Increasing the Density of Available Pareto Optimal Solutions

Ioannis Giagkiozis, and Peter J. Fleming,
Department of Automatic Control and Systems Engineering
The University of Sheffield
Research Report No. 1028
November 2012

***Abstract*—The set of available multi-objective optimization algorithms continues to grow. This fact can be partially attributed to their widespread use and applicability. However this increase also suggests several issues remain to be addressed satisfactorily. One such issue is the diversity and the number of solutions available to the decision maker (DM). Even for algorithms very well suited for a particular problem, it is difficult - mainly due to the computational cost - to use a population large enough to ensure the likelihood of obtaining a solution close to the DMs preferences. In this paper we present a novel methodology that produces additional Pareto optimal solutions from a Pareto optimal set obtained at the end run of any multi-objective optimization algorithm. This method, which we refer to as Pareto estimation, is tested against a set of 2 and 3-objective test problems and a 3-objective portfolio optimization problem to illustrate its' utility for a real-world problem.**

***Index Terms*—Pareto Estimation, Evolutionary Algorithms, Metaheuristics, Multi-Objective Optimization, Nonlinear Estimation**

I. Introduction

MUCH research effort has been dedicated to finding the Pareto optimal set (PS) of multi-objective optimization problems (MOPs). Many algorithms have been developed to solve MOPs. Broadly speaking, these can be categorised in two groups, based on their approach to fitness assignment; - (i) algorithms based on Pareto dominance relations and (ii) decomposition based approaches. Most algorithms developed during the 1990s and early 2000s were Pareto-based [1]–[4], and to this day the majority of the methodologies rely on some variant of this type of fitness assignment. Decomposition-based optimization techniques draw upon the fact that Pareto optimal solutions can be obtained by aggregating the objective functions into a set of scalar optimization problems [5]. This set of problems can, in principle, be solved using some single objective optimization algorithm.

In an *a posteriori* preference articulation[1] scenario [5, pp. 77], the main focus of multi-objective optimization algorithms is to approximate the Pareto optimal Set (PS)[2] as quickly as possible, and distribute the Pareto optimal solutions evenly. Although, this objective can easily become unmanageable for problems with more than 3 objectives. This is because for increasing number of dimensions in objective space the number of solutions required to *cover* the entire front increases exponentially [6] and the *sweet spot* of the evolutionary algorithm parameters that produce good solutions is reduced in size [7]. A potential solution to this problem could be to restrict the search space by enforcing constraints in objective space, see for instance [1]. In the present work we focus our attention to 2 and 3-objective problems to avoid such complications and better illustrate the presented concepts.

However, if the decision maker (DM) is not completely satisfied with the obtained PS, he/she can either recourse to, a different algorithm or restart the preferred algorithm with different parameters in the hope that the new PS will more closely satisfy the requirements. Progressive-preference articulation algorithms [1], [5], offer an alternative approach - however the drawback is that the DM must be *in-the-loop* for the algorithm execution [5] and this can be rather demanding.

The proposed method in this paper alleviates these difficulties for continuous MOPs by producing more and, usually, better distributed Pareto optimal solutions along the entire Pareto front (PF). Additionally, an important feature that may be helpful to both the analyst and the decision maker is that if there is a specific region of interest on the PF, the generation of solutions can be focused on that region. This can be helpful in situations where there is a set of solutions about a part of the PF that the DM is interested in; but no solution is found by the algorithm in that region. The method we propose achieves this result by estimating the mapping of a convex set to the decision vectors corresponding to the Pareto optimal solutions obtained on the final iteration of a multi-objective optimization algorithm. This convex set is used in lieu of the objective vectors for reasons that are clarified in Section III and Section IV. This mapping, identified here using a radial basis function neural network (RBFNN), is then used to generate estimates of decision vectors that would lead to Pareto optimal solutions in the neighbourhood of the original solutions. It should be noted that an RBFNN is chosen mainly because it is computationally efficient to train and the produced results are reasonable for the selected test problems. However this choice is not restrictive and does not characterise the presented methodology, as any modeling or metamodeling method can be used instead - should the situation demand it.

The authors are with the Department of Automatic Control and Systems Engineering, University of Sheffield, Sheffield, UK, S1 3JD.
E-mail: i.giagkiozis@sheffield.ac.uk

[1]The interaction of the analyst and the decision maker takes place after a set of Pareto optimal solutions has been generated.

[2]See Definition 4.

The idea that supports the proposed method, which we refer to as Pareto Estimation method (PE), is that, for continuous MOPs, it can be deduced from the Karush-Kuhn-Tucker optimality conditions that the PS is piecewise continuous in the decision variables, as previously noted in [8]. This fact, combined with a reasonable approximation of the PS, can be used constructively to infer the mapping of the above-mentioned convex set to decision variables that produces Pareto-optimal solutions.

The main contributions of this work can be summarised as follows:

- A method, which we call Pareto Estimation (PE), is presented. PE can be used to increase the number of Pareto optimal solutions, for 2 and 3-objective problems, from an approximation of the Pareto set. This can be useful in a situation where the evolutionary algorithm has not produced a solution close enough to the desired location on the PF. Furthermore the Pareto estimation method does not necessitate any alteration to the optimization algorithm that is used to produce the Pareto set and is dependent on it only as far as the quality of the PS is concerned.
- The effectiveness of PE is validated using a set of test problems, commonly used in the MOEA community, for 2 and 3-objectives. It is shown that PE can produce more Pareto optimal solutions across the entire PF with a much lower cost compared to the alternative of restarting the optimization or using an alternative algorithm to solve the MOP. Also it is much more flexible compared with progressive preference articulation methods [1]. Although this is not the main purpose of the PE method, it is a good test since if it can produce more solutions on the entire PF then it should be able to increase the number of solutions in specific regions as well, which we believe to be the main utility of PE.
- Furthermore we consider a real-world problem, namely a 3-objective portfolio optimization problem, whereby, an increased number of Pareto optimal solutions is produced along the entire PF as well as in specific regions,with the help of the Pareto estimation method.

The remainder of this paper is organized as follows. In Section II a general formulation of a multi-objective optimization problem is given followed by some fundamental definitions. Related concepts and motivating ideas for the proposed method are discussed in Section III. In Section IV the Pareto estimation method is described for Pareto and decomposition-based algorithms. The method is tested against a set of multi-objective optimization problems and these tests are reported in Section V and in Section VI PE is applied to a 3-objective portfolio optimization problem. Lastly in Section VII we discuss problems, potential solutions and ideas related to the PE method and in Section VIII this paper is summarized and concluded.

II. Problem Setting and Definitions

A general definition of a multi-objective problem (MOP) is

$$\min_{\mathbf{x}} \mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_k(\mathbf{x})), \quad \text{subject to } \mathbf{x} \in S, \tag{1}$$

where the number of *objective functions* is k, S is the *feasible region* for the *decision* vectors $\mathbf{x}$, and $f_i(\mathbf{x})$ is a scalar objective function, with $i \in \{1, ..., k\}$. Additionally, in this work it is assumed that $\mathbf{x} \in \mathbb{R}^n$ and that n is the number of decision variables in the decision vector x. Also, depending on the definition of S, (1) can be a constrained MOP. For simplicity, only minimization problems are considered; however, this does not limit the generality of the produced results since due to the duality principle, maximization of a scalar objective function f_i is the same as minimization of $-f_i$. An implicit assumption is that the scalar objective functions in (1) are mutually competing and possibly non-commensurate.

If the above assumptions obtain then only a partial ordering can be defined unambiguously. Namely when comparing two decision vectors $\mathbf{x}, \tilde{\mathbf{x}} \in S$, it can so happen that their corresponding objective vectors are incomparable. In practise, this situation is resolved by a decision maker who will select one solution over all others, thus inducing a form of complete ordering. However this ordering is mostly subjective, even in the case that utility functions [9] are used to ease the work of the DM. In the absence of a DM a usual assumption is that the relative importance of the objectives, f_i, is unknown hence it is reasonable to obtain several non-comparable solutions. The problem of inducing partial ordering in Euclidean spaces was initially studied by Edgeworth [10], and later further expanded by Pareto [11]. The relations introduced by Pareto are defined as follows for a minimization problem:

Definition 1. *A decision vector $\mathbf{x}^\star \in S$ is said to **weakly dominate** a decision vector $\mathbf{x}$ iff $f_i(\mathbf{x}^\star) \leq f_i(\mathbf{x}),\ \forall i \in \{1, 2, \ldots, k\}$ **and** $f_i(\mathbf{x}^\star) < f_i(\mathbf{x})$, for at least one $i \in \{1, 2, \ldots, k\}$ then $\mathbf{x}^\star \preceq \mathbf{x}$.*

Definition 2. *A decision vector $\mathbf{x}^\star \in S$ is said to **dominate** a decision vector $\mathbf{x}$ iff $f_i(\mathbf{x}^\star) < f_i(\mathbf{x}),\ \forall i \in \{1, 2, \ldots, k\}$ then $\mathbf{x}^\star \prec \mathbf{x}$.*

Definition 3. *A decision vector $\mathbf{x}^\star \in S$ is said to be **Pareto optimal** if there is no other decision vector $\mathbf{x} \in S$ such that $f_i(\mathbf{x}) \leq f_i(\mathbf{x}^\star),\ \forall i \in \{1, 2, \ldots, k\}$ and $f_i(\mathbf{x}) < f_i(\mathbf{x}^\star),$ for at least one $i \in \{1, 2, \ldots, k\}$.*

Definition 4. *Let $\mathbf{F} : S \to Z$, with $S \in \mathbb{R}^n$ and $Z \in \mathbb{R}^k$. If S is the feasible region then the set Z is the feasible region in objective space. Given a set $\mathbf{A} \subset Z$, the **non-dominated set**[3] is defined as $\mathcal{P} = \{\mathbf{z} : \nexists \tilde{\mathbf{z}} \preceq \mathbf{z}, \forall \tilde{\mathbf{z}} \in \mathbf{A}\}$. If $\mathbf{A}$ is the entire feasible region in the objective space, Z, then the set $\mathcal{P}$ is called the **Pareto optimal set** (PS) or **Pareto Front** (PF). Any element $\mathbf{z} \in Z$ is referred to as **objective vector**.*

Also the following definitions are used in this work in various contexts:

Definition 5. *The **ideal objective vector**, $\mathbf{z}^\star$, is the vector with elements $(\inf(f_1), \ldots, \inf(f_k))$ [5, pp. 16].*

[3] Or **Pareto Front approximation**.

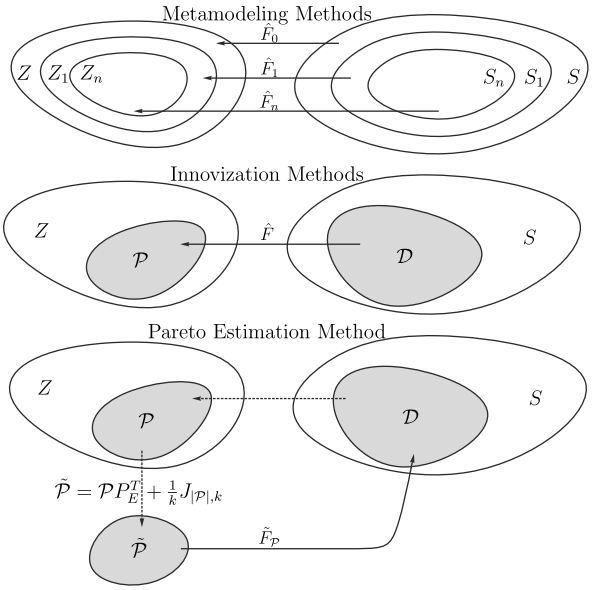

Fig. 1. Metamodeling methods in EAs gradually refine a surrogate model and then use it to find a better Pareto set approximation. Innovization methods use the final Pareto set approximation to identify *design* rules, namely decision vector relations that map to Pareto optimal solutions. Pareto Estimation, the method proposed in this work, proceeds in the reverse direction by mapping a surrogate set, $\tilde{\mathcal{P}}$, of a Pareto front approximation, $\mathcal{P}$, to the decision vector set that maps to $\mathcal{P}$.

