proceedings of the Genetic and Evolutionary Computation Conference 2019, Prague, Czech Republic, July 13–17, 2019 (GECCO '19)*, 8 pages. DOI: 10.1145/nnnnnnnn.nnnnnnnn

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 [33] or inverted generational distance [4]), however visualising how the search population moves/converges in its native domain to the Pareto set and other attractors to understand e.g. search bias is much more difficult.

Widely used, the popular parallel coordinate plot and heatmap visualisations 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 objective number K . Specialised scatterplot visualisation approaches are lossy in general due their data compression from a higher number of dimensions into the two or three dimension used to visualise the data [8, 17, 27]. Alternatively if pair-wise plots are used the number required become rapidly overwhelming (as $K^2 - K$ plots are needed for K objectives).

*Corresponding author

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

GECCO '19, Prague, Czech Republic

© 2019 Copyright held by the owner/author(s). Publication rights licensed to ACM. 978-x-xxxx-xxxx-x/YY/MM...\$15.00

DOI: 10.1145/nnnnnnnn.nnnnnnnn

The Evolutionary Multi-criterion 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 multiobjective knapsack [33] and NK-landscapes [1], while commonly used permutation problems include multiobjective travelling salesman problem [5] and flowshop scheduling [14]. Arguably, the largest number of test problems have been proposed for the continuous domain ones including test suites such as DTLZ [6] and WFG [10], and, more recently, many-objective test problems [3, 25]. Although these multi/many-objective problems allow the user to adjust various problem features, such as dimension of the decision and/or objective space, and aspects of the Pareto-set shape, the issue of being unable to visualise the movement of the search population in its native domain remains.

Distance-based multi- and many-objective problems, which were initially popularised in [15, 16] for visualisation, sidestep these issues — by creating problems which 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 many design dimensions that could be projected back to the 2D visualisation space [21], (ii) disconnected Pareto sets of same [13] or different shapes [11], and (iii) dominance resistance regions [7]. Distance-based problems have been used in a number of empirical studies (e.g. [12, 13, 20, 26]) in order to visualise the *distribution* of designs 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 [18, 19], 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).

To cover in Intro:

- Why do we focus on point-based and not on line-based formulations.

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.

Contributions of this work are:

- (1) the introduction of local fronts into the DBMOPP framework¹.
- (2) 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.
- (3) an alternative approach to create disconnected Pareto sets which map to different regions of the Pareto front;
- (4) the ability to have discontinuities in the f_i ;
- (5) the ability to have the objectives on markedly different scales;

- (6) A generator to supply well-formed problems with arbitrarily many objectives, design variables, local fronts, disconnected fronts, dominance resistance regions and varying projection densities — all visualisable in the plane.

The rest of this work proceeds as follows: [TODO: DETAIL CONTENTS]

2 VISUALISABLE DISTANCE-BASED TEST PROBLEMS

Before outlining the problem properties, it is useful to formally defining Pareto optimality and dominance.

2.1 Pareto optimality

For multi/many-objective optimisation problems, w.l.o.g. we seek to simultaneously minimise K objectives: $f_k(\mathbf{x})$, $d = 1, \dots, K$ where each objective depends upon a vector $\mathbf{x} = (x_1, \dots, x_N)$ of N parameters or decision variables. These parameters may also be subject to equality and inequality constraints. Such constraints define $\mathcal{X} \subseteq \mathbb{R}^N$, the feasible search 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 objective space is known as the *Pareto front*, \mathcal{F} . Identifying such solutions relies on Pareto dominance. A decision vector \mathbf{x} is said to dominate another \mathbf{x}' iff

$$f_d(\mathbf{x}) \leq f_d(\mathbf{x}') \quad \forall d = 1, \dots, D \quad \text{and} \quad \mathbf{f}(\mathbf{x}) \neq \mathbf{f}(\mathbf{x}') \quad (1)$$

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

2.2 Problem definition

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

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

Note the subscript on m_k indicates the number of elements of V_k . It is legal for $|V_i| \neq |V_j|$, but $|V_i| \geq 1 \forall i$. The function $\text{dist}(\mathbf{x}, \mathbf{v})$ typically returns the Euclidean distance between \mathbf{x} and \mathbf{v} . [An alternative distance metric, not considered in this paper, is the Manhattan distance \[31, 32\].](#)

Work up until now has hand-tuned such 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 [24], multi-objective NK landscapes [30] and discrete optimisation problems [28]).

