
Proceedings Paper:
Yan, Y., Giagkiozis, I. and Fleming, P.J. (2015) Improved Sampling of Decision Space for Pareto Estimation. In: Silva, S., (ed.) GECCO '15 Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation. GECCO '15, 11-15th July 2015, Madrid, Spain. ACM , pp. 767-774. ISBN 978-1-4503-3472-3

https://doi.org/10.1145/2739480.2754713

Improved Sampling of Decision Space for Pareto Estimation

Y. Yan
Department of Automatic Control and Systems Engineering
The University of Sheffield
Sheffield, UK, S1 3JD
yiming.yan@sheffield.ac.uk

I. Giagkiozis
Department of Automatic Control and Systems Engineering
The University of Sheffield
Sheffield, UK, S1 3JD
i.giagkiozis@sheffield.ac.uk

P. J. Fleming
Department of Automatic Control and Systems Engineering
The University of Sheffield
Sheffield, UK, S1 3JD
p.fleming@sheffield.ac.uk

ABSTRACT

Pareto Estimation (PE) is a novel method for increasing the density of Pareto optimal solutions across the entire Pareto Front or in a specific region of interest. PE identifies the inverse mapping of Pareto optimal solutions, namely, from objective space to decision space. This identification can be performed using a number of modeling techniques, however, for the sake of simplicity in this work we use a radial basis neural network. In any modeling method, the quality of the resulting model depends heavily on the *training* samples used. The original version of PE uses the resulting set of Pareto optimal solutions from any multi-objective optimization algorithm and then utilizes this set to identify the aforementioned mapping. However, we argue that this selection may not always be the best possible and propose an alternative scheme to improve the resulting set of Pareto optimal solutions in order to produce higher quality samples for the identification scheme in PE. The proposed approach is integrated with MAEA-gD, and the resulting solutions are used with PE. The results show that the proposed method shows promise, in that there is measurable improvement in the quality of the estimated PE in terms of the coverage and density.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous; D.2.8 [**Software Engineering**]: Metrics—*complexity measures, performance measures*

Keywords

Local search, Metaheuristics, Neural networks, Surrogate model/fitness approximation, Machine learning

GECCO '15, July 11 - 15, 2015, Madrid, Spain

DOI: http://dx.doi.org/10.1145/2739480.2754713

1. INTRODUCTION

Real world problems often have multiple competing objectives that are to be optimized simultaneously, see [5, 3, 13, 27] to mention but a few. However, when multiple competing objectives are considered there no longer exists a single optimal solution. Rather, there exists a set of optimal solutions. This set is the minimal (maximal) set of the feasible objective vectors when all objectives are to be minimized (maximized). This introduces a subtle yet important difficulty, as in practice, only one solution can be selected at a given time. Therefore, a reduction of this optimal set is required. The process of reduction is context sensitive and most often requires the aid of a decision-maker (DM). Broadly speaking, there are 3 ways that the DM can be involved in the optimization process, i) *a-priori* preference articulation [23, p. 114], ii) interactive preference articulation (see for example [4, 24, 21]), and, iii) *a-posteriori* preference articulation [23, p. 77]. In *a-priori* preference articulation the DMs preferences are distilled into a utility function which can then be incorporated within the optimization process to reduce the multi-objective problem to a single objective problem [23, p. 115]. Interactive preference articulation methods involve the DM throughout the optimization process, and, *a-posteriori* preference articulation methods rely on the production of a *representative* optimal set from which the DM can select a single solution. For problems where there is a clear way to construct a utility function, *a-priori* methods are preferable as their computational requirements are lower than both alternatives. This is because only one problem is considered at a time, while in *a-posteriori* and interactive preference articulation methods a set of optimal points must be maintained [23, p. 115]. However, the creation of a utility function that accurately represents the DMs preferences is a challenging task. Interactive preference articulation methods can offer an alternative, which can be computationally more efficient than *a-posteriori* methods, since the optimization algorithm will have a progressively more restricted search space to explore. These methods however require a substantial time investment from the DM. Lastly, when there is no clear way for a utility function to be defined and the DM cannot be continuously involved with the optimization process, then *a-posteriori* preference articulation methods are preferable. For a more detailed exposition on methods and specific algorithms employed in decision-making the reader is referred to [25] and [23].

A-posteriori preference articulation can be summarized as follows:

Problem Formulation The analyst, using specifications described by the DM formulates an optimization model, which could be a mathematical programming formulation or a data-driven model, etc.

Solution Process Subsequently, based on the properties of the problem, the analyst identifies a suitable algorithm (e.g. gradient-based or evolutionary algorithms) to solve the optimization model.

Decision Making Based on the information provided from the algorithm, the DM can, i) make their decision if the solutions provided by the algorithm are desirable, ii) reformulate the model and repeat the process, or, iii) choose another algorithm and repeat the process.

One caveat with *a-posteriori* preference articulation methods is that computational resources impose a practical limit on the size of the *representative* optimal set that can be produced by the algorithm. This reveals an underlying meta-trade-off, namely, the size of the representative optimal set and the satisfiability of the DM, to wit, the larger the size of the representative set the more likely it is that a solution that will match the DMs preferences will be identified, albeit simultaneously the more expensive the optimization process becomes.

Pareto Estimation (PE) was introduced to alleviate some of these problems in the *a-posteriori* decision-making paradigm [6, 7]. In general, PE utilizes a set of candidate Pareto optimal solutions generated by an algorithm to identify the relationship between objective space and decision space. This can then be employed to generate more Pareto optimal solutions on the entire Pareto front or in specific regions, see for example in [6]. PE was shown to produce promising results on standard test problems with 2 and 3 objectives [6, 7]; particularly, a real world portfolio optimization example is also presented in [6] where the authors illustrate the ability of PE to increase the density of Pareto solutions within the regions of interest of a DM. Further development of PE includes [20, 19], where the authors extend PE to increase the density of multi-modal solutions of an MOP by using clustering analysis and employing PE for every cluster separately. Although in previous studies PE was used at the end of the optimization procedure, this is not a requirement, and, as we show in this work can in fact be beneficial to be incorporated in the algorithm. A recent work using similar ideas as PE employs Gaussian models to actively manage solutions during the optimization with promising results, opening a new dimension of application for Pareto estimation [1].