Definition 6. *The **nadir objective vector**, $\mathbf{z}^{nd}$, is the vector with elements $(\sup(f_1), \ldots, \sup(f_k))$, subject to f_i be elements of objective vectors in the Pareto optimal set [5, pp. 16].*

Definition 7. *The **convex hull** [12, pp. 24] of the set $C = \{\mathbf{e}_1, \ldots, \mathbf{e}_k\}$, denoted as $\mathbf{conv}\, C$, where $\mathbf{e}_i$ is a $k \times 1$ vector of zeros with 1 on the i^{th} position, is referred to as $\mathbf{CH_I}$.*

Definition 8. *The **extended convex hull** ($\mathbf{EH_I}$) of the set C, is the union of CH_I and the points in the affine space of the set C produced by the projection of a Pareto optimal front, with ideal vector $\mathbf{0}$ and nadir vector $\mathbf{1}$, onto the hyper-surface of C.*

A multi-objective evolutionary algorithm (MOEA) will in general attempt to obtain a *representative* approximation of the Pareto optimal set. However what is considered to be a representative approximation of the PS is not context independent, but metrics have been devised to measure the *quality* of the PS. For an excellent review of this topic the reader is referred to [13].

III. Related Work

Multi-objective evolutionary algorithms have had tremendous success in solving real-world problems. They have been applied in control systems [14]–[16], in economics and finance [17]–[20] and aerospace [21], [22]. This can be attributed to the fact that evolutionary algorithms (EAs) perform well for a wide range of problems that classical methods, such as convex optimization [12], are inapplicable. However the robustness of EAs does not come for free. For example, contrary to convex optimization, there is no guarantee of global optimality for solutions produced by evolutionary algorithms. Although in practise there is strong evidence that very good approximations of Pareto optimal solutions are generated.

A. Metamodeling Methods in Multi-Objective Optimization

An additional challenge that MOEAs face is that the cost, of objective function evaluations, for a Pareto optimal solution to be found is relatively high. This coupled with objective functions that can take hours or days to evaluate constitutes a severe limitation which is widely acknowledged in the MOEA community [23]–[26]. A prevalent methodology employed by researchers to tackle this issue is the use of metamodeling methods in optimization. The insight is that, if a surrogate model of the actual objective function can be created with relatively few samples, then this surrogate model can be used instead of the objective function in the optimization process.

Since the purpose of the surrogate model is to relieve the EA from evaluating an expensive objective function as much as possible, the primary selection criteria for a surrogate model are adapted accordingly. Namely the suitability of a modeling method is judged according to; - (i) the ease with which the model parameters can be identified and, (ii) the cost of one evaluation of the surrogate model which must be much smaller than that of the actual objective function. Therefore, for a metamodeling method that satisfies the above criteria, a large number of objective function evaluations can be substituted with calls to the surrogate model, hence reducing the total cost of the optimization. Another criterion that is definitive in the success of the aforementioned procedure is the model precision. This so because if the surrogate model cannot capture important features of the objective function the search will be grossly misled, although caution should be exercised not to overcomplicate the surrogate model to a degree that its cost becomes comparable to the original objective function. In a way, a surrogate model function can viewed as a low-pass filter, hopefully separating the *noise* from the important features of the objective function, that is its' minima (or maxima). This is why such methods have been employed in noisy optimization problems as well [27].

The general approach when substituting the real objective function with a surrogate model for use in an EA, has the following structure:

- **Step 1** Sample the real objective function.
- **Step 2** Using the obtained samples create a surrogate model.
- **Step 3** Use the surrogate model in the optimization.
- **Step 4** If the convergence criteria are met stop, if not go to **Step 1**.

An illustration of this iterative procedure can be seen in *Fig. 1*, where an ever more accurate mapping of the decision space, S, to the objective space Z, is created in every iteration, $\{\hat{F}_0, \hat{F}_1, \ldots\}$. This approach was initially limited to serial implementations [28], [29], however later advances in metamodeling-based EAs employed local models [30] thus reinstating a key strength of EAs, their potential to be executed in parallel.

The idea to employ surrogate models in lieu of the the *true* model of a process can be dated back to Box and Wilson [31]

where they employed polynomial basis functions to create a model from data. This approach is commonly referred to in the literature as response surface method (RSM). Other examples of modeling methods used in combination with an evolutionary algorithm are neural networks [25], [32], [33] (multi-layer perceptron as well as radial basis function networks), Kriging or Gaussian processes generalized response surface methods [34] as well as Bayesian regression [35].

B. Innovization Methods

Another issue that is not yet been satisfactorily addressed, especially for many-objective problems[4], is that the final Pareto set contains information that can be used to infer relationships in decision space that result in Pareto optimal solutions. A method that attempts to answer this question was presented by [36], which the authors call *innovization*. The authors argue that by identifying a set of *design rules* the multi-objective problem will not have to be solved again. Although this premise seems intriguing, to generate such design rules requires great effort on the behalf of the analyst, thus is limited to very low dimensional problems in decision and objective space [37]. Another difficulty with this method is that the optimization algorithm has to be specifically tailored to the process [36]–[38]. To deal with this shortcoming further work presented in [37] attempts to resolve this by partially automating the procedure. The objective in such methods is to identify a mapping from decision space to objective space that will guarantee that the resulting solutions will be Pareto optimal, see *Fig. 1*. This amounts to identifying a set of constraints/relationships in decision space that if adhered to, will produce the desired results. However in these methods there isn't a clear way to obtain Pareto optimal solutions in a specific region on the Pareto front, except by manually constructing different relationships on different parts of the front, something that can easily become unmanageable for even the smallest problems. This fact can attested by the size of the problems selected in [36], [37] which never exceeded $2-5$ decision variables and 2 objectives.

C. Pareto Estimation Method - Motivation

In this work we bring forward and resolve, to some extent, a question that seems to be ignored by the literature[5]. Namely given an approximation of the Pareto front by any MOEA, is there a way to obtain solutions, in specific parts of the PF, that are not present in the given set, and if the answer is positive, how can this be achieved? This question stems from the fact that if there was a way to obtain a Pareto optimal solution that adheres exactly to the decision makers' preferences, then there would be no need to evaluate any other solutions as it is usually the case for multi-objective optimization algorithms. However such algorithms inherit this strategy because there is no clear path in incorporating all preferences since it is unknown if they are in fact reasonable, that is to say if there exist such a solution at all. Additionally there is no clear way in obtaining a specific solution with accuracy. The remainder of this paper considers this question and an answer is presented.

However to appreciate the importance of this question, let us embark on a though experiment. Assume that we have a function,

$$G(\mathbf{z}) = \begin{cases} \mathbf{x} & \text{if and only if } F(\mathbf{x}) = \mathbf{z}, \text{ and } \mathbf{z} \in \mathcal{P} \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

Namely the function, G, returns the corresponding Pareto optimal decision vector if a Pareto optimal solution, $\mathbf{z}$, is used and 0 otherwise. Obviously such a function would be of limited use if the the analyst had no information about the shape of the Pareto front as well as its location. Namely the function, G, is a special indicator function with domain of definition the Pareto optimal set, $\mathcal{P}$, and range the Pareto optimal decision vectors, $\mathcal{D}$. Therefore given such a function and the information about the exact location of the Pareto front; it would be simply a matter of evaluating (2) in order to obtain the decision vector that would result in a Pareto optimal solution. Such a description of the Pareto front geometry can be given by a parametric or non-parametric model if the problem has already been successfully solved by some method. A potential issue with such an approach is that a different description of the PF will be required for different problems. Although this seems troubling, there is nothing to preclude the existence of a function with a *convenient* domain of definition, that would map to the Pareto front of any given problem. Naturally such function must depend, and adapt to, the Pareto optimal set or some approximation of it, and hopefully a procedure can be found to map the former to the latter. Strictly speaking such a function would perform the following task,

$$\Pi(\mathbf{w}) = \mathbf{z},\ \mathbf{z} \in \mathcal{P}. \tag{3}$$

Additionally it would be even more convenient if the mapping, Π, was predictable in the sense that for a given $\mathbf{w}$ the resulting $\mathbf{z}$ is not very hard to predict, as this would ease the complexity of using the function (3). A natural candidate for such a task would be an affine function, that is a linear function plus an offset.

The final piece of this puzzle lies in the domain of definition of the function described in (3). The requirements on such a domain would be; - (i) that points within the domain of definition of the function, Π, should be easy to obtain and (ii) any convex combination of the points in the set must still be in the set, that is to say the set must be convex. By adhering to these requirements, and if relations similar to (2) and (3) could be identified, then by the following procedure a Pareto optimal solution could be obtained at any desired location on the PF,

- Choose a $\mathbf{w}$ that would produce the desired $\mathbf{z}$. This is verified by (3), if the resulting $\mathbf{z}$ is not the intended one; it would be sufficient to change $\mathbf{w}$ a little. In this step we exploit the *predictability* of the mapping, Π.
- Use the obtained $\mathbf{z}$ in (2) to obtain the decision vector, $\mathbf{x}$, that would produce the objective vector $\mathbf{z}$.
- Evaluate the actual objective function, F, using the obtained $\mathbf{x}$ to verify that $F(\mathbf{x}) = \mathbf{z}$. Although strictly

[4]Problems with more than three objectives.

[5]To the authors' best knowledge.

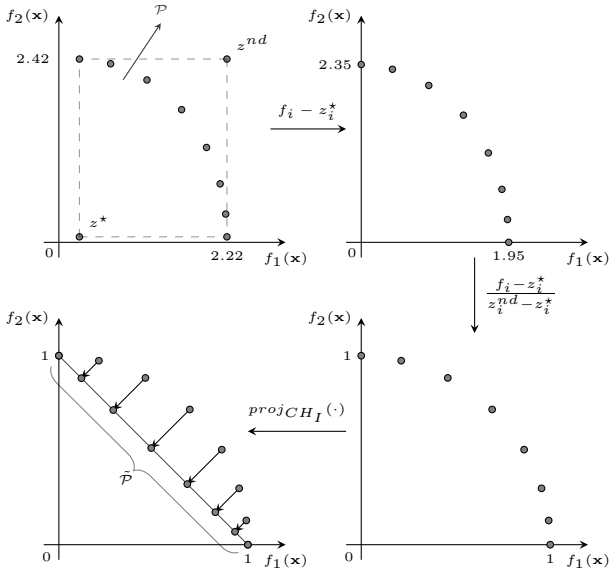

Fig. 2. Illustration of the Π^{-1} mapping for a hypothetical Pareto set $\mathcal{P}$.

speaking, should the mappings described in (2) and (3) be exact, this step is redundant.

So, if such a procedure was available in practise, there would be a way to obtain the decision vector that satisfies the requirements of the decision maker exactly, instead of repeatedly solving a multi-objective optimization problem, in hope to obtain a solution that closely satisfies the aforementioned requirements.

IV. Pareto Estimation Method

A. Overview

The question posed in Section III-C, is interesting because depending on how well it can be answered, the information that is in the analysts' possession increases dramatically, thus allowing the analyst to cater to more specific requests from the decision maker. This is so because given absolute knowledge of the aforementioned functions, (2) and (3), a multi-objective problem is virtually solved, as any solution on the Pareto front could, theoretically, be obtained with a very small additional expense and high precision. Although to obtain the entire Pareto optimal set may be infeasible in practice, this is the predominant definition of what it means to *solve* a multi-objective optimization problem [5, pp. 61].