Let us consider the simplest distance-based problem formulation using points, where $|V_i| = 1 \forall i$. This means there is a single connected Pareto set, and there are no other attractors in design

¹This was mentioned as potential future work in [7] but not pursued.

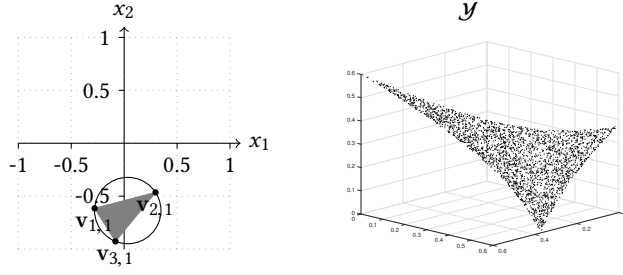


Figure 1: A basic problem with three objectives, $V_k = \{v_{k,i}\}$, $|V_k| = 1$. *Left:* the three locations in \mathcal{X} , which lie on the circumference of the black circle, determine the objective value minima. They describe a three-sided polygonal Pareto set (coloured grey). *Right:* samples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

space providing additional features, as illustrated in Figure 1. We could define our placement options directly as the locations of the vector(s) in the V_i , 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 problem. This has the advantage of having the same or fewer parameters for all $K > 2$ compared to directly choosing the point coordinates. Additionally, the polygon defined by the points generated in this fashion will always result in a well-formed Pareto set (a convex hull formed from them will have every element of point on its perimeter). We use this convention here in our generator, and to illustrate how we achieve the various feature additions to the DBMOPP framework.

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 $|V_i| > 1 \forall i$ it is possible to generate a disconnected Pareto set of solutions (as long as the relative positions of the groups of points defining each Pareto set region are kept the same) [13]. We denote the j th of these regions containing Pareto optimal designs \mathcal{R}_j . This is actually 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. Care does need to be taken to ensure 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 \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 illustration.

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 [21], generating a new design space $\mathcal{Z} \in \mathbb{R}^N, N > 2$. Designs from this larger space, z , can be

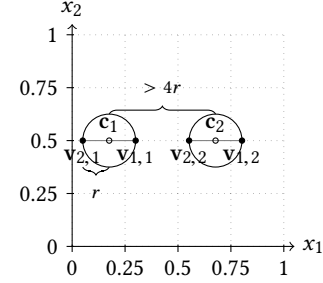


Figure 2: Requirement of $4r$ separation. Two Pareto sets (grey lines) defined by centres c_1 and c_2 . If the centres were $\leq 4r$ apart the Pareto set would be induced between the two regions, as $v_{1,1}$ would be closer to $v_{2,2}$ than $v_{2,1}$.

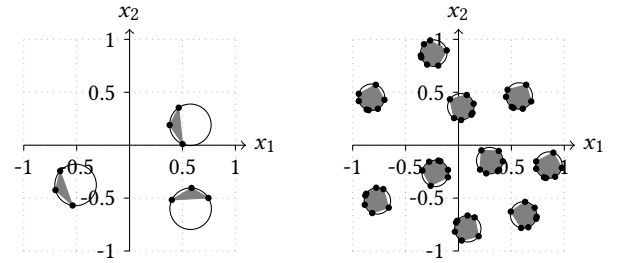


Figure 3: Illustrations of disconnected Pareto sets. *Left:* a $K = 3$ problem with three disconnected \mathcal{R}_i . *Right:* a $K = 7$ problem with 10 disconnected \mathcal{R}_i .

mapped to a corresponding x using the orthogonal projection vectors, the basis (π_1, π_2) , where they can be subsequently evaluated and visualised.

$$x = \frac{(z \cdot \pi_1)}{\|\pi_1\|} \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \frac{(z \cdot \pi_2)}{\|\pi_2\|} \begin{pmatrix} 0 \\ 1 \end{pmatrix}.$$

It is possible to have a single 2D space with multiple \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 a z using each of these projections. This allows the different \mathcal{R}_j to be oriented differently in \mathcal{Z} (and be more distant than in the single projection case) [21].