As is the case with all methods that are based on data-driven models, the performance of PE is contingent upon the quality of the Pareto set approximation produced by the optimization algorithm, for instance see [22]. In this work, we present an initial numerical study on the effect that the *quality* of samples has on the resulting PE model and its performance. In principle, for any interpolation method, Shannon's theorem (see [17] for a review) places a lower bound on the number of samples required for the complete determination of the function. However, the *placement* of these samples in the domain of the function is also very important, this placement is the object of study of optimal sampling [22].

The remainder of this work is organized as follows. In Section 2, we introduce a general formulation of multi-objective problems and the original Pareto estimation method. Section 3 presents a motivating example to better illustrate the basis for this work, i.e. generating better candidates of Pareto optimal solutions for PE by redistributing the solutions in the population. In Section 4 we demonstrate our methodology using MAEA-gD introduced in [10, 9]. Lastly, in Section 6, we conclude, summarize and reflect on future research directions based on this work.

2. BACKGROUND

2.1 Multi-Objective Optimization

Without loss of generality, a (continuous) multi-objective optimization problem can be defined as follows,

$$\begin{aligned} \min_{x} \quad & \mathbf{F}(\mathbf{x}) = (\, f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_k(\mathbf{x})\,) \\ \text{subject to} \quad & \mathbf{x} \in \mathbf{S} \subseteq \mathbb{R}^n, \end{aligned} \tag{1}$$

where $\mathbf{F}$ is a vector of scalar objective functions f_i with $i \in \{1, 2, \ldots, k\}$, k is the number of objectives, n is the number of decision variables, $\mathbf{x}$ is an n-dimensional vector of decision variables, $\mathbf{S}$ is the feasible region of $\mathbf{x}$ and $\mathbf{S}$ is an subset of the n-dimensional real space. For any $\mathbf{x} \in \mathbf{S}$, there exists a vector $\mathbf{z} = \mathbf{F}(\mathbf{x})$, which is an objective vector corresponding to the decision vector $\mathbf{x}$. The set of all possible objective vectors resulting from decision vectors in the feasible region are denoted as $\mathbf{Z}$, namely $\mathbf{Z} = \{\mathbf{z} \in \mathbb{R}^k \,|\, \mathbf{z} = \mathbf{F}(\mathbf{x}), \quad \forall \mathbf{x} \in \mathbf{S}\}$. A decision vector $\mathbf{x}^* \in \mathbf{S}$ is said to Pareto-dominate a decision vector $\mathbf{x}$, if and only if $f_i(\mathbf{x}^*) \leq f_i(\mathbf{x})$ for all $i \in \{1, \ldots, k\}$ and $f_i(\mathbf{x}^*) < f_i(\mathbf{x})$ for at least one i. A decision vector $\mathbf{x}^* \in \mathbf{S}$ is said to be Pareto optimal if it is not dominated by any other decision vectors in $\mathbf{S}$. For an objective vector $\tilde{\mathbf{z}} = (\tilde{f}_1, \ldots, \tilde{f}_k) \in \mathbf{Z}$, if there is no other objective vector $\mathbf{z} = (f_1, \ldots, f_k) \neq \mathbf{z}$, such that $f_i \leq \tilde{f}_i$ for all $i \in \{1, \ldots, k\}$ and $f_i < \tilde{f}_i$ for at least one i, we also say this objective vector $\tilde{\mathbf{z}}$ is non-dominated. The set of all non-dominated objective vectors in $\mathbf{Z}$, denoted as $\mathcal{P}$, is called the Pareto Front (PF). The decision vectors corresponding to $\mathcal{P}$ is denoted as $\mathcal{D}$. Optimization algorithms in the *a-posteriori* decision-making paradigm strive to obtain an approximation of the PF. In the remainder of this work we also denote the PF approximation as $\mathcal{P}$.

2.2 Pareto Estimation

In general, PE attempts to identify a relationship from the objective space to the decision space (inverse mapping) and then utilize this relationship to produce more Pareto solutions on the entire PF or in a specific region of the PF. To identify this relationship, the set of Pareto optimal objective vectors, $\mathcal{P}$, is first transformed into a projected set, $\tilde{\mathcal{P}}$. One element in $\mathcal{P}$ is mapped exactly to one element in $\tilde{\mathcal{P}}$, namely this transformation can be described as a function

$$\Pi^{-1} : \mathcal{P} \mapsto \tilde{\mathcal{P}}.$$

In the identification of the inverse mapping points in the projected space, $\tilde{\mathcal{P}}$, are used in lieu of $\mathcal{P}$, and, the motivation for this is to simplify the use of the resulting inverse mapping [7]. Elements in $\tilde{\mathcal{P}}$ are easily obtained and manipulated and therefore new samples are obtained in this space rather than the actual PF, whose topology is mostly unknown. One potential projection is to first normalize the objective vectors in $\mathcal{P}$ by

$$\tilde{z}_i = \frac{z_i - z_i^\star}{z_i^{nd} - z_i^\star}, \tag{2}$$

where $\mathbf{z}^\star$ is the ideal objective vector and $\mathbf{z}^{nd}$ is the nadir objective vector [23, pp. 15-16], which can be estimated from $\mathcal{P}$, and then project $\tilde{\mathbf{z}}_i$ onto the $(k-1)$-simplex, to produce the $\tilde{\mathcal{P}}$. The $(k-1)$-simplex is defined by $\{\mathbf{e}_1, \ldots, \mathbf{e}_{k-1}\}$, where $\mathbf{e}_i$ is a vector with a one in the i^{th} position and zeros in the rest. Following the projection, the relationship from $\tilde{\mathcal{P}}$ to the decision vectors $\mathcal{D}$ is identified, namely,

$$\tilde{F}_{\mathcal{P}} : \tilde{\mathcal{P}} \mapsto \mathcal{D}. \tag{3}$$

It is clear that a composite of Π^{-1} and $\tilde{F}_{\mathcal{P}}$ provides the relationship from the objectives to the decision variables,