However, such a relationship is usually unknown for real-world problems and sometimes it is unknown even for test problems. Most multi-objective optimization algorithms strive to generate a PS which possesses two key properties, first, it should produce objective vectors as close as possible to the true PF and, second, these objective vectors should be evenly spread across the PF hyper-surface. Under the assumption that the optimization algorithm of choice has succeeded, to a reasonable degree, in producing a PS that possesses the aforementioned properties, then the mapping, $F_{\mathcal{P}}$, of Pareto optimal objective vectors, $\mathcal{P}$, into[6] their corresponding decision variables $\mathcal{D}$,

$$F_{\mathcal{P}} : \mathcal{P} \rightarrow \mathcal{D} \ , \tag{4}$$

can be identified using a modeling method [34]. A theoretical argument based on the Karush-Kuhn-Tucker (KKT) optimality conditions, which further fosters the idea that the mapping in (4) should be identifiable, was proposed in [8] which is further supported by [39], [40]. The authors stated that for continuous multi-objective problems the Pareto optimal set is piecewise continuous in decision space. This point is revisited in Section VII. In the present work, a radial basis function neural network (RBFNN) is used for this purpose, since it is both robust and accurate for a wide set of problems [41]. The structure and further details regarding the way this type of neural network is employed in this work is discussed in Section IV-B.

However, even if the mapping, $F_{\mathcal{P}}$, was explicitly known, it is still unclear how the desired Pareto optimal objective vectors should be generated in order to obtain their corresponding decision variables, using $F_{\mathcal{P}}$. This problem is related to the issue encountered in Section III-C with the function G. For example, assume that we have the exact mapping $F_{\mathcal{P}}$ for a multi-objective problem, with the only restriction being that we provide the exact coordinates of Pareto optimal points. In order to be able to provide this information, we are required to know exactly the shape of the PF, meaning a mathematical description of the PF hyper-surface must be available for all potential problems. If such information is available for the given problem, then all decision variables corresponding to the PF could be obtained using $F_{\mathcal{P}}$. This point becomes clearer if we view the mapping $F_{\mathcal{P}}$ as the inverse of the objective function $\mathbf{F}^{-1}(\cdot) = F_{\mathcal{P}}(\cdot)$, which leads to

$$F_{\mathcal{P}}(\mathbf{F}(\mathbf{x})) = \mathbf{x}. \tag{5}$$

Even if the function, $\mathbf{F}(\cdot)$, is not a bijection[7] a mapping $G : \mathcal{P} \rightarrow \mathcal{D}$ can still be obtained but can no longer be called the inverse image of, $\mathbf{F}$, however for practical purposes its function would the same. Therefore it is relatively *safe* to ignore for the moment that the objective function $\mathbf{F}(\cdot)$ may be many-to-one, this issue is further discussed in Section VII.

Now, let us assume that we can transform the set $\mathcal{P}$ to a set $\tilde{\mathcal{P}}$, with the only difference being that we can very easily obtain and manipulate the elements in $\tilde{\mathcal{P}}$ and that any element in $\tilde{\mathcal{P}}$ is mapped exactly to one element in the Pareto optimal set $\mathcal{P}$. That is we require the mapping $\Pi^{-1} : \mathcal{P} \rightarrow \tilde{\mathcal{P}}$ to be a bijection. In which case we can obtain the inverse transform $\Pi : \tilde{\mathcal{P}} \rightarrow \mathcal{P}$, and,

$$F_{\mathcal{P}}(\Pi(\tilde{\mathcal{P}})) = \mathcal{D}, \tag{6}$$

would enable the DM to generate any required solution. One way to produce such a mapping is to initially normalize the objective vectors in $\mathcal{P}$ according to,

$$\tilde{f}_i = \frac{f_i - \mathbf{z}_i^\star}{\mathbf{z}_i^{nd} - \mathbf{z}_i^\star}, \tag{7}$$

where $\mathbf{z}^\star$ and $\mathbf{z}^{nd}$ are estimated from the set $\mathcal{P}$. This normalization scales the objectives in the range $[0, 1]$. The Π^{-1}

[6]See Section VII, for an explanation why this mapping is usually *into* and not *onto*.

[7]A function that is an injection and a surjection is a bijection or one-to-one.

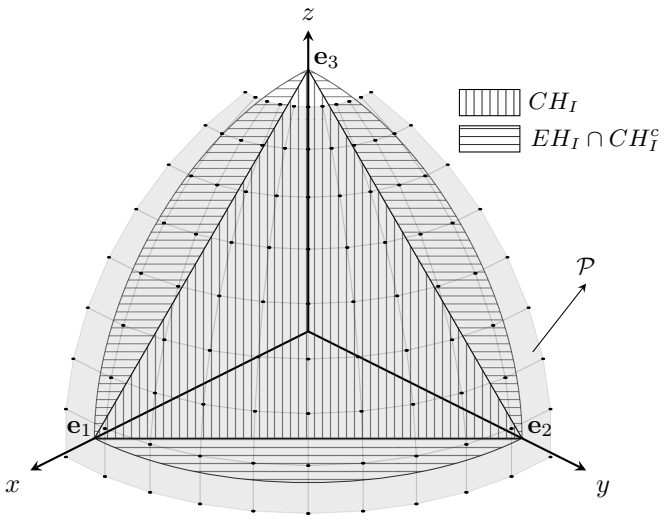

Fig. 3. Illustration of the Π^{-1} mapping for a Pareto set $\mathcal{P}$ with 3 objectives. The points on the outer grid are in $\mathcal{P}$, while the transformed $\tilde{\mathcal{P}}$ set is within the hashed regions.

mapping is illustrated in *Fig. 2*. After the normalization the resulting objective vectors are projected onto EH_I; for problems with two objectives this is the same as CH_I. Subsequently the mapping $\tilde{F}_{\mathcal{P}}$,

$$\tilde{F}_{\mathcal{P}} : \tilde{\mathcal{P}} \rightarrow \mathcal{D} \tag{8}$$

is identified using a RBFNN, as shown in *Fig. 1*. This model in essence subsumes the composition of the mapping $F_{\mathcal{P}}$ and Π in (6).

Π^{-1} effectively takes a set of vectors in $\mathbb{R}^k$, $\mathcal{P}$, and creates its corresponding set in EH_I, $\tilde{\mathcal{P}}$. For two dimensions, vectors in $\tilde{\mathcal{P}}$ will be part of the convex set CH_I and this set will be identical to EH_I, see *Fig. 2*. For more than two dimensions, both EH_I and CH_I are still convex sets, but a more elaborate procedure will be required to obtain points on the EH_I due to its geometry, see *Fig. 3*.

For example, consider a concave Pareto front as the one shown in *Fig. 3*. This front is the first octant of a sphere centred at the origin with radius 1.2. If we apply the Π^{-1} transform to this Pareto optimal set, the resulting $\tilde{\mathcal{P}}$ set will be on the union of the striped areas in *Fig. 3*, i.e. EH_I. The part of $\tilde{\mathcal{P}}$ in CH_I is the set within the triangle with vertices $\mathbf{e}_1$, $\mathbf{e}_2$ and $\mathbf{e}_3$. The remaining points in $\tilde{\mathcal{P}}$ are part of[8] $EH_I \cap CH_I^c$, and, since the edges of the EH_I set are curved it is no longer straightforward to generate points within this set that are evenly distributed. Therefore the desired property of the function, Π, discussed in Section III-C, that is the ability to easily generate points within its domain, would be restricted. A partial solution to this is to simply to bound the domain of definitions of the Π mapping to the CH_I artificially. This would maintain the aforementioned desirable property but such a restriction would limit the method in producing solutions that their projection is within the CH_I. The solutions in $EH_I \cap CH_I^c$ correspond to *extreme* Pareto optimal points which are, potentially, of low interest [42]. However, if this assumption is not true and the decision maker requires solutions within these regions, the method described in Section VI-B could be employed to obtain estimates from the PE method. This can be achieved as the entire, $\tilde{\mathcal{P}}$, set is used in the model creation process (see Section IV-B).

Finally, to generate the estimated Pareto optimal solutions, a set of evenly spaced convex combinations of the set $C = \{\mathbf{e}_1, \ldots, \mathbf{e}_k\}$ is created, let the resulting set be, $\mathcal{E}$. Subsequently this set can used as an input to a model of $\tilde{\mathbf{F}}_{\mathcal{P}}$. The resulting decision vectors may then be used in the objective function to verify that they correspond to Pareto optimal objective vectors. An alternative is to create $\mathcal{E}$ for a specific region of interest in the PF, for example using points that are within the $\mathbf{conv}\, C$.

B. Radial Basis Function Neural Networks

Before we delve into a detailed description of how the proposed method can be applied to decomposition and Pareto-based multi-objective optimization algorithms, we first explore the technique used to model the $\tilde{F}_{\mathcal{P}}$ mapping.

Neural networks, or more precisely artificial neural networks[9], are widely used in an array of different disciplines [43]–[45]. They are well known for their *universal approximator* property [46]. Furthermore, a subclass of NNs, namely radial basis function neural networks (RBFNNs) have been shown to be robust and accurate predictors when compared to Kriging, multivariate adaptive splines and polynomial regression methods [41]. RBFNNs have a single hidden layer and an output layer. Their output layer is often comprised of linear functions since this guarantees a unique solution to their weights w [47] without the need to resort to the renowned back-propagation algorithm [48].

RBFNNs usually employ basis functions that are radially symmetric about their centres μ, for the chosen norm, and decreasing as $\mathbf{x}$ drifts away from μ. A commonly used basis function is the Gaussian [47], given in its general form by,

$$\phi_i(\mathbf{x}) = \exp\left(\frac{\|\mathbf{x} - \mu_i\|^2}{2\sigma_i^2}\right), \tag{9}$$

where the norm $\|\cdot\|$ is often the Euclidean (ℓ_2-norm). Perhaps, at this point a difficulty associated with RBFNNs is evident, namely that, although the output layer is comprised of linear functions, the hidden layer is highly non-linear in the parameters μ and σ, which can prove a challenge in the selection of their optimal values. Various techniques are suggested in the literature addressing this problem [47]. In this work we choose to use all the training data as centres for the radial basis functions, ϕ_i. Therefore the number of basis functions is equal to the number of training vectors used. Additionally, a uniform value for the parameter, σ, is used for all basis functions, and it is set to $5 \cdot \bar{d}_\mu$, where $\bar{d}_\mu$ is the mean distance of solutions in $\tilde{\mathcal{P}}$ to their nearest neighbour. This value for σ was chosen experimentally. Intuitively, this almost guarantees that the basis functions overlap, thus minimizing the number of regions in the interior of the set $\tilde{\mathcal{P}}$ for which no basis function

[8] CH_I^c is the complement of the set CH_I.

[9] Artificial neural networks are simply referred to as neural networks or NNs in this work for convenience.

is *active*. Therefore (9) becomes,

$$\phi_i(\mathbf{x}) = \exp\left(\frac{\|\mathbf{x} - \tilde{\mathcal{P}}_i\|_2^2}{2(5\bar{d}_\mu)^2}\right). \quad (10)$$

Arguably, this is the simplest way to choose the parameters of the basis functions and was used to retain focus on the proposed methodology. For a more elaborate and comprehensive methodology on selection of the parameters of RBFNNs, the reader is referred to [49].

The output of a RBFNN is a linear combination of the basis functions $\phi_i(\cdot)$,

$$y_m(\mathbf{x}) = \sum_{i=0}^{|\tilde{\mathcal{P}}|} w_{m,i}\phi_i(\mathbf{x}), \quad (11)$$

where $\phi_0(\cdot) = 1$ is the output layer bias term and $m \in \{1, \ldots, n\}$, where n is the number of outputs, i.e. the number of decision variables.

To validate the created neural network $(n-1)$-cross validation was used as suggested in [23]. Namely, for a Pareto set of size N, N NNs where created using $(N-1)$ samples for the training set and the remaining sample was used to estimate the generalization error. This procedure is repeated until all the solutions in the Pareto set have been used as a test sample and then the mean square error is calculated. After estimating the NN expected generalization error using cross validation, the final NN is generated using the entire Pareto set.