2.3.3 Non-identical disconnected Pareto sets. [11] illustrate 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 subregion of the polygon. another disconnected Pareto set is illustrated via a map based problem, 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 design is less advantageous as here we would like to control a number of other problem properties when automatically generating problem instances, and ensure instances are viable, and Pareto sets are easy to identify *a priori*. We detail the approach we

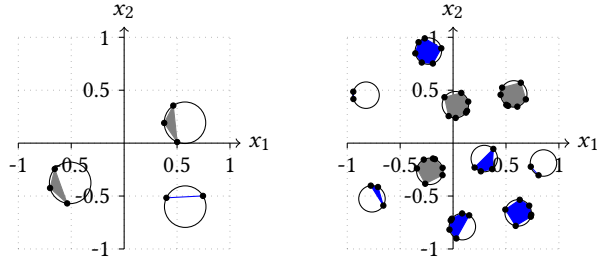


Figure 4: Illustrations of dominance resistance regions (coloured blue) in DBMOPP instances. Left: a $K = 3$ problem with two disconnected \mathcal{R}_i and one dominance resistance region. Right: a $K = 7$ problem with three disconnected \mathcal{R}_i and seven dominance resistance regions.

use here in section 3.3 (which relies on some additional features we must introduce first).

2.3.4 Dominance resistance regions. The usual generation of a DBMOPP results in all solutions which minimise any individual objective f_i being Pareto optimal point. [7] introduced region constructions which would overcome this limitation and supply designs which were dominance resistant [9] (i.e. dominated but weakly Pareto optimal when compared to Pareto set members [22]). 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 an \mathcal{R}_1 , 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.

3 NEW/ENHANCED DBMOPP FEATURES

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

3.1 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 regions². An illustration is provided in the top panel of Figure 5, with the corresponding *local dominance landscape* shown in the bottom 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 Pareto-neutral local optima regions). These may be identified by point-based Pareto

²Note, for computational reasons a problem instance generator must to pre-calculate the maximum local front radius, which must be smaller than m^{max} .

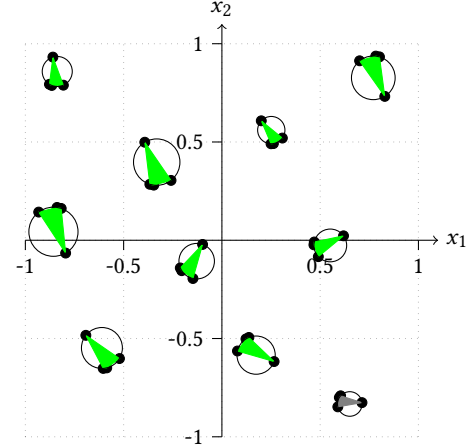


Figure 5: Illustration of local fronts. Top: A $K = 4$ problem instance with one global Pareto set and nine local front regions (in green). Bottom: local dominance landscape approximated by sampling \mathcal{X} on a 500×500 grid.

hill-climbing [29], but note 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. [23]). 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 bottom panel of Figure 5. The local Pareto-neutral regions include the

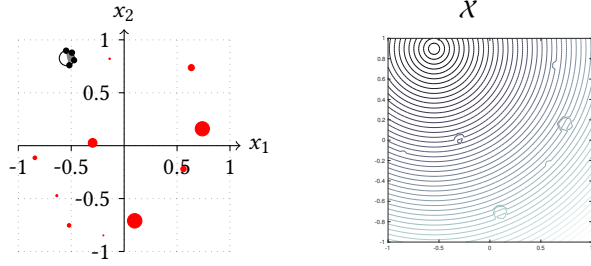


Figure 6: Illustrations of a penalty points and their effect on an objective landscape. Left: A $K = 4$ problem instance with 10 penalty regions (indicated with filled red circles). **Right:** f_1 quality landscape.