$$\tilde{F}_{\mathcal{P}}(\Pi^{-1}(\mathcal{P})) = \mathcal{D}.$$

This procedure is illustrated in *Fig. (1)*. Since it is very

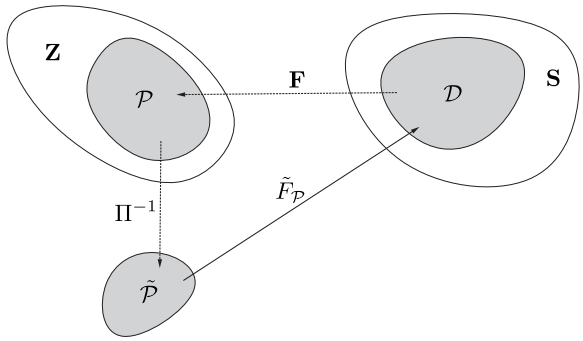

Figure 1: Pareto Estimation Method [7]

difficult, if not impossible, to mathematically derive the relationship $\tilde{F}_{\mathcal{P}}$, some meta-modeling method is used to identify this relationship approximately. Considering computational efficiency, the Radial Basis Function Neural Network (RBFNN) can be employed and the performance is promising [7]. A promising alternative to RBFNNs appear to be Gaussian processes, see for example [1].

With a set of evenly spaced samples, $\mathcal{E}$ as an input, $\tilde{F}_{\mathcal{P}}$ is able to generate a set of decision variables $\mathcal{D}_{\mathcal{E}} = \tilde{F}_{\mathcal{P}}(\mathcal{E})$ and this set of decision variables can then be used to generate the corresponding Pareto optimal objective vectors. If $\mathcal{E}$ is sampled in the entire projected objective space $\tilde{\mathcal{P}}$, PE will generate more solutions on the entire PF. Alternatively, if $\mathcal{E}$ is sampled in a specific region, then PE will be able to provide more solutions in a specific region of the PF.

For a given projection Π^{-1} and a meta-modeling method (e.g. RBFNN) for identifying the mapping $\tilde{F}_{\mathcal{P}}$, the quality of the identified relationship $\tilde{F}_{\mathcal{P}}$ from PE is mainly affected by the training data, i.e. the approximation of Pareto optimal solutions generated from the algorithm that is employed to solve a given MOP. Therefore the question we seek to answer is the following: How to manipulate the placement of the population during the optimization, so as to produce better data samples for PE, and therefore improve the quality of the resulting PE model and its performance, whilst ensuring that the convergence rate of the algorithm is not reduced. In the following section, we reflect on these questions and explore methods to address this problem.

3. MOTIVATING EXAMPLE

Consider the process of identifying $\tilde{F}_{\mathcal{P}}$ as the process of interpolating a function, whose domain is $\mathcal{D}$, using the candidate Pareto optimal solutions as samples. It was argued in the introduction that for a fixed number of samples their position can be an important in the quality of the resulting interpolations of the same function. To better understand the importance of sampling, let us consider the following example in *Fig. (2)*. In the example shown in *Fig. (2)*, we

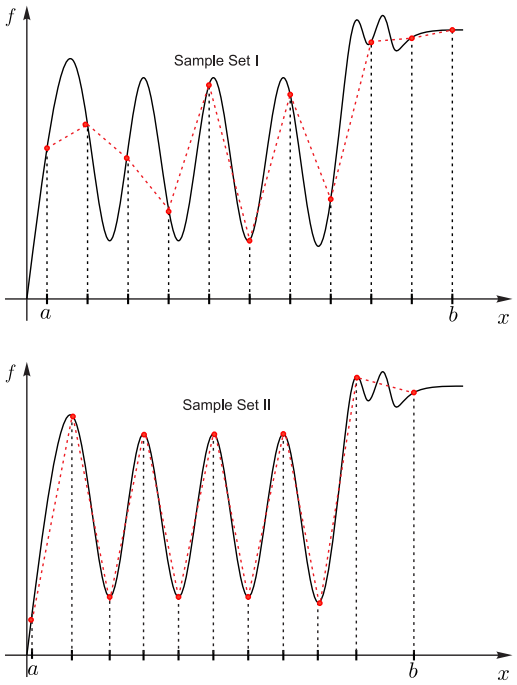

Figure 2: Piecewise linear interpolation (red dashed lines) of a function, $f(x)$, using two different sample sets (red dots).

assume that a piecewise linear interpolation function is used to obtain an approximation of the function f (black solid line). We compare the interpolations from two sample sets, Sample Set I and Sample Set II. Both sets contain 11 samples but the positions of the samples are different. It is clear that Sample Set II better approximates the function $f(x)$ when compared with Sample Set I. Samples from Sample Set I are evenly spaced, but due to the location of the samples, in particular the first four points from the left-hand side, the variation of the function is not sufficiently captured. Sample Set II is obtained by a left shift of the position of all samples within (a, b) in Sample Set I and a few other sample relocations. By doing this, Sample Set II produces a model that is more representative of the underlying function $f(x)$.

The above example illustrates that the topology of a function can be better interpolated if the high variation (and high amplitude) regions of the function are well sampled. The variation of a function can be measured by its *frequency* and *amplitude*. By amplitude of a function, we mean the distance from the 'top of a crest' or the 'bottom of a trough' to a baseline (or base hyperplane in a space of more than two dimensions); in the above example, we can consider a line that horizontally passes thorough the 'center' of the function as the base line which can be estimated by the mean of the

samples. We consider the regions with high frequency and amplitude as the *high variation regions*. Therefore, if we can identify the high variation regions of a function and allocate more samples there, we can produce better interpolation of a function.

Using the same line of reasoning we expect that if we distribute samples so as to capture most of the variations in decision space, we should be able to obtain a better approximation of the mapping $\tilde{F}_{\mathcal{P}}$, defined in (3) and therefore produce a better estimate of the PF using PE. Specifically, we break down our method into three parts.