C. Pareto Dominance Based Algorithms

The method described in Section IV-A introduced the general procedure of the proposed technique, however certain details were abstracted. Optimization algorithms based on Pareto dominance for fitness assignment have several control parameters. One of these parameters is the size of the population to be used in the optimization process. This parameter effectively provides an upper bound on the resulting number of Pareto optimal solutions in the final set $\mathcal{P}$. One requirement for the methodology to function correctly, for the entire PF, is that there be a sufficient number of non-dominated solutions in the final population. An additional requirement, that is evident from experiments, is that the non-dominated set produced by the algorithm is well spread across the PF, i.e. the solutions are diverse and the mean distance from their neighbours has small variance. This simply states that the performance of the proposed method is dependent on the performance of the algorithm used to solve the MOP.

Once the execution of a multi-objective evolutionary algorithm (MOEA) has come to an end, the non-dominated solutions of the resulting set, constitute the set $\mathcal{P}$, with corresponding decision variables $\mathcal{D}$. Then each objective in $\mathcal{P}$ is normalized according to (7) in the range $[0, 1]$ and the ideal and nadir vectors are estimated from the set $\mathcal{P}$ as follows,

$$\mathbf{z}^\star = (\min\{f_1\}, \ldots, \min\{f_k\}), \quad (12)$$

$$\mathbf{z}^{nd} = (\max\{f_1\}, \ldots, \max\{f_k\}), \quad (13)$$

where f_i is the i^{th} objective function and its corresponding values for different solutions are found in the i^{th} column of $\mathcal{P}$. Note that since the produced Pareto set approximation has finite size, the $\inf$ and $\sup$ operators are replaced by the $\min$ and $\max$ operators, which return the minimum and maximum element of a set respectively. Next, the normalized set is projected onto the hyperplane E defined by $\{\mathbf{e}_1, \ldots, \mathbf{e}_{k-1}\}$ where $\mathbf{e}_i$ is a vector of zeros and a one in the i^{th} position. This is achieved by initially projecting onto the subspace[10] parallel to E and then shifting the result by $\frac{1}{k}J_{|\mathcal{P}|,k}$, where $J_{|\mathcal{P}|,k}$ is the $|\mathcal{P}| \times k$ unit matrix. To obtain the projection matrix, $k-1$ linearly independent vectors in the E plane are required. These vectors are obtained in the following way:

$$H = \left(\mathbf{e}_1 - \frac{1}{k}\mathbf{1} \cdots \mathbf{e}_{k-1} - \frac{1}{k}\mathbf{1}\right), \quad (14)$$

where H is a $k \times (k-1)$ matrix. Subsequently the projection matrix P_E is obtained by,

$$P_E = H(H^T H)^{-1} H^T, \quad (15)$$

where P_E is a $k \times k$ matrix with rank $k-1$. The transformed Pareto set $\tilde{\mathcal{P}}$ is,

$$\tilde{\mathcal{P}} = \mathcal{P}P_E^T + \frac{1}{k}J_{|\mathcal{P}|,k}. \quad (16)$$

Finally the neural network used to identify the mapping $\tilde{F}_P$, is created as described in Section IV-B, using $\tilde{\mathcal{P}}$ and $\mathcal{D}$ as the training inputs and outputs respectively.

Once the neural network is trained it can be used to create additional solutions for a new set of convex combinations $\mathcal{E}$. However, this set has to be generated by the DM according to his/her preference in a particular region of the PF; alternatively, a more densely and evenly spaced convex set spanning the entire PF could be created. The first option is likely to be preferred when the cost of evaluating the objective function is considerable or there is a clear preference towards a particular region of the PF.

The described procedure is summarised as follows:

Step 1 Obtain the non-dominated individuals from the final population of a Pareto based MOEA, $\mathcal{P}$, and its corresponding decision variables $\mathcal{D}$.

Step 2 Normalize $\mathcal{P}$ according to (7).

Step 3 Project the normalized $\mathcal{P}$ onto the the $k-1$ hyperplane going through $\{\mathbf{e}_1, \ldots, \mathbf{e}_{k-1}\}$ according to (14), (15) and (16), to produce $\tilde{\mathcal{P}}$. For 2 objectives this is the line through $(0,1)^T$ and $(1,0)^T$, and for 3 objectives, it is the plane through $(1,0,0)^T$, $(0,1,0)^T$ and $(0,0,1)^T$.

Step 4 Identify the mapping $\tilde{F}_{\mathcal{P}}$ using $\tilde{\mathcal{P}}$ and $\mathcal{D}$ as inputs and outputs, respectively, to train a RBFNN as described in Section IV-B.

Step 5 Create the set $\mathcal{E}$, in this work this is a set of evenly spaced convex vectors.

Step 6 Use the set $\mathcal{E}$ as inputs to the NN created in **Step 5**, to obtain estimates of decision vectors $\mathcal{D}_{\mathcal{E}}$.

Step 7 The set $\mathcal{D}_{\mathcal{E}}$ can be used with the objective function $\mathbf{F}(\cdot)$ to verify that the produced solutions are acceptable.

[10]The parallel plane to E that goes through the $\mathbf{0}$ vector.

D. Decomposition-Based Algorithms

Decomposition-based MOEAs have recently increased in popularity, a trend that was reinforced by the introduction of MOEA/D by Zhang and Li [50]. In MOEA/D, the MOP in (1), is decomposed into a set of scalar sub-problems. This is achieved with the help of one of several decomposition techniques, weighted sum, Chebyshev [5] and normal boundary intersection [42] decompositions are some of the available options. The multi-objective optimization problem, see (1), is restated in the following way with the aid of the Chebyshev decomposition,

$$\begin{aligned} \min_{\mathbf{x}} g_\infty(\mathbf{x}, \mathbf{w}^s, \mathbf{z}^\star) &= \|\mathbf{w}^s \circ |\mathbf{F}(\mathbf{x}) - \mathbf{z}^\star| \,\|_\infty, \\ &\forall s = 1, \ldots, N \,, \\ \text{s.t. } &\mathbf{x} \in S, \end{aligned} \tag{17}$$

where $\mathbf{w}^s$ are N evenly distributed weighting vectors and N is the population size and g_∞ is the scalar objective function. The $\circ$ operator denotes the Hadamard product which is element-wise multiplication of vectors or matrices. The intuition behind this is that since g_∞ is a continuous function of $\mathbf{w}$ [50], N evenly distributed weighting vectors should produce a well distributed set of Pareto optimal solutions.

Consequently, since decomposition based algorithms already have a set of convex combinations, namely the weighting vectors $\mathbf{w}$, and the correspondence of weighting vectors to objective vectors is clear, the set $\tilde{\mathcal{P}}$ can be substituted with the weighting vectors $\mathbf{w}$ that produce Pareto optimal solutions. This has the potential to greatly simplify the described procedure in Section IV-C. However, although this simplifies the algorithm, the choice of input vectors, $\mathbf{w}_\mathcal{E}$, by the DM is more difficult because of its indirect nature compared to the general method described in Section IV-C, and this problem becomes increasingly more difficult for increased number of objectives.

Therefore, although the method described for Pareto based algorithms can be applied directly to decomposition based algorithms, if we choose to use the weighting vectors $\mathbf{w}$ instead of creating the set $\tilde{\mathcal{P}}$,

$$\tilde{F}_\mathcal{P} : \mathbf{w} \to \mathcal{D}. \tag{18}$$

Thus, a simplification to the proposed method is available when the MOEA used is based on decomposition, and is summarised as follows:

- **Step 1** Obtain the weighting vectors, $\mathbf{w}$, corresponding to non-dominated solutions.
- **Step 2** Identify the mapping $\tilde{F}_\mathcal{P}$ using $\mathbf{w}$ and $\mathcal{D}$ as inputs and outputs respectively, to train a RBFNN.
- **Step 3** Generate a new set of weighting vectors $\mathbf{w}_\mathcal{E}$ in the PF region of interest, or using one of the methods discussed so far.
- **Step 4** Use the set $\mathbf{w}_\mathcal{E}$ as inputs to the neural network created in **Step 2**, to obtain estimates of decision vectors $\mathcal{D}_\mathbf{w}$.
- **Step 5** The set $\mathcal{D}_\mathbf{w}$ can be used with the objective function $\mathbf{F}(\cdot)$ to verify that the produced solutions are acceptable.

V. Experiment Results

To test the merits of the proposed method, the Pareto-based algorithm was chosen to be NSGA-II [4] and the decomposition-based algorithm was chosen to be MOEA/D [50]. The algorithms were run 50 times, using a different seed for the random number generator on every run, for six MOPs with two and three objectives. The population size used for both algorithms was set to 101 for the two objective problems and to 276 for the three objective problems, as these values are commonly employed in benchmarks [50]. Additionally, the algorithms were allowed to run for 300 generations for the WFG problems and for 500 generations for the DTLZ problems. The DTLZ test problems are, DTLZ1 and DTLZ2 for two and three objectives. For completeness a definition of the DTLZ1–2 test problems is given:

- DTLZ1, see [51]

$$\begin{aligned} f_1(\mathbf{x}) &= (1 + g(\mathbf{x}))x_1x_2, \\ f_2(\mathbf{x}) &= (1 + g(\mathbf{x}))x_1(1 - x_2), \\ f_3(\mathbf{x}) &= (1 + g(\mathbf{x}))(1 - x_1), \\ g(\mathbf{x}) &= 100(n - 2) + \\ &\quad 100 \sum_{i=3}^{n} \left((x_i - 0.5)^2 - \cos(20\pi(x_i - 0.5)) \right), \end{aligned}$$

  where n is the number of decision variables, here $n = 10$. The two dimensional problem is $F(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}))^T$.
- DTLZ2, see [51]

$$\begin{aligned} f_1(\mathbf{x}) &= (1 + g(\mathbf{x})) \cos\left(\frac{x_1\pi}{2}\right) \cos\left(\frac{x_2\pi}{2}\right), \\ f_2(\mathbf{x}) &= (1 + g(\mathbf{x})) \cos\left(\frac{x_1\pi}{2}\right) \sin\left(\frac{x_2\pi}{2}\right), \\ f_3(\mathbf{x}) &= (1 + g(\mathbf{x})) \sin\left(\frac{x_1\pi}{2}\right), \\ g(\mathbf{x}) &= \sum_{i=3}^{n} x_i^2, \end{aligned}$$

  with $n = 10$.

Additionally the test problems WFG2-3 and WFG6-7 from the WFG toolkit [52] were used. The settings used for these test problems can be seen in Table I. The parameters k and l in Table I are the position and distance related parameters respectively. This particular collection of test problems was

TABLE I
Test problem settings summary.

2-Obective Problem Instances					
	# Generations	N	n	k	l
WFG	300	101	24	4	20
DTLZ1	500	101	10	-	-
DTLZ2	500	101	10	-	
3-Objective Problem Instances					
	# Generations	N	n	k	l
WFG	300	276	24	4	20
DTLZ1	500	276	10	-	-
DTLZ2	500	276	10	-	-

chosen with several considerations in mind. First, the problem set had to be broadly used and recognised by the MOEA community. Second, the problems should be challenging and diverse. It is our belief that these aims are accomplished by this particular problem set. It is hoped that future research will provide further validation of the proposed methodology through experiments on more test problems as well as real-world problems. More specifically, DTLZ1 and DTLZ2 [51] have been used in numerous studies [53], [54], [50], something that is also true for the WFG toolkit [53], [54]. Furthermore, each of these problems pose a different challenge. For instance, WFG2 has a discontinuous Pareto front and is non-separable. WFG3 is also non-separable and its Pareto front is linear for two dimensions and degenerate for three or more. WFG6 has a concave Pareto front and is non-separable and unimodal; and, lastly, WFG7 is separable with a concave Pareto front and has parameter dependent bias [52]. The settings for the two algorithms were chosen in a similar fashion.

The hypothesis of this paper is that by using the Pareto estimation methodology the number of Pareto optimal solutions available to the DM can be increased significantly, and despite the fact that on many test instances the estimated Pareto set actually turns out to be superior to the initial set this is not the intended purpose of the method and can be treated as a positive side effect. For performance assessment purposes, the ratio of the following indices was used as - our main focus is the relative quality of the Pareto set, produced by MOEA/D and NSGA-II, before the application of the proposed method and after - and not the performance of the employed algorithms in absolute terms.