Pareto set and the regions denoting specified local fronts from the top panel, but also additional Pareto-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 Pareto-neutral regions may be larger than the corresponding region illustrated in the top panel of Figure 5 – this is because the Pareto-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).

3.2 Discontinuous objective surfaces

The use of the V in DBMOPP construction results in smooth objective landscapes. We propose here the introduction of discontinuities. These can be introduced via *penalty regions*, 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 f_i affected by the penalty at all locations that lie on the perimeter of the penalty region. An illustration of this is provided in Figure 6.

Here we use arbitrary convex polygonal shapes to define these penalty regions, and circular penalty regions defined by a centre and radius.

Where a penalty region intersects a \mathcal{R}_j , or lies entirely within one, additional features are induced, which we now detail.

3.3 Non-identical disconnected Pareto sets

Under most current DBMOPP formulations the image of each \mathcal{R}_j in \mathcal{X} under f describes the entire Pareto front. However, if we place penalty regions which intersecting with a \mathcal{R}_j (or which lie entirely within a \mathcal{R}_j), whose penalty is sufficient to make points within the penalty zone dominated by elements of \mathcal{X} , we can effectively ‘cut-out’ a chunk of that \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 of this is provided in Figure 7. Given the penalty locations some objective combinations are only available in one of the \mathcal{R}_i (e.g. the right-hand edge of the front in the middle – as this area is removed from two of the three \mathcal{R}_i),

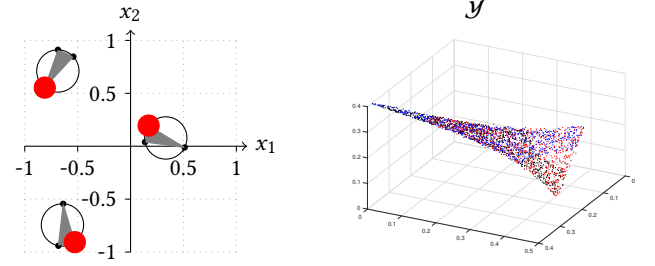


Figure 7: Illustrations of penalty coordinates generating non-identical disconnected Pareto sets. Right: Penalty zones intersect the three \mathcal{R}_i in different areas. **Left:** \mathcal{X} is sampled and evaluated under f , the non-dominated subset is shown, with the three different colours represented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

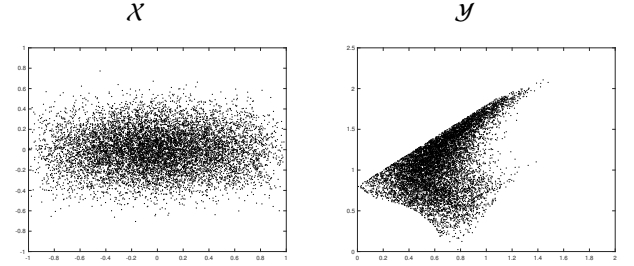


Figure 8: Illustration of varying density in \mathcal{X} , and thereby in the objective space. $N = 10$, $\sum \pi_1 = 2$, $\sum \pi_2 = 8$. The problem from Figure ?? is used. \mathcal{Z} is Monte Carlo sampled, with projected locations in \mathcal{X} plotted (right panel), and their image under f (left panel). Observe how the density of solutions has decreased in some areas and increased in others compared to the right panel of Figure ??.

some to different pairs of \mathcal{R}_i (i.e. the corner regions, where one of the three \mathcal{R}_i each have a penalty centred). Some optimal objective combinations reside in all three disconnected sets (i.e. the central portion of the front).