1. Manipulate the current population to identify a model that maps from the objective space to the decision space. This process is identical to the PE method, and can be considered as building a *local approximate* of the mapping $\tilde{F}_{\mathcal{P}}$ using currently available information. This model is regarded as a local approximate because the current population may not be close enough to the actual PF and therefore can only provide partial information of the actual mapping. Quality of the local approximate improves, as the algorithm approaches the PF.
2. Utilize the local approximate of $\tilde{F}_{\mathcal{P}}$ to identify the areas in the decision space that are more 'important' for obtaining better topology of the Pareto optimal solution set in the decision space. Since $\tilde{F}_{\mathcal{P}}$ maps $\tilde{\mathcal{P}}$ to $\mathcal{D}$, the domain of the function is the projected space $\tilde{\mathcal{P}}$. This implies that we have to identify the areas of $\tilde{F}_{\mathcal{P}}$ that present more geometrical changes in the decision space and then find the corresponding regions in the projected objective space $\tilde{\mathcal{P}}$. This can be achieved by as follows: first we evenly sample the projected space, then use these samples and the local estimate of $\tilde{F}_{\mathcal{P}}$ to generate new decision vectors, and finally identify the regions according to certain measure of the geometrical changes (frequency and amplitude) of $\tilde{F}_{\mathcal{P}}$. In this work, we simply use the density of decision vectors as the measure. Alternative methods will be explored in future research.
3. Reallocate the position of the samples in the projected objective space, with more samples presenting in the areas identified above and generate the corresponding weighting vectors to the new samples using generalized decomposition (see Section 4.1).

To have consistently more samples in the identified areas in successive generations, the change of samples' locations in $\tilde{\mathcal{P}}$ need to be *transformed* to affect the search of the algorithm. For decomposition-based algorithm, for instance, we could transform these changes to alter the direction of corresponding weighting vectors; see next section for an algorithm that incorporate the above idea into a decomposition based algorithm.

4. MAEA-GD/RD

Decomposition-based multi-objective optimization methods have steadily increased in popularity in the last decade, see for example [15, 12, 26]. These methods transform (1) into a set of single-objective subproblems with the help of a scalarizing function and a set of weighting vectors. These subproblems are subsequently solved simultaneously to produce an approximation of the Pareto optimal set.

4.1 Generalized Decomposition

Given a measure for the quality of distribution of points on the Pareto front and a scalarizing function, generalized decomposition (gD) can be used to create a set of weighting vectors that will result in sub-problems that produce a set of solutions that are optimally distributed according to the given measure [10]. The optimality of the resulting distribution is contingent on the convergence of the sub-problems and knowledge of the Pareto front geometry *a-priori* to the solution of the problem. Nevertheless, it has been shown that when the PF geometry is unknown *a-priori*, which is most often the case, assuming an affine PF geometry still produces results that are several orders of magnitude better, in the selected measure, than commonly used alternative methods [9].

Generalized decomposition obtains weighting vectors by solving the following program:

$$\begin{aligned} &\min_{\mathbf{w}} G(\mathbf{w}, \mathbf{F}(\mathbf{x})), \\ &\text{subject to } \sum_{i=1}^{k} w_i = 1, \\ &\text{and } w_i \geq 0, \forall\, i \in \{1, \ldots, k\}. \end{aligned} \tag{4}$$

The assumption in (4) is that $G(\mathbf{w}, \mathbf{F}(\mathbf{x}))$ is a convex function with respect to the weighting vectors. This is the case for all ℓ_p-norm based scalarizing functions which are most commonly employed in decomposition-based optimization algorithms (see for example [16, 14, 18, 26]), a family that also includes the widely used Chebyshev decomposition as a limiting case [9].

4.2 MAEA-gD

MAEA-gD is a many objective optimization evolutionary algorithm based on generalized decomposition introduced in [10]. MAEA-gD is similar to MOEA/D [26], however, there are two significant differences, i) the neighborhood is based on distance in objective space and not weighting vector space, and, ii) generalized decomposition is used to generate the weighting vectors, see [10]. The main concepts in the algorithm are summarized as follows.

Step 1 Initialization

1. Generate N evenly distributed points on the $(k-1)$-simplex, where N is the size of population. We refer to this set as reference PF.
2. Utilize generalized decomposition, (4) to generate the set of weighting vectors, $\{\mathbf{w}_1, \ldots, \mathbf{w}_N\}$, using the reference PF.
3. For each point in the reference PF, find the T closest weighting vectors. Set $\mathcal{A}(i) = \{i_1, \ldots, i_T\}$, where $\mathbf{w}_{i_1}, \ldots, \mathbf{w}_{i_T}$ are T closest neighbors of $\mathbf{w}_i$.
4. Generate an initial population of size N, either randomly or by a problem-specific method.

Step 2 Update

For $i = 1, \ldots, N$, **do**

1. Randomly select two indices from the neighborhood $\mathcal{A}(i)$ and generate a new population $\mathbf{x}_i^{\text{new}}$ using genetic operators.
2. Evaluate $\mathbf{z}_i^{\text{new}} = \mathbf{F}(\mathbf{x}_i^{\text{new}})$.
3. Update population in the neighborhood $\mathcal{A}(i)$.

Step 3 Stopping Criteria
Terminate the algorithm if the stopping criteria are satisfied. Otherwise go to **Step 2**. Here we use number of iterations as the stopping criterion.

Step 4 Output the non-dominated solutions.

4.3 MAEA-gD with Sample Redistribution

By sample redistribution, we mean the reallocation of samples in the current population in objective space, so that more samples are present in regions where higher quality topological information is more likely to be obtained and used to improve the identification of $\tilde{F}_{\mathcal{P}}$. To achieve this, in the context of a decomposition based algorithm, we choose to alter the direction of the weighting vectors, which will guide the algorithm to generate samples in the desirable regions. This change however means that this re-sampling procedure cannot be performed on every iteration as this has the potential risk of stalling the algorithm convergence rate [8]. To minimize the impact to algorithm convergence, the sample redistribution is performed every $K = 10$ iterations. Furthermore, this *delay* also allows the algorithm to adapt the current solutions towards the new target solutions.

For every K iterations of MAEA-gD, the non-dominated decision and objective vector sets, $\mathcal{D}$ and $\mathcal{P}$, respectively, are used by the following redistribution algorithm.