- Inverted Generational Distance (IGD), introduced in [55],

$$D(A,\mathcal{P}^\star)=\frac{\sum\limits_{s\in\mathcal{P}^\star}\min\{\|A_1-s\|_2,\ldots,\|A_N-s\|_2\}}{|\mathcal{P}^\star|}, \tag{19}$$

  where $|\mathcal{P}^\star|$ is the cardinality of the set $\mathcal{P}^\star$ and A is an approximation of the PF. The IGD metric measures the distance of the elements in the set A from the nearest point of the actual PF. The ratio of this metric was used as,

$$D_R(A,B)=\frac{D(A,\mathcal{P}^\star)}{D(B,\mathcal{P}^\star)}, \tag{20}$$

  where B is another PF approximation set. In this work B is the estimated PF using the Pareto estimation methodology.
- Mean Distance to Nearest Neighbour,

$$S(A)=\frac{\sum\limits_{i=1}^{|A|}d_i}{|A|}, \tag{21}$$

  where d_i is,

$$d_i=\min_j\{\|f_1(\mathbf{x}_i)-f_1(\mathbf{x}_j)\|_2+\cdots+\|f_k(\mathbf{x}_i)-f_k(\mathbf{x}_j)\|_2\}.$$

  This metric can serve as a measure of the density of solutions. Again, the ratio of this metric is used as,

$$S_R(A,B)=\frac{S(A)}{S(B)}. \tag{22}$$

and the coverage metric, described below, was used exactly as defined in [56],

- Coverage Metric (C-Metric)

$$C(A,B)=\frac{|\{u\in B|\exists v\in A: v\preceq u\}|}{|B|}, \tag{23}$$

  $C(A,B)=0$ is interpreted as: there is no solution in A that dominates any solution in B. And $C(A,B)=1$ is interpreted as the exact opposite, i.e. all the solutions in B are dominated by at least one solution in A.

A. Pareto Dominance Based Algorithms

For every run of NSGA-II, with settings as explained in Section V, the proposed method was applied using an evenly spaced convex set $\mathcal{E}$ of size $\sim$10 times greater than the initial population used in the optimization algorithm. The set $\mathcal{E}$ was used as input to the identified mapping $\tilde{F}_{\mathcal{P}}$ resulting in the estimated decision vectors $\mathcal{D}_{\mathcal{E}}$. Subsequently $\mathcal{D}_{\mathcal{E}}$ was used with the objective function generating the objective vectors $\mathcal{P}_{\mathcal{E}}$.

Specifically, for the 2-objective test problems, the size of the set $\mathcal{P}_{\mathcal{E}}$ was set to $1\,000$ and for the 3-objective problems the size of the set $\mathcal{P}_{\mathcal{E}}$ was set to $3\,003$. The original Pareto optimal solutions used in the estimation process can be seen in *Fig. 6* and *Fig. 8*, and the corresponding estimates $\mathcal{P}_{\mathcal{E}}$ are shown in *Fig. 7* and *Fig. 9*. It should also be noted, as is perhaps apparent from the figures, that the entire estimated population $\mathcal{P}_{\mathcal{E}}$ is presented and not a non-dominated subset. The same procedure was performed for all 50 runs of NSGA-II for all test problems for two and three objectives and the results are summarized in Table II-Table V and their non-parametric counterparts are presented in *Fig. 4*. Furthermore, the number of valid solutions produced by the RBFNN, the number of Pareto optimal solutions and the RBFNN estimated generalization error using cross validation (see Section IV-B), are presented in *Fig. 5*.

Table II presents the ratio of the IGD index $D_R(\mathcal{P},\mathcal{P}_{\mathcal{E}})$, and the mean distance to the nearest neighbour $S_R(\mathcal{P},\mathcal{P}_{\mathcal{E}})$ for problems with 2 objectives. The IGD index, in principle, attains smaller values the closer the set under testing is to the known PF. Additionally if the set does not cover certain regions of the PF, this will cause the value of the IGD index to increase, signifying a degraded performance. Therefore, for this problem set, the proposed methodology is consistent in producing solutions that are at least of the same distance from the actual PF. Values of $D_R(\mathcal{P},\mathcal{P}_{\mathcal{E}})>1$ mean that the set $\mathcal{P}_{\mathcal{E}}$ produce better values for the IGD index compared to the original set $\mathcal{P}$, and for $D_R(\mathcal{P},\mathcal{P}_{\mathcal{E}})<1$ the converse is true. Regarding the mean nearest neighbour distance ratio $S_R(\mathcal{P},\mathcal{P}_{\mathcal{E}})$, values of $S_R(\mathcal{P},\mathcal{P}_{\mathcal{E}})>1$ mean that the mean distance from a solution to its nearest neighbour is smaller in $\mathcal{P}_{\mathcal{E}}$ compared to $\mathcal{P}$, and for $S_R(\mathcal{P},\mathcal{P}_{\mathcal{E}})<1$ the converse is true. In all cases the mean distance of neighbouring solutions in $\mathcal{P}_{\mathcal{E}}$ is much smaller, this fact combined with the results for $D_R(\mathcal{P},\mathcal{P}_{\mathcal{E}})$ strongly indicates that the density of the available Pareto optimal solutions has significantly increased using our proposed method.

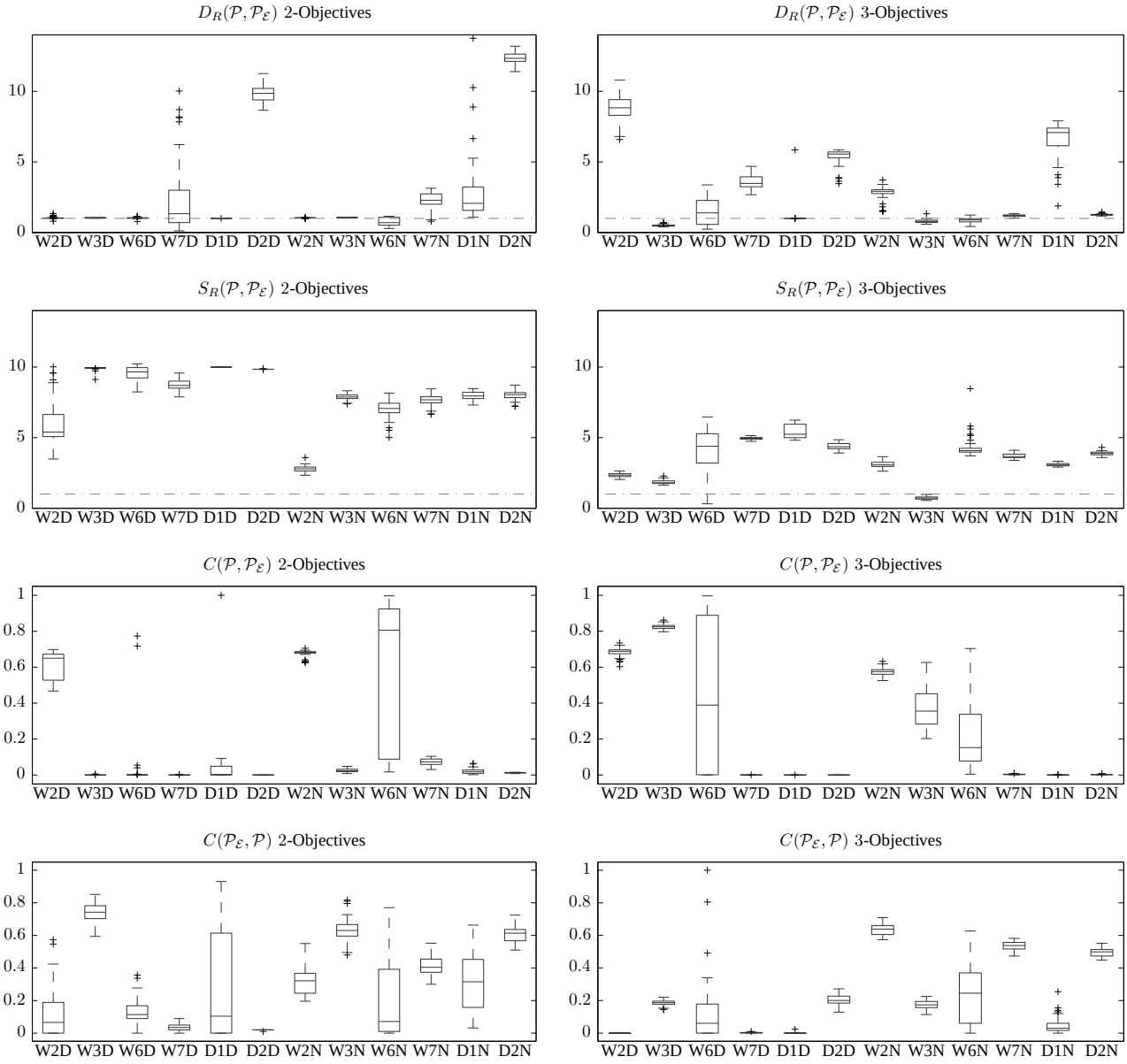

Fig. 4. Boxplots of the experiment results of the Pareto estimation method using Pareto set approximations generated by MOEA/D and NSGA-II. The labels have the following format *Problem family:Problem number:Algorithm used*, where *W* refers to the WFG problem set and *D* to the DTLZ problem set. Also the postfix *D* means that the Pareto set used was produced by MOEA/D, while *N* by NSGA-II. For example the label *W6N* refers to results obtained for the WFG6 test problems using NSGA-II. The horizontal line in the top 4 plots marks the value 1.

TABLE II
$D_R(\mathcal{P},\mathcal{P}_\mathcal{E})$ AND $S_R(\mathcal{P},\mathcal{P}_\mathcal{E})$ VALUES OF THE SOLUTIONS FOUND BY NSGA-II, $\mathcal{P}$, AND THE ESTIMATED SET $\mathcal{P}_E$, FOR THE 2-OBJECTIVE PROBLEM SET.

	$D_R(\mathcal{P},\mathcal{P}_\mathcal{E})$			$S_R(\mathcal{P},\mathcal{P}_\mathcal{E})$		
Problem	min	mean	std	min	mean	std
WFG2	0.9879	1.0370	0.0174	2.3355	2.7844	0.2177
WFG3	1.0488	1.0589	0.0046	7.3917	7.8959	0.2286
WFG6	0.2834	0.7504	0.2730	5.0093	7.0383	0.6354
WFG7	0.7962	2.2765	0.5875	6.6541	7.6369	0.3695
DTLZ1	1.0772	4.0822	8.6262	7.3109	7.9676	0.2790
DTLZ2	11.3970	12.3377	0.3990	7.2193	8.0198	0.3061

TABLE III
C-METRIC VALUES OF THE SOLUTIONS FOUND BY NSGA-II, $\mathcal{P}$, AND THE ESTIMATED SET $\mathcal{P}_E$, FOR THE 2-OBJECTIVE INSTANCES OF THE SELECTED PROBLEM SET.