3.4 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 \mathcal{R}_j , and regions of the Pareto front, more difficult to attain than others.³ An illustration is provided in Figure 8.

³Alternatively a non-linear transform to the f_i value may be applied (i.e. taking the natural logarithm) to vary the density of mapping from the search domain to the objective space.

3.5 Varying objective scales

In standard formulations of DBMOPP the range of each objective does not vary greatly, and the minimum of all f_i 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, i.e. $f_i^{\text{rescaled}}(\mathbf{x}) = a_i + b_i \times f_i(\mathbf{x})$.

3.6 Neutrality

Neutral (flat) regions of the objective/domination landscape can be generated using the penalty region approach detailed in Section 3.2, 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 variables in the region have identical objective values for the set of objectives affected.

4 PROBLEM INSTANCE GENERATOR

Given the wealth of features described above extending the DBMOPP framework, which can be incorporated in a DBMOPP 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 regions types (Pareto sets, dominance resistance regions, penalty zones, non-identical disconnected Pareto sets including penalty zones, local fronts, etc.).

Algorithm 1 outlines the procedure at a high-level.⁴

4.1 Randomly placing region centres

We allocate the centres defining each of the regions at random, but subject to a lying at least $4r$ from the closest next region for all attractor regions. 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. In the case of 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 (i.e. at centres least $2r$ from other region centres), as they cannot induce Pareto optimal regions if placed too close (unlike the other region types).

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 $r^{\text{max}} = 1/(2 + \lceil \sqrt{(n + \frac{n'}{4})} \rceil)$, and for a particular problem instance $r \sim \mathcal{U}(0, r^{\text{max}})$.

⁴A 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 \LaTeX to generate illustrations (as in many of the subplots here) are available at URL-ANONYMISED-FORREVIEW.

Algorithm 1 DPMOPP-generator

```

1: function GENERATOR( $K, n_{\mathcal{R}}, n_l, n_d, n_{rr}, non\_identical,$ 
    $non\_convex, vary\_scales, random\_seed$ )
2:   set_seed( $random\_seed$ )  $\triangleright$  Ensure instance reproducibility
3:    $V_k := \emptyset \forall k$   $\triangleright$  Empty set of objective minima coordinates
4:    $(P_k, R_k) := (\emptyset, \emptyset) \forall k$   $\triangleright$  Empty penalty locations and radii
5:    $C := \emptyset$   $\triangleright$  Empty set of region centres
6:    $n := n_{\mathcal{R}} + n_l + n_d$   $\triangleright$  Total number of centred regions
7:    $r^{\text{max}} := \mathcal{U}(0, 1/(1 + \lceil 2\sqrt{n} \rceil))$   $\triangleright$  Draw  $r^{\text{max}}$ 
8:    $r^{\text{Pareto}} := r^{\text{max}}$   $\triangleright$  Default radius of Pareto set regions
9:    $\mathbf{a} := [0]_{K \times 1}$   $\triangleright$  Additive constants for objective rescaling
10:   $\mathbf{b} := [1]_{K \times 1}$   $\triangleright$  Multiplicative constants for objective
      rescaling
11:  if  $n_l > 0$  then  $\triangleright$  If local fronts are generated
12:     $(V, C, r^{\text{Pareto}}) := \text{place\_local\_fronts}(V, C, n_l, r^{\text{max}})$ 
13:  end if
14:   $(V, C) := \text{place\_Pareto\_set}(V, C, n_{\mathcal{R}}, r^{\text{Pareto}})$ 
15:  if  $n_{rr} > 0$  then  $\triangleright$  If dominance resistance regions needed
16:     $(V, C) := \text{place\_dom\_resistance}(V, C, n_{rr}, r^{\text{Pareto}})$ 
17:  end if
18:  if  $non\_identical = \text{true}$  then  $\triangleright |\mathcal{R}_i|$  vary
19:     $(P, R) := \text{modfiy\_regions\_with\_assym\_pen}(V, C, P, R)$ 
20:  end if
21:  if  $n_d > 0$  then  $\triangleright$  If discontinuities outside of  $\mathcal{R}_i$ 
22:     $(P, R, C) := \text{place\_discontinuities}(V, C, n_l, r^{\text{max}})$ 
23:  end if
24:  if  $vary\_scales = \text{true}$  then  $\triangleright$  Modify ojectives ranges
25:     $(\mathbf{a}, \mathbf{b}) := \text{sample\_scaling\_constants}(K)$ 
26:  end if
27:  return  $(V, P, R, \mathbf{a}, \mathbf{b})$   $\triangleright$  (Points, penalties, penalty radii,
      objective rescaling)
28: end function