Step 1: First, using (2), we normalize $\mathcal{P}$. Subsequently, the normalized $\mathcal{P}$ is projected onto the $(k-1)$-simplex. This results in $\tilde{\mathcal{P}} = \Pi^{-1}(\mathcal{P})$, as described in *Sec. 2.2*. In contrast with [7], in this work Π^{-1} is a radial projection. Namely, all elements in $\tilde{\mathbf{z}} = \mathbf{z}/\|\mathbf{z}\|_1$, where $\|\cdot\|_1$ is the ℓ_1-norm. With this projection the edge effects observed in [7] are removed and all points on the PF can be reached.

Step 2: $\tilde{\mathcal{P}}$ and $\mathcal{D}$ are used as the input and output, respectively, in the training of a RBFNN.

Step 3: Generate N evenly spaced vectors $\mathcal{E} = \{\bar{\mathbf{e}}_i\}, \forall i = 1, \ldots, N$, within the $(k-1)$-simplex, as described in [7]. Here N is the population size.

Step 4: Use $\mathcal{E}$ as the input to the RBFNN trained in **Step 2** to obtain estimates of decision vectors, $\mathcal{D}_{\mathcal{E}} = \{\bar{\mathbf{x}}_i\}, \forall i = 1, \ldots, N$.

Step 5: Compute the average distance from one vector to all others in $\mathcal{D}_{\mathcal{E}}$, namely $\forall i = 1, \ldots, N$, compute $d_i = \sum_{t=1, t\neq i}^{N} \|\bar{\mathbf{x}}_i - \bar{\mathbf{x}}_t\|_2/N$, where $\|\cdot\|_2$ is the ℓ_2-norm. Denote the index set $\mathcal{B} = \{b_j\}, \forall j = 1, \ldots, T$, where $d_{b_1}, \ldots, d_{b_T}$ are the T smallest average distances.

Step 6: Utilize the generalized decomposition method [10] to transform $\mathcal{E}$ to weighting vectors $\bar{\mathbf{W}} = \{\bar{\mathbf{w}}_i\}, \forall i = 1, \ldots, N$. Set $M = \sum_{j=1}^{T} d_{b_j}$ and assume the current set of weighting vectors for MAEA-gD is $\mathbf{W}$. In the following, we update the weighting vectors.

For $j = 1, \ldots, T$, **do**

6.1 Compute $s_j = (M - d_{b_j})/\sum_{t=1}^{T}(M - d_{b_t})$. Here, s_j, is a measure of the relative variation, i.e. the higher the value of s_j, the more variation in the neighborhood of $\bar{\mathbf{x}}_{b_j}$.

6.2 Calculate $n_j = \lfloor N \cdot s_j \rfloor$, where $\lfloor a \rfloor$ means the largest integer not greater than a. Note that $\sum_{t=1}^{T} n_t \leq N$.

6.3 Assume $\tilde{\mathbf{z}}_p, \forall p = 1, \ldots, n_j$, are the n_j closest vectors, in $\tilde{\mathcal{P}}$, of $\bar{\mathbf{e}}_{b_j}$, $\mathbf{w}_p \in \mathbf{W}$ is the weighting vector corresponding to $\tilde{\mathbf{z}}_p$, and $\bar{\mathbf{w}}_{b_j} \in \bar{\mathbf{W}}$ to $\bar{\mathbf{e}}_{b_j}$ Update weighting vectors for MAEA-gD by setting $\mathbf{w}_p = (1-\alpha)\mathbf{w}_p + \alpha\bar{\mathbf{w}}_{b_j}$, where α is in $(0, 1)$. Here we choose $\alpha = 0.05$. A small α value avoids large changes in the weighting vectors, and, therefore large changes in the sub-problems. This is aimed to counter the effect that varying weighting vectors have on algorithm convergence [8].

We refer to the algorithm MAEA-gD with sample redistribution as MAEA-gD/RD.

5. NUMERICAL EXPERIMENTS

In this section, we perform numerical tests to investigate the potential merit of the proposed methodology. by comparing the quality of the Pareto optimal solutions from PE based on the estimate of optimal decision vectors and objective vectors generated by algorithms MAEA-gD and MAEA-gD/RD respectively. Every algorithm is run 50 times, using a different seed in the random number generator on every run. The size of the population for both MAEA-gD and MAEA-gD/RD is 465 and both algorithms have 400 iterations. For each algorithm on every run, the approximation of the Pareto optimal set is obtained and subsequently PE is used to generate an estimated PF. For the estimation we employ the Pareto Estimation Toolbox from [6] and 3240 estimated Pareto optimal solutions are generated. For the test problems, we choose WFG6-9 [11] with 3 objectives and 24 variables, and DTLZ1-2 [2] with 3 objectives and 10 variables. To compare the quality of the estimated solutions obtained using PE from MAEA-gD and MAEA-gd/RD, we employ the following three measures.

- Inverted Generational Distance (IGD). IGD measures the distance from elements of one set to the actual PF of the test problem. In this test, we employ the the ratio of two IGD values, $D_R(\bar{P}_E, \hat{P}_E)$, defined as the ratio of the distance from $\bar{P}_E$ to the actual PF over that from the $\hat{P}_E$ to the actual PF, where $\bar{P}_E$ is the Pareto estimate generated by PE based on the solutions from MAEA-gD, and $\hat{P}_E$ from MAEA-gD/RD.
- Mean Distance to Nearest Neighbor. This metric measures the density of the solutions on a PF. Also, we employ the ratio $S_R(\bar{P}_E, \hat{P}_E)$ in our test, which is the ratio of the mean distance to the nearest neighbor in $\bar{P}_E$ divided by that in $\hat{P}_E$.
- Coverage Metric (C-Metric). C-metric (ranging from 0 to 1 inclusive) measures the ratio of non-dominated points between two sets of Pareto solutions, namely higher value of $C(\bar{P}_E, \hat{P}_E)$ indicates that $\bar{P}_E$ have more dominating points; higher value of $C(\hat{P}_E, \bar{P}_E)$ indicates the opposite. It is worth mentioning that $C(\bar{P}_E, \hat{P}_E)$ does not have to equal 1- $C(\hat{P}_E, \bar{P}_E)$ and so both values are presented in the test.