	$C(\mathcal{P},\mathcal{P}_\mathcal{E})$			$C(\mathcal{P}_\mathcal{E},\mathcal{P})$		
Problem	min	mean	std	min	mean	std
WFG2	0.6244	0.6789	0.0153	0.1959	0.3154	0.0756
WFG3	0.0080	0.0253	0.0096	0.4796	0.6306	0.0761
WFG6	0.0170	0.6046	0.3884	0.0000	0.1778	0.2115
WFG7	0.0305	0.0726	0.0185	0.3000	0.4150	0.0551
DTLZ1	0.0020	0.0192	0.0139	0.0316	0.3222	0.1814
DTLZ2	0.0080	0.0122	0.0020	0.5102	0.6108	0.0494

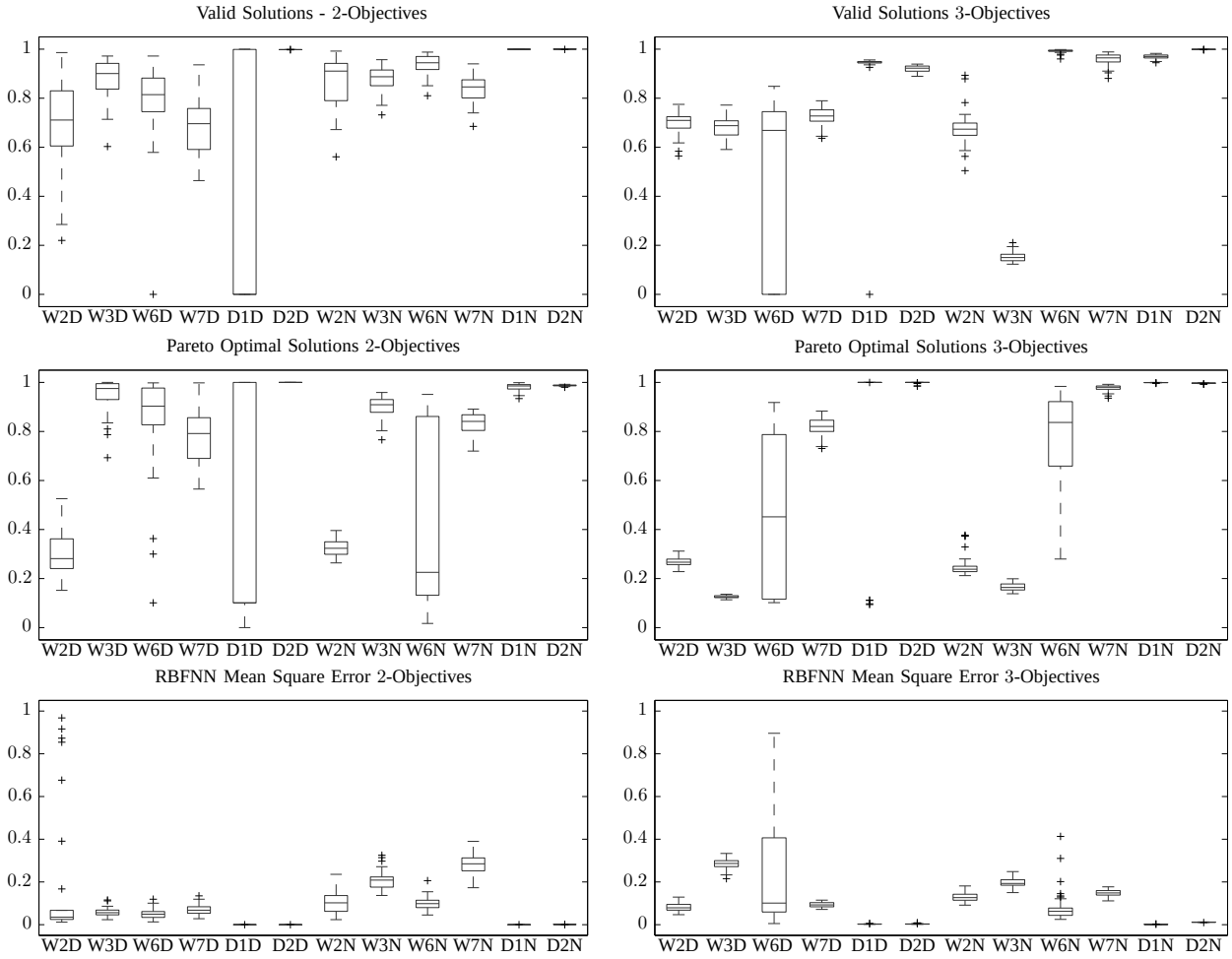

Fig. 5. **Top Row:** The number of valid solutions produced by the RBFNN in the Pareto estimation method for 2 and 3-objective problems instances, normalized to the $[0, 1]$ interval. So a value of 1 means that all produced solutions are valid, and a value of 0 that no valid solution was produced. **Middle Row:** Number of Pareto optimal solutions generated by the RBFNN in the PE method, here too the values are normalized to the $[0, 1]$ interval. **Bottom Row:** The mean square error (MSE) of the RBFNN. Note that all outputs of the NN have been normalized to the $[0, 1]$ interval before calculating the MSE. The labels on the x-axis have the same interpretation as *Fig. 4*.

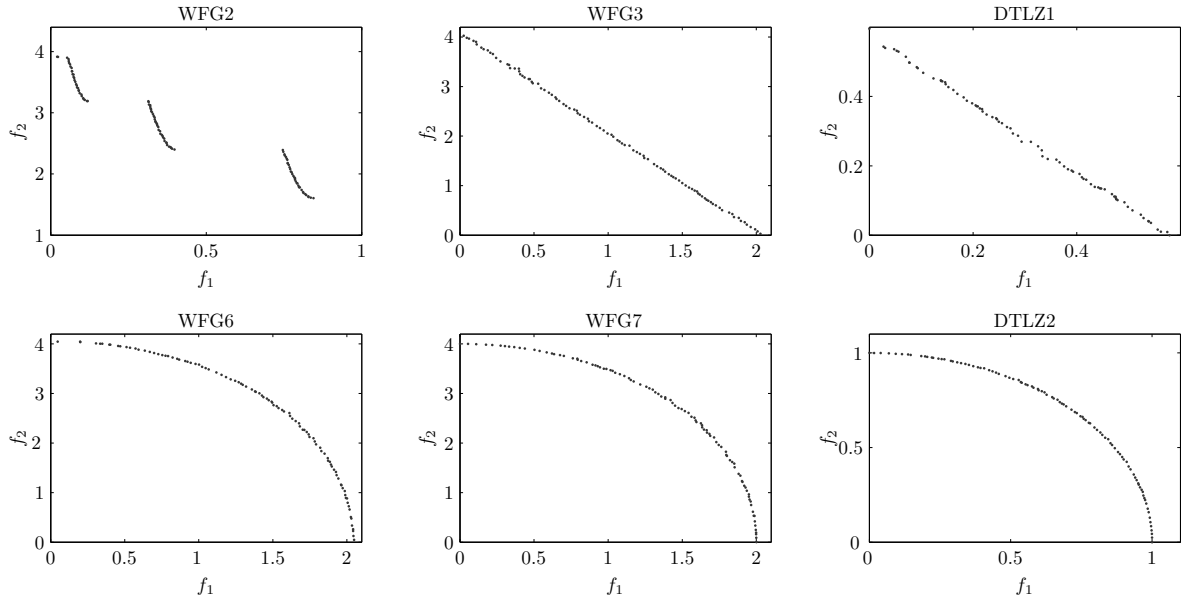

Fig. 6. Pareto front solutions found by NSGA-II for the 2-objective problem set.

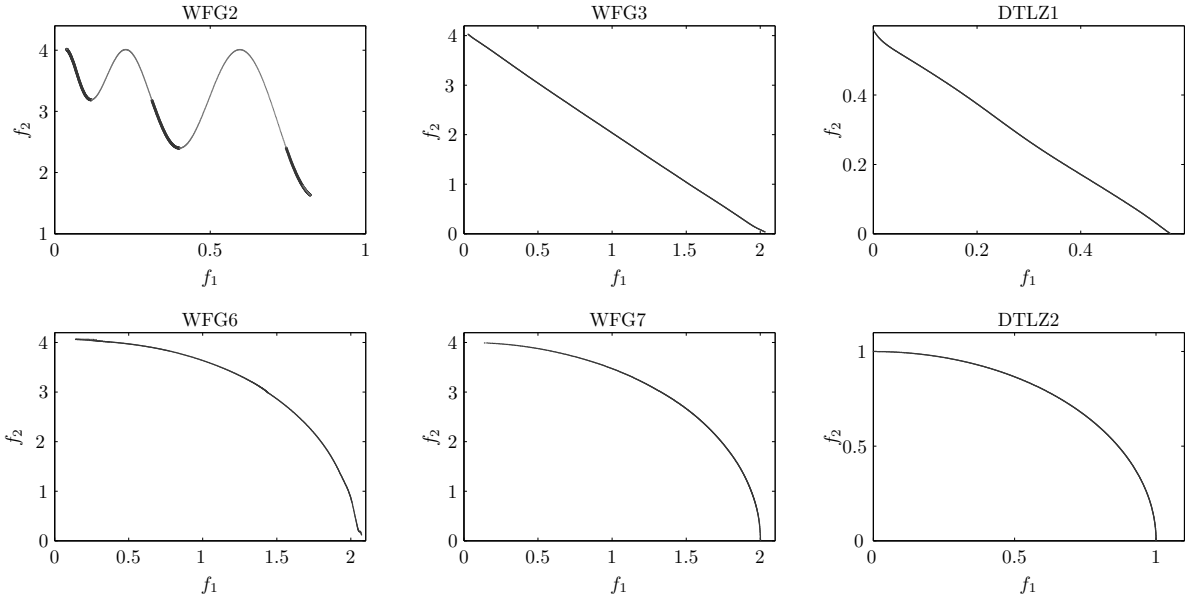

Fig. 7. Estimated solutions $\mathcal{P}_E$ ($|\mathcal{P}_E| = 1\,000$) from the non-dominated solutions found by NSGA-II for the 2-objective problem set. The dominated solutions, for the WFG2 problem, are drawn in gray.

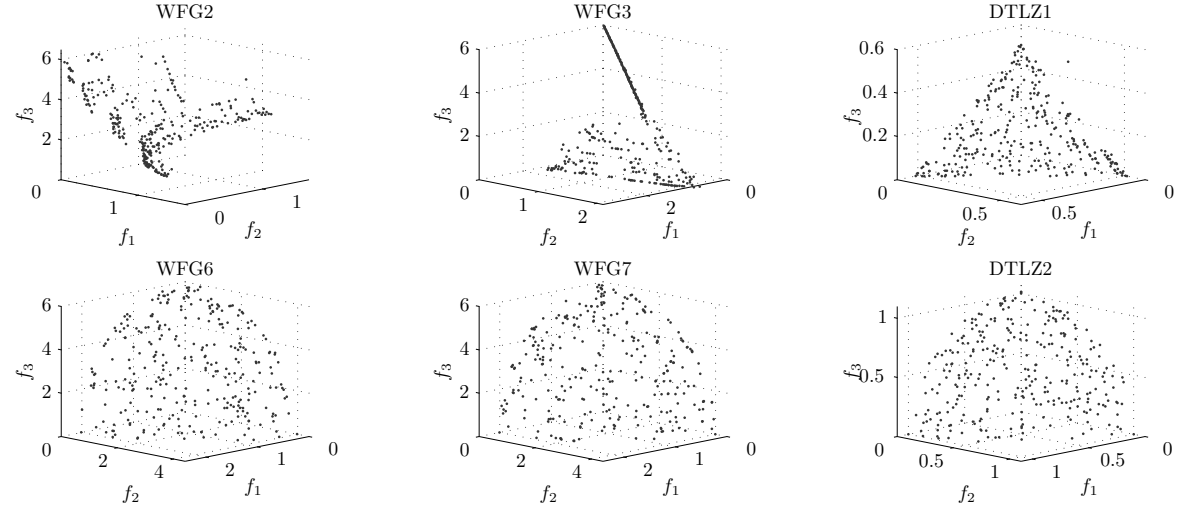

Fig. 8. Pareto front solutions found by NSGA-II for the 3-objective problem set.