```

In reality the legal Monte Carlo allocation of all $n + n'$ centres with $r = r^{\text{max}}$ is vanishingly small, as it essentially requires the random generation of $n + n'$ points on a regular grid. Subsequently, so although for each instance r is drawn from the uniform distribution $\mathcal{U}(0, r^{\text{max}})$, 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.⁵

5 ILLUSTRATION ON SOME POPULAR OPTIMISERS

⁵In practise we found that for all $r < 0.7r^{\text{max}}$ we found a legal set immediately, without recourse to shrinkage.

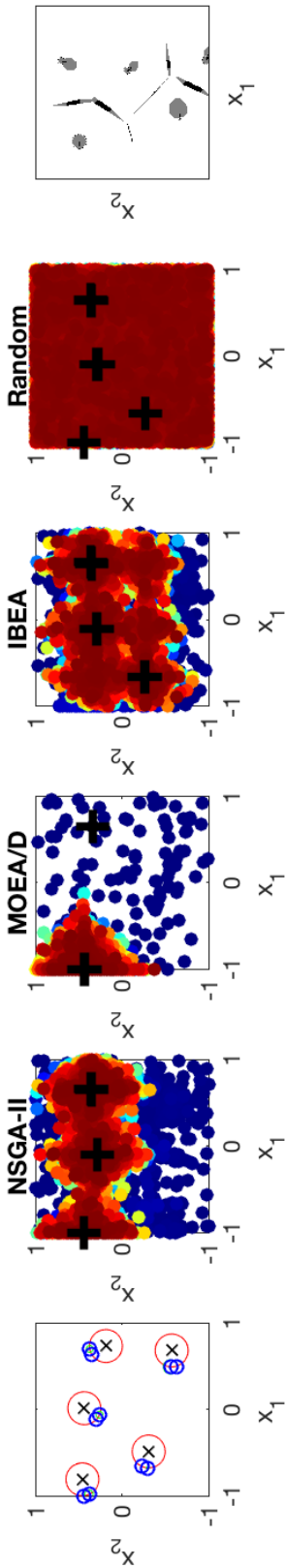


Figure 9: Caption

6 DISCUSSION

A number of different illustrations are possible:

- \mathcal{X} with regions indicated (Pareto set, local attractor sets, etc.)
- Basins of dominance attraction (greedy local domination moves shown with arrows, indicating total basin sizes)
- Densities in objective space for $K = 2$
- Projection densities in 2D when using orthogonal mapping vectors from higher D design space

Future work:

- More complex penalty geometries, allowing more complicated front surfaces (arbitrary polygons)
- Dynamic problem variants
- Disconnected front regions
- rigorous analysis of popular optimisers on range of problem instances