For details and mathematical formulation of the three measures, please refer to [7, Section V].

In Table 1 we summarize the ratios of the IGD values $D_R(\bar{P}_E, \hat{P}_E)$ and the mean distance to the nearest neighbor $S_R(\bar{P}_E, \hat{P}_E)$ for the selected 3-objective problems. Values for $D_R(\bar{P}_E, \hat{P}_E) > 1$ indicate that $\bar{P}_E$ produces a better IGD value than $\hat{P}_E$, namely $\bar{P}_E$ is closer to the actual PF than $\hat{P}_E$. From the results of $D_R(\bar{P}_E, \hat{P}_E)$ in Table 1, we can see that, for all 6 test problems, the values of $D_R(\bar{P}_E, \hat{P}_E)$ are close to or greater than one. This implies that, the the quality of the estimated PF, $\hat{P}_E$, from the optimal solutions of

MAEA-gD/RD, is at least as good as $\bar{P}_E$ from MAEA-gD, in terms of coverage and the distance to the actual PF. Regarding $S_R(\bar{P}_E, \hat{P}_E)$, the value of $S_R(\bar{P}_E, \hat{P}_E) > 1$ means that the average distance of the neighboring solutions is smaller in $\hat{P}_E$ compared to $\bar{P}_E$. Since the new fronts from $D_R(\bar{P}_E, \hat{P}_E)$ have similar coverage, a smaller average distance to the nearest neighbor in $\hat{P}_E$ indicates more evenly distributed solutions on the estimated PF. It can be seen by the results in Table 1, that MAEA-gD/RD has superior performance in comparison to PE applied on MAEA-gD, in 4 out of 6 problems, with only marginally inferior results on WFG6 and DTLZ2.

In Table 2, we present the values of C-Metric, $C(\bar{P}_E, \hat{P}_E)$ and $C(\hat{P}_E, \bar{P}_E)$, for the selected test problems. $C(\hat{P}_E, \bar{P}_E) > C(\bar{P}_E, \hat{P}_E)$ indicates that larger proportion of solutions in $\hat{P}_E$ dominates the solutions in $\bar{P}_E$; for $C(\hat{P}_E, \bar{P}_E) < C(\bar{P}_E, \hat{P}_E)$, the converse is true. The results suggest that, except for WFG6, MAEA-gD/RD improves Pareto estimation when compared with the estimation that results by using MAEA-gD. The reason for this improvement, although a welcome byproduct, is still an open question which we plan on investigating further in future work. A potential explanation could be that when altering the location of samples, our method implicitly performs local search on a surrogate model of the inverse of the objective function. Therefore, the updating procedure we use may also be improving algorithm convergence as on every K iterations a redistribution takes place.

The plots in *Fig. (4)* show the normalized local density of the estimated PF from PE for problems WFG7 and DTLZ2. The density is estimated based on the average distance from each Pareto optimal solution to its five nearest neighbors on the PF and is regularized between $[0, 1]$, where 0 is the smallest relative distance and hence highest density (densest, marked with warm color), and 1 represents the sparest, marked with color of cooler tones. Higher density translates to a larger number of Pareto optimal solutions in a neighborhood. It can be observed that the density of Pareto optimal solutions on the estimated PF obtained from the MAEA-gD/RD is slightly more uniform across the estimated PF, as there are fewer color variations when compared with the results obtained using MAEA-gD. See for example the bright yellow areas on the edges for both test problems for MAEA-gD/RD fade and dissolve into their neighborhood compared with these ares for MAEA-gD. This indicates that the resulting estimated solutions on the PF obtained by MAEA-gD/RD are more evenly distributed.

In *Fig. (3)* the final distribution of solutions on the Pareto front is illustrated for MAEA-gD and MAEA-gD/RD. At this point, it could be argued that the distribution of solutions from MAEA-gD/RD is *obviously* inferior. However, we argue that this is not in fact the case as better model for Pareto estimation has been identified using this distribution (for this problem) of solutions, which in turn can be used to create any desirable distribution in certain regions or across the PF. Furthermore, the distribution of the approximated Pareto optimal set in MAEA-gD/RD gives us information about the location of regions of high variation. This information can be leveraged by the analyst to produce more robust solutions, for example by requesting points from PE that are below a certain sensitivity threshold.

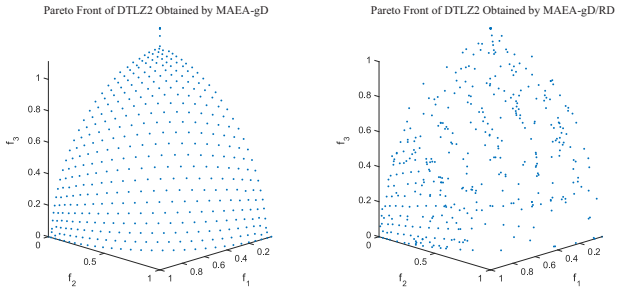

Figure 3: Pareto Front obtained from MAEA-gD and MAEA-gD/RD on DTLZ2

6. CONCLUSIONS

PE is a novel method of improving the density of a given approximate Pareto optimal solutions on the entire PF or in a specific region in which the DM is interested. In this paper, we present a method for improving the quality of the estimated Pareto optimal solutions from PE by allocating more samples in areas of high variation thus acquiring better topological information of the relationship between the projected objective vectors, $\tilde{\mathcal{P}}$, and decision vectors.

As the performance of PE is contingent upon the quality of the Pareto set approximation produced by the optimization algorithm, in this work, we present an initial numerical study on the effect that the quality of samples has on the resulting PE model, $\tilde{F}_{\mathcal{P}}$ and its performance. Given that in real world problems the objective function is usually computationally more expensive than the optimization algorithm, improvements in the PE model can result in better utilization of the available samples, therefore resulting in a more efficient use of computational resources.

In this work we explored a potential direction for improving the utilization of the information obtained from a MOEA

Table 1: $D_R(\bar{P}_E, \hat{P}_E)$ and $S_R(\bar{P}_E, \hat{P}_E)$ values of the Pareto estimate sets from MAEA-gD, $\bar{P}_E$ and MAEA-gD/RD, $\hat{P}_E$.