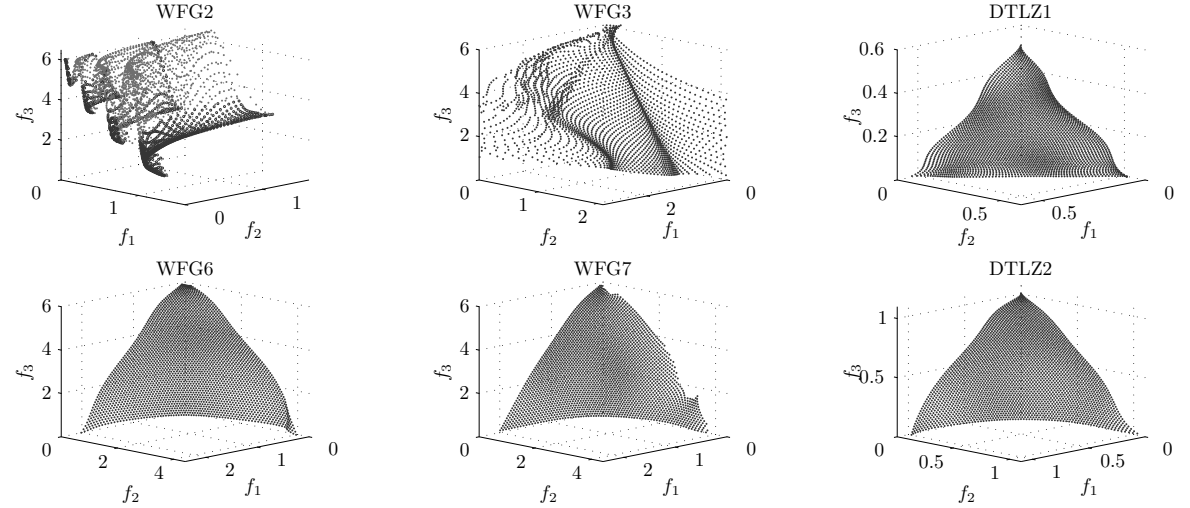

Fig. 9. Estimated solutions $\mathcal{P}_E$ ($|\mathcal{P}_E| = 3\,003$) from the non-dominated solutions found by NSGA-II for the 3-objective problem set. The non-dominated solutions in the WFG2 test problem are the represented by the darker points on the plot.

TABLE IV
$D_R(\mathcal{P}, \mathcal{P}_\mathcal{E})$ AND $S_R(\mathcal{P}, \mathcal{P}_\mathcal{E})$ VALUES OF THE SOLUTIONS FOUND BY NSGA-II, $\mathcal{P}$, AND THE ESTIMATED SET $\mathcal{P}_E$, FOR THE 3-OBJECTIVE PROBLEM SET.

	$D_R(\mathcal{P}, \mathcal{P}_\mathcal{E})$			$S_R(\mathcal{P}, \mathcal{P}_\mathcal{E})$		
Problem	min	mean	std	min	mean	std
WFG2	1.4909	2.8047	0.4878	2.6249	3.1172	0.2306
WFG3	0.5846	0.8013	0.1292	0.5519	0.7331	0.1036
WFG6	0.4242	0.8790	0.1679	3.7070	4.2835	0.7461
WFG7	1.0079	1.1950	0.0720	3.3899	3.7008	0.1752
DTLZ1	1.8795	6.6262	1.2512	2.9125	3.0811	0.1057
DTLZ2	1.1649	1.2655	0.0587	3.5877	3.8767	0.1289

TABLE V
C-METRIC VALUES OF THE SOLUTIONS FOUND BY NSGA-II, $\mathcal{P}$, AND THE ESTIMATED SET $\mathcal{P}_E$, FOR THE 3-OBJECTIVE INSTANCES OF THE SELECTED PROBLEM SET.

	$C(\mathcal{P}, \mathcal{P}_\mathcal{E})$			$C(\mathcal{P}_\mathcal{E}, \mathcal{P})$		
Problem	min	mean	std	min	mean	std
WFG2	0.5253	0.5753	0.0197	0.5738	0.6342	0.0348
WFG3	0.2026	0.3773	0.1153	0.1141	0.1739	0.0262
WFG6	0.0043	0.2137	0.1768	0.0000	0.2417	0.1887
WFG7	0.0002	0.0035	0.0019	0.4733	0.5360	0.0280
DTLZ1	0.0000	0.0003	0.0004	0.0000	0.0463	0.0468
DTLZ2	0.0011	0.0027	0.0012	0.4482	0.4952	0.0264

In Table III the results for the C-metric are given for $C(\mathcal{P}, \mathcal{P}_\mathcal{E})$ and $C(\mathcal{P}_\mathcal{E}, \mathcal{P})$ for the 2-objective test problems. This metric was employed to further verify the consistency of the method. And as can be seen for all problems, except WFG2 and WFG6, the results are favourable. However it is interesting to explore the potential reasons for the less impressive performance in these two problems. Regarding WFG2, since we did not use only the non-dominated subset of $\mathcal{P}_\mathcal{E}$, the identified PF is, as can be seen in *Fig. 7*, an oscillating function; this is exactly the PF directly obtained from the WFG2 problem. Therefore, in a way, the method did actually perform rather well in identifying the front. A remedy to avoid such a behaviour would be that the requested solutions $\mathcal{E}$ are reasonably close to the transformed set $\tilde{\mathcal{P}}$ of the original Pareto optimal solutions $\mathcal{P}$, more elaborate methods are left for future research. And regarding the test problem WFG6, combined with the same moderate results in Table II, it seems that our methodology has consistent difficulties with this particular problem instance. A potential cause for these difficulties is perhaps the simplicity of the modelling technique.

Table IV presents $D_R(\mathcal{P}, \mathcal{P}_\mathcal{E})$ and $S_R(\mathcal{P}, \mathcal{P}_\mathcal{E})$ indices for the 3-objective case. Again $D_R(\mathcal{P}, \mathcal{P}_\mathcal{E})$ has acceptable values, meaning that there is no significant sign of degradation of the IGD index. $S_R(\mathcal{P}, \mathcal{P}_\mathcal{E})$ shows that the mean neighbour distance is consistently lower for $\mathcal{P}_\mathcal{E}$. One noticeable feature for the values of $S_R(\mathcal{P}, \mathcal{P}_\mathcal{E})$ is that they are almost half of their counterparts for the 2-objective case, as seen in Table II. This can partly be attributed to the *curse of dimensionality*, in the sense that to obtain similar results to Table II, we have to produce approximately $\mathcal{O}(n^2)$ order of solutions more than for the 2-objective case. This is not the case for the 3-objective instances of WFG2 and WFG3, which is mainly due to their PF geometry.

In Table V the results for the C-metric are given for $C(\mathcal{P}, \mathcal{P}_\mathcal{E})$ and $C(\mathcal{P}_\mathcal{E}, \mathcal{P})$ for the 3-objective problems. Again the results are consistent, with WFG2 performing rather moderately for the same reasons as for the 2-objective case. The surprising fact is that for the 3-objective WFG6 performs extremely well.

B. Decomposition Based Algorithms

The same experimental procedure as in Section V-A is applied for the decomposition based version of the MOEA. As previously mentioned, for this test case MOEA/D [50] was used with the same population size as NSGA-II. Instead of a set $\mathcal{E}$, an evenly distributed set of weighting vectors $\mathbf{w}_\mathcal{E}$ was used, as described in Section IV-D. In all other respects the experimental setup is identical. The original Pareto optimal solutions used in the estimation process can be seen in *Fig. 10* and *Fig. 12*, and the corresponding estimates $\mathcal{P}_\mathcal{E}$ in *Fig. 11* and *Fig. 13*. As before, the entire estimated population $\mathcal{P}_\mathcal{E}$ is presented and used for the calculation of the statistical results. Also the results are summarised in *Fig. 4* and *Fig. 5*.

Table VI presents the ratio of the IGD index $D_R(\mathcal{P}, \mathcal{P}_\mathcal{E})$, and the mean distance to the nearest neighbour $S_R(\mathcal{P}, \mathcal{P}_\mathcal{E})$ for problems with 2 objectives. A distinctive pattern, when compared with the corresponding values in Table II, is that when the $D_R(\mathcal{P}, \mathcal{P}_\mathcal{E})$ index is very close to 1 the mean value for $S_R(\mathcal{P}, \mathcal{P}_\mathcal{E})$ is very close to 10, which is almost equal to the scaling factor we chose to increase the size of the set $\mathcal{P}_\mathcal{E}$ relative to the initial set $\mathcal{P}$. One possible reason for this behaviour, which no doubt is desirable, is that the solutions produced by MOEA/D are very well distributed across the PF. If we view the 2-dimensional PF as a function, the fact that solutions are well distributed can be seen as sampling the function at regular intervals, hence their mean distance has low variance. This enables the modelling technique we used to better estimate the mapping, since a uniform σ value was chosen for all the basis functions, see Section IV-B. Another interesting fact is that, although the minimum value of $D_R(\mathcal{P}, \mathcal{P}_\mathcal{E})$ for the problem WFG2, is less than 1, the mean value is 1.0341 and the standard deviation is relatively small. This indicates that, in general, the performance of our method is producing good results with low deviations, for this problem instance.

In Table VII the results for the C-metric are given for $C(\mathcal{P}, \mathcal{P}_\mathcal{E})$ and $C(\mathcal{P}_\mathcal{E}, \mathcal{P})$ for the 2-objective test problems. The results are very consistent, for all problems except WFG2 which is to be expected due to the shape of its PF. $C(\mathcal{P}, \mathcal{P}_\mathcal{E})$ is very close to 0, signifying that a very small number of the solutions in $\mathcal{P}_\mathcal{E}$ are dominated by solutions in the original Pareto set $\mathcal{P}$.

Table VIII presents $D_R(\mathcal{P}, \mathcal{P}_\mathcal{E})$ and $S_R(\mathcal{P}, \mathcal{P}_\mathcal{E})$ indices for the 3-objective case. In line with the results in Table VI the $D_R(\mathcal{P}, \mathcal{P}_\mathcal{E})$ index is satisfactory. Although, for problems WFG3 and WFG6 it seems to be somewhat low. This is to a certain extent also reflected in the $S_R(\mathcal{P}, \mathcal{P}_\mathcal{E})$ index. This behaviour, regarding problem WFG3, can be attributed to the fact that the real PF was not successfully identified by

TABLE VI
$D_R(\mathcal{P}, \mathcal{P}_\mathcal{E})$ AND $S_R(\mathcal{P}, \mathcal{P}_\mathcal{E})$ VALUES OF THE SOLUTIONS FOUND BY MOEA/D, $\mathcal{P}$, AND THE ESTIMATED SET $\mathcal{P}_E$, FOR THE 2-OBJECTIVE PROBLEM SET.

	$D_R(\mathcal{P}, \mathcal{P}_\mathcal{E})$			$S_R(\mathcal{P}, \mathcal{P}_\mathcal{E})$		
Problem	min	mean	std	min	mean	std
WFG2	0.7984	1.0341	0.0742	3.4859	6.1070	1.6286
WFG3	1.0037	1.0401	0.0216	9.1222	9.9152	0.1197
WFG6	0.7774	1.0288	0.0514	8.2379	9.5327	0.5143
WFG7	0.1164	2.3724	2.5573	7.8871	8.7620	0.3967
DTLZ1	1.0000	1.0000	0.0000	9.9820	9.9932	0.0049
DTLZ2	8.6618	9.8542	0.6573	9.8320	9.8454	0.0062

TABLE VII
C-METRIC VALUES OF THE SOLUTIONS FOUND BY MOEA/D, $\mathcal{P}$, AND THE ESTIMATED SET $\mathcal{P}_E$, FOR THE 2-OBJECTIVE INSTANCES OF THE SELECTED PROBLEM SET.