ACKNOWLEDGMENTS

ANONYMISED FOR REVIEW

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*. Springer, 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 *Mexican International Conference on Artificial Intelligence*. Springer, 688–697.
- [5] D. W. Corne and J. D. Knowles. 2007. Techniques for highly multiobjective optimisation: some nondominated points are better than others. In *Proceedings of the 9th annual conference on Genetic and evolutionary computation*. ACM, 773–780.
- [6] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler. 2005. Scalable test problems for evolutionary multiobjective optimization. In *Evolutionary multiobjective optimization*. Springer, 105–145.
- [7] J. E. Fieldsend. 2016. Enabling Dominance Resistance in Visualisable Distance-Based Many-Objective Problems. In *GECCO'16 Companion*. 1429–1436.
- [8] J. E. Fieldsend and R. M. Everson. 2013. Visualising high-dimensional Pareto relationships in two-dimensional scatterplots. In *Evolutionary Multi-criterion Optimization, EMO 2013*. 558–572.
- [9] T. Hanne. 1999. On the convergence of multi objective evolutionary algorithms. *European Journal of Operational Research* 117 (1999), 553–564.
- [10] 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.
- [11] H. Ishibuchi, N. Akedo, and Y. Nojima. 2011. A Many-Objective Test Problem for Visually Examining Diversity Maintenance Behavior in a Decision Space. In *Genetic and Evolutionary Computation Conference, GECCO'11*. 649–656.
- [12] H. Ishibuchi, N. Akedo, H. Ohyanagi, and Y. Nojima. 2011. Behavior of EMO Algorithms on Many-Objective Optimization Problems with Correlated Objectives. In *IEEE Congress on Evolutionary Computation (CEC)*. 1465–1472.
- [13] 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 *International Conference on Parallel Problem Solving from Nature*. Springer, 91–100.
- [14] 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.
- [15] 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*. Springer Berlin Heidelberg, 399–412.
- [16] M. Köppen and K. Yoshida. 2007. Substitute Distance Assignments in NSGA-II for handling Many-objective Optimization Problems. In *Evolutionary Multi-Criterion Optimization (EMO) (Lecture Notes in Computer Science)*. 727–741.
- [17] 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.
- [18] 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 (Feb 2018), 61–78. DOI: <http://dx.doi.org/10.1109/TEVC.2017.2655451>
- [19] 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)*. 2140–2147.
- [20] 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, EMO 2013 (Lecture Notes in Computer Science)*. 216–275.
- [21] 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)*. 2633–2640.
- [22] K. Miettinen. 1999. *Nonlinear Multiobjective Optimization*. Vol. 12. Springer Science & Business Media.
- [23] L. Paquete, T. Schiavinotto, and T. Stützle. 2007. On local optima in multiobjective combinatorial optimization problems. *Annals of Operations Research* 156, 1 (2007), 83.
- [24] J. Rönkkönen, X. Li, V. Kyrki, and J. Lampinen. 2008. A generator for multimodal test functions with multiple global optima. In *Asia-Pacific Conference on Simulated Evolution and Learning*. Springer, 239–248.
- [25] 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 *International Conference on Evolutionary Multi-Criterion Optimization*. Springer, 197–211.
- [26] 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*. Springer Berlin Heidelberg, 401–410.
- [27] T. Tušar and B. Filipič. 2015. Visualization of Pareto front approximations in evolutionary multiobjective optimization: A critical review and the projection method. *IEEE Transactions on Evolutionary Computation* 19, 2 (2015), 225–245.
- [28] M. Ullrich, T. Weise, A. Awasthi, and J. Lässig. 2018. A Generic Problem Instance Generator for Discrete Optimization Problems. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion (GECCO '18)*. ACM, New York, NY, USA, 1761–1768. DOI: <http://dx.doi.org/10.1145/3205651.3208284>
- [29] S. Verel, A. Liefooghe, L. Jourdan, and C. Dhaenens. 2011. Pareto local optima of multiobjective NK-landscapes with correlated objectives. In *European Conference on Evolutionary Computation in Combinatorial Optimization*. Springer, 226–237.
- [30] 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.
- [31] J. Xu, K. Deb, and A. Gaur. 2015. *Identifying the Pareto-Optimal Solutions for Multi-point Distance Minimization Problem in Manhattan Space*. Technical Report COIN Report Number 2015018. Michigan State University.
- [32] H. Zille and S. Mostaghim. 2015. Properties of scalable distance minimization problems using the manhattan metric. In *Evolutionary Computation (CEC), 2015 IEEE Congress on*. IEEE, 2875–2882.
- [33] 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.