	$D_R(\bar{P}_E, \hat{P}_E)$			$S_R(\bar{P}_E, \hat{P}_E)$		
Problem	min.	mean	std.	min.	mean	std.
WFG6	0.6865	**1.0288**	0.1743	0.6163	0.9489	0.1195
WFG7	0.6744	0.979	0.1744	0.9775	**1.025**	0.0204
WFG8	0.7902	0.989	0.0829	0.9609	**1.0087**	0.0225
WFG9	0.6315	**1.0149**	0.1883	0.6756	**1.0416**	0.207
DTLZ1	0.9338	**1.0005**	0.0139	0.0061	**1.9934**	4.5145
DTLZ2	0.225	**1.593**	1.1893	0.9726	0.9924	0.0106

Table 2: C-Metric values of the Pareto estimate sets from MAEA-gD, $\bar{P}_E$ and MAEA-gD/RD, $\hat{P}_E$.

	$C(\bar{P}_E, \hat{P}_E)$			$C(\hat{P}_E, \bar{P}_E)$		
Problem	min.	mean	std.	min.	mean	std.
WFG6	0.0280	**0.6023**	0.2384	0.0573	0.4378	0.2413
WFG7	0.0004	0.0314	0.0305	0.0240	**0.1317**	0.0713
WFG8	0.1688	0.4073	0.0941	0.3077	**0.4191**	0.0661
WFG9	0.2306	0.5378	0.1726	0.2936	**0.6942**	0.1835
DTLZ1	0.0000	0.2572	0.4138	0.0000	**0.4779**	0.4931
DTLZ2	0.0000	0.0160	0.0355	0.0000	**0.0884**	0.0829

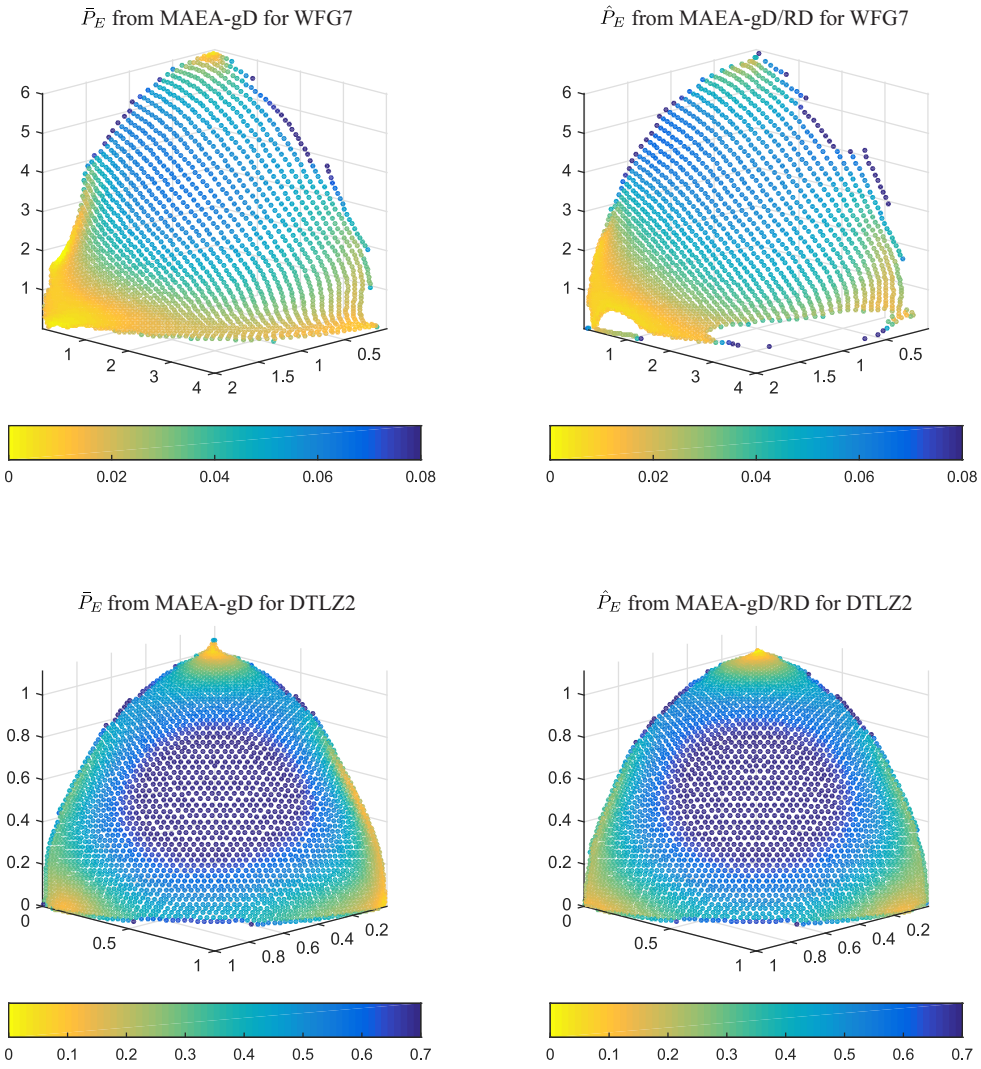

Figure 4: Density maps for WFG7 and DTLZ2. The color changes from bright yellow to dark blue as the the density decreases. Here 0 stands for the smallest distance and therefore highest density, and 1 for the sparsest.

to better estimate the PF using PE. The preliminary numerical results in *Sec. 5*, i.e. the statistics of three different measures and also the density plots, suggest that the proposed method can be useful for the production of better samples for PE which can be used to create a higher quality model of the inverse mapping, i.e. the mapping from the objective space to the decision space.

Nevertheless, despite the apparent superior performance of the method presented in this work in comparison with the original version of Pareto estimation [7], there is a number of open questions that need to be further investigated. As mentioned in the introduction real-world problems are usually multi-objective, and, often more than 3 objectives (many-objective problems) are involved. However, the scalability of PE to many-objectives has not been explored. This can have significant practical implications. Another issue is that the statistical results suggest that the solutions from PE and PE with the presented improved sampling strategy seem to produce superior solutions to the original approximation of the Pareto front generated from the algorithm. Although this result is a positive by-product, it should be further investigated as better understanding of the root cause for this behavior can suggest currently unforeseen improvements to PE. Lastly, we envisage that the obtained results could be further improved by exploring more better techniques for identifying high variation regions in decision space. These improvements can also lead to a method for identifying robust solutions, if the used robustness measures are based on the sensitivity of the inverse mapping.