	$C(\mathcal{P}, \mathcal{P}_\mathcal{E})$			$C(\mathcal{P}_\mathcal{E}, \mathcal{P})$		
Problem	min	mean	std	min	mean	std
WFG2	0.4675	0.6101	0.0787	0.0000	0.1174	0.1405
WFG3	0.0000	0.0003	0.0009	0.5941	0.7430	0.0591
WFG6	0.0000	0.0323	0.1474	0.0000	0.1289	0.0745
WFG7	0.0000	0.0001	0.0003	0.0000	0.0362	0.0209
DTLZ1	0.0000	0.0441	0.1414	0.0000	0.2875	0.3366
DTLZ2	0.0000	0.0000	0.0000	0.0099	0.0196	0.0014

the algorithm, which for WFG3 is a line in 3-dimensions. This conclusion is further supported by the fact that the corresponding values for $C(\mathcal{P}, \mathcal{P}_\mathcal{E})$ in Table IX are very close to 0 attesting to the fact that the produced estimated Pareto set $\mathcal{P}_\mathcal{E}$, does in fact model the given set rather well. Therefore this behaviour could be remedied by choosing the non-dominated solutions in the set $\mathcal{P}_\mathcal{E}$. However for our purposes this option was avoided since this would *mask* such deficiencies, thus disallowing further insight for possible improvements of the proposed methodology.

In Table IX the results for the C-metric are given for $C(\mathcal{P}, \mathcal{P}_\mathcal{E})$ and $C(\mathcal{P}_\mathcal{E}, \mathcal{P})$ for the 3-objective problems.

VI. PARETO ESTIMATION APPLIED TO PORTFOLIO OPTIMIZATION

The seminal work of Markowitz [57] changed drastically the way that managers and investors decide on what portfolio of securities is appropriate for a given tolerance of risk. The main idea is that given a portfolio composition, there are two

TABLE VIII
$D_R(\mathcal{P}, \mathcal{P}_\mathcal{E})$ AND $S_R(\mathcal{P}, \mathcal{P}_\mathcal{E})$ VALUES OF THE SOLUTIONS FOUND BY MOEA/D, $\mathcal{P}$, AND THE ESTIMATED SET $\mathcal{P}_E$, FOR THE 3-OBJECTIVE PROBLEM SET.

	$D_R(\mathcal{P}, \mathcal{P}_\mathcal{E})$			$S_R(\mathcal{P}, \mathcal{P}_\mathcal{E})$		
Problem	min	mean	std	min	mean	std
WFG2	6.5869	8.8543	0.8840	2.0372	2.3451	0.1331
WFG3	0.3963	0.4957	0.0709	1.6416	1.8425	0.1406
WFG6	0.2327	1.4326	0.8999	0.3148	4.3185	1.2262
WFG7	2.6739	3.5314	0.4975	4.7421	4.9538	0.0914
DTLZ1	1.0003	1.0983	0.6850	4.8150	5.4396	0.5015
DTLZ2	3.4609	5.3924	0.5535	3.9083	4.3850	0.2362

TABLE IX
C-METRIC VALUES OF THE SOLUTIONS FOUND BY MOEA/D, $\mathcal{P}$, AND THE ESTIMATED SET $\mathcal{P}_E$, FOR THE 3-OBJECTIVE INSTANCES OF THE SELECTED PROBLEM SET.

	$C(\mathcal{P}, \mathcal{P}_\mathcal{E})$			$C(\mathcal{P}_\mathcal{E}, \mathcal{P})$		
Problem	min	mean	std	min	mean	std
WFG2	0.4675	0.6101	0.0787	0.0000	0.1174	0.1405
WFG3	0.0000	0.0003	0.0009	0.5941	0.7430	0.0591
WFG6	0.0000	0.0323	0.1474	0.0000	0.1289	0.0745
WFG7	0.0000	0.0001	0.0003	0.0000	0.0362	0.0209
DTLZ1	0.0000	0.0441	0.1414	0.0000	0.2875	0.3366
DTLZ2	0.0000	0.0000	0.0000	0.0099	0.0196	0.0014

main objectives to be considered. First, the expected return is to be maximized and second, the variance of the expected return is to be minimized. Variance of a portfolio allocation is essentially a metric of risk. What was shown by Markowitz is that these two objectives are competing, namely if an investor wants extremely high expected returns, then he or she must concede a high level of risk which could mean that the chance for the entire portfolio to be diminished is increased. Although not without its critics, Markowitz portfolio theory has taken by storm the financial markets and is today employed virtually by every investor. However, a strong critique of this approach in selecting an *optimal* allocation of a portfolio of stocks is that the measure of risk, namely the variance of the portfolio, is not entirely realistic due to the assumption that the expected returns are normally distributed. This assumption is usually not entirely true, and as it can be seen by the recent market crash, can often prove to be devastatingly flawed.

A. Portfolio Optimization - Problem Definition

The classical portfolio optimization problem extended with an additional measure of risk as a third objective, namely the value-at-risk (VaR), can be defined as,

$$\begin{aligned} &\min_{\mathbf{x}} \mathbf{F}(\mathbf{x}) = (R(\mathbf{x}), V(\mathbf{x}), M(\mathbf{x}))\,, \\ &\text{subject to } \sum_{i=1}^{n} x_i = 1, \text{ and } x_i \geq 0,\ i = 1, \ldots, n, \end{aligned} \tag{24}$$

where the decision vector $\mathbf{x}$ represents the allocation of capital on n securities. The constraint imposed on the decision vector in (24) means that no gearing[11]is allowed as the maximum allocation must be equal to the available capital and the non-negativity constraint in the allocation (decision vector components) means that short positions are not allowed. A short position is one in which the investor *borrows* a security and sells it, in hope that he can later buy it at a lower price, repay the loan by returning the security to the lender and make a profit from the difference. Furthermore the scalar objective functions in (24) are the negative of the expected return, $R(\mathbf{x})$, the portfolio variance, $V(\mathbf{x})$, and the value at risk calculated from historical data, $M(\mathbf{x})$. The problem defined in (24) closely follows the formulation used in [19]. However contrary to the work in [19] we employ a non-parametric method to calculate the portfolio VaR, instead of using the

[11]Gearing or leveraging is when securities are purchased on credit.

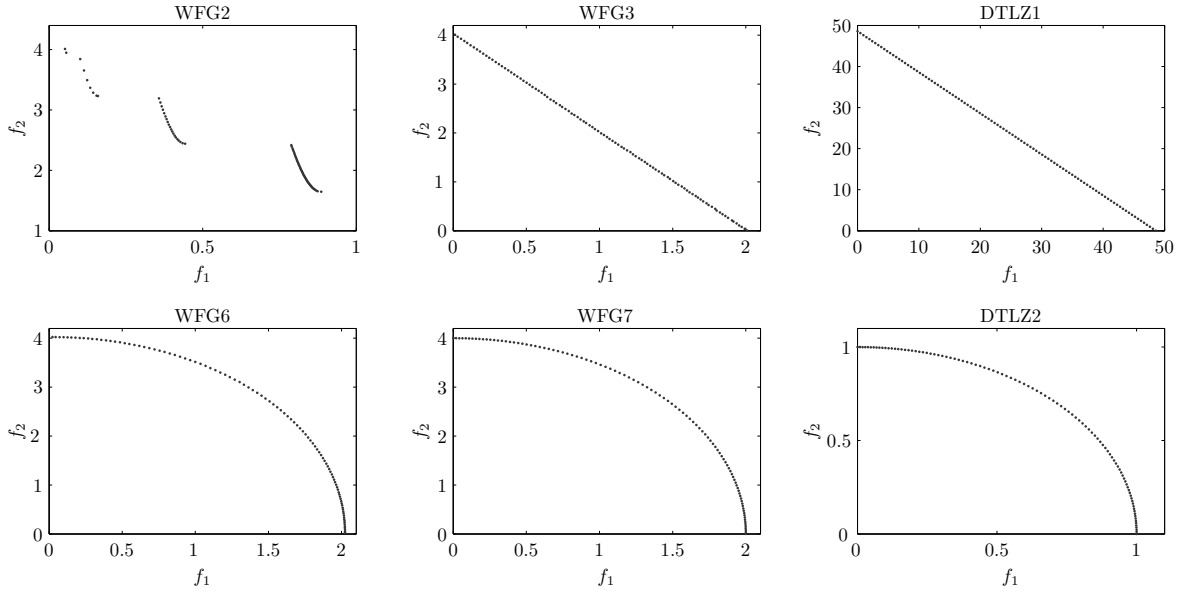

Fig. 10. Pareto front solutions found by MOEA/D for the 2-objective problem set.

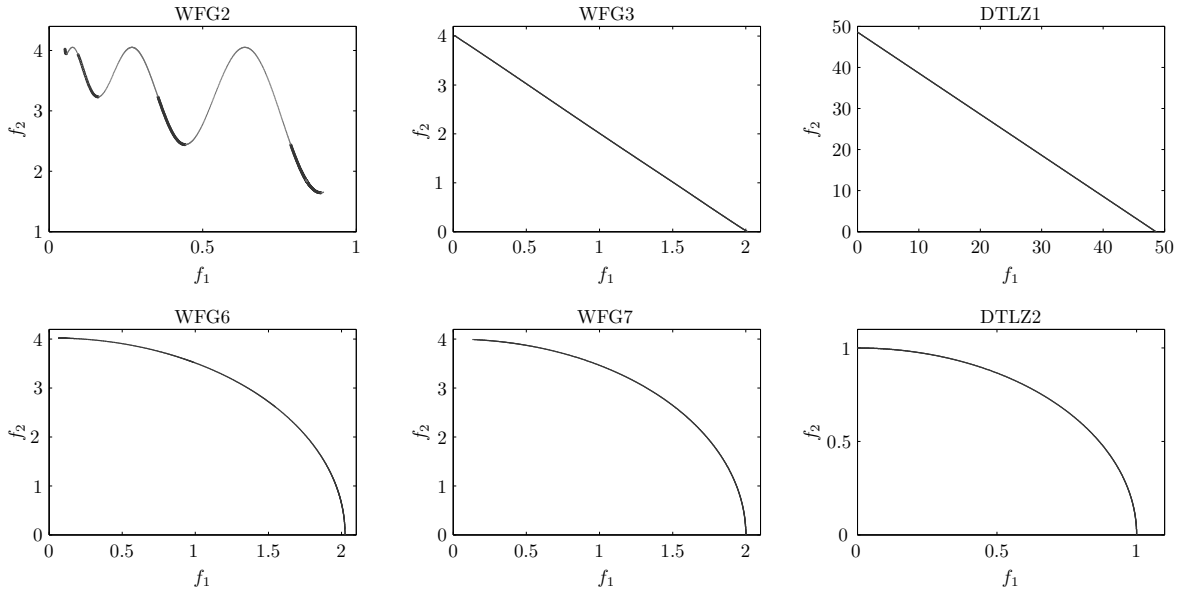

Fig. 11. Estimated solutions $\mathcal{P}_E$ ($|\mathcal{P}_E| = 1\,000$) from the non-dominated solutions found by MOEA/D for the 2-objective problem set. The parts of the PF, for the WFG2 problem, drawn in gray represent dominated solutions.

simplified VaR. Specifically these objectives are defined as follows,

$$R(\mathbf{x}) = -\frac{1}{N-1}\sum_{i=1}^{N-1}\ln\left(\frac{\mathbf{x}^T\mathbf{r}_{i+1}}{\mathbf{x}^T\mathbf{r}_i}\right), \qquad \mathbf{r}_i = (r^i_{s_1},\ldots,r^i_{s_n}), \tag{25}$$

where $r^i_{s_n}$ is the return of the security s_n at time i. The expression in (25) represents the negative of the expected compounded return. The second objective, namely the portfolio variance, is defined as:

$$V(\mathbf{x}) = \mathbf{x}^T\Sigma\mathbf{x}, \tag{26}$$

where, Σ, is the covariance matrix of the underlying securities. The covariance matrix is calculated using historic data, as is the case for the value-at-risk, see Section VI-C. Lastly the third objective is the value-at-risk calculated by a non-parametric method via historic simulation, see for example [58], [59],

$$M(\mathbf{x}) = VaR^{t+1}_\alpha,$$
$$VaR^{t+1}_\alpha = -\inf_y\left\{y\in\mathbb{R} : P\left(\ln\left(\frac{\mathbf{x}^T\mathbf{r}_{t+1}}{\mathbf{x}^T\mathbf{r}_t}\right)\le y\right)\ge\alpha