7. ACKNOWLEDGMENTS

This work was supported by Jaguar Land Rover and the UK-EPSRC grant EP/L025760/1 as part of the jointly funded Programme for Simulation Innovation.

8. REFERENCES

[1] R. Cheng, Y. Jin, K. Narukawa, and B. Sendhoff. A multiobjective evolutionary algorithm using gaussian processbased inverse modeling. *IEEE Transactions on Evolutionary Computation*, PP(99), 2015.

[2] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler. Scalable multi-objective optimization test problems. In *Congress on Evolutionary Computation*, volume 1, pages 825 –830, may 2002.

[3] C. Erbas, S. Cerav-Erbas, and A. Pimentel. Multiobjective optimization and evolutionary algorithms for the application mapping problem in multiprocessor system-on-chip design. *IEEE Transactions on Evolutionary Computation*, 10(3):358–374, 2006.

[4] C. Fonseca and P. Fleming. Multiobjective Optimization and Multiple Constraint Handling with Evolutionary Algorithms. I. A Unified Formulation. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 28(1):26–37, 1998.

[5] C. Fonseca and P. Fleming. Multiobjective Optimization and Multiple Constraint Handling with Evolutionary Algorithms. II. Application Example. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 28(1):38–47, 1998.

[6] I. Giagkiozis and P. Fleming. Increasing the Density of Available Pareto Optimal Solutions. Research Report No. 1028, Department of Automatic Control and Systems Engineering, The University of Sheffield, November 2012.

[7] I. Giagkiozis and P. J. Fleming. Pareto front estimation for decision making. *Evolutionary Computation*, 22(4):651–678, 2014.

[8] I. Giagkiozis, R. Purshouse, and P. Fleming. Towards understanding the cost of adaptation in decomposition-based optimization algorithms. In *IEEE International Conference on Systems, Man and Cybernetics*, pages 615–620, Oct 2013.

[9] I. Giagkiozis, R. Purshouse, and P. Fleming. Generalized decomposition and cross entropy methods for many-objective optimization. *Information Sciences*, Available Online:1–25, Jun. 2014.

[10] I. Giagkiozis, R. C. Purshouse, and P. J. Fleming. Generalized decomposition. In *Evolutionary Multi-Criterion Optimization*, volume 7811 of *Lecture Notes in Computer Science*, pages 428–442. Springer Berlin Heidelberg, 2013.

[11] S. Huband, P. Hingston, L. Barone, and L. While. A Review of Multiobjective Test Problems and A Scalable Test Problem Toolkit. *IEEE Transactions on Evolutionary Computation*, 10(5):477–506, 2006.

[12] E. Hughes. Multiple Single Objective Pareto Sampling. In *Congress on Evolutionary Computation, 2003*, volume 4, pages 2678–2684. IEEE, 2003.

[13] E. Hughes. Radar waveform optimisation as a many-objective application benchmark. In *Evolutionary Multi-Criterion Optimization*, volume 4403 of *Lecture Notes in Computer Science*, pages 700–714. Springer Berlin / Heidelberg, 2007.

[14] E. Hughes. Many-objective directed evolutionary line search. In *Proceedings of the 13th annual conference on Genetic and evolutionary computation*, pages 761–768. ACM, 2011.

[15] A. Jaszkiewicz. On the Performance of Multiple-Objective Genetic Local Search on the 0/1 Knapsack Problem - A comparative Experiment. *IEEE Transactions on Evolutionary Computation*, 6(4):402–412, 2002.

[16] A. Jaszkiewicz. Do Multiple-Objective Metaheuristics Deliver on Their Promises? A Computational Experiment on the Set-Covering Problem. *IEEE Transactions on Evolutionary Computation*, 7(2):133–143, 2003.

[17] A. J. Jerri. The shannon sampling theoremâĂŤits various extensions and applications: A tutorial review. *Proceedings of the IEEE*, 65(11):1565–1596, 1977.

[18] Y. Jin, T. Okabe, and B. Sendho. Adapting Weighted Aggregation for Multiobjective Evolution Strategies. In *Evolutionary Multi-Criterion Optimization*, pages 96–110. Springer, 2001.

[19] R. Kudikala, I. Giagkiozis, and P. Fleming. Estimation of pareto optimal solutions for multi-objective multi-modal problems. In *19th International Conference on Soft Computing, MENDEL*, 2013.

[20] R. Kudikala, I. Giagkiozis, and P. Fleming. Increasing the density of multi-objective multi-modal solutions using clustering and pareto estimation techniques. In *The 2013 World Congress in Computer Science Computer Engineering and Applied Computing*, 2013.

[21] M. Luque, K. Miettinen, P. Eskelinen, and F. Ruiz. Incorporating preference information in interactive reference point methods for multiobjective optimization. *Omega*, 37(2):450–462, 2009.

[22] A. McBratney, R. Webster, and T. Burgess. The design of optimal sampling schemes for local estimation and mapping of of regionalized variablesâĂŤi: Theory and method. *Computers & Geosciences*, 7(4):331–334, 1981.

[23] K. Miettinen. *Nonlinear Multiobjective Optimization*, volume 12. Springer, 1999.

[24] M. Sun, A. Stam, and R. Steuer. Interactive multiple objective programming using tchebycheff programs and artificial neural networks. *Computers & Operations Research*, 27(7):601–620, 2000.

[25] C. Vira and Y. Haimes. *Multiobjective Decision Making: Theory and Methodology*. Number 8. North-Holland, 1983.

[26] Q. Zhang and H. Li. MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition. *IEEE Transactions on Evolutionary Computation*, 11(6):712–731, 2007.

[27] Z. Zhang. Multiobjective optimization immune algorithm in dynamic environments and its application to greenhouse control. *Applied Soft Computing*, 8(2):959–971, 2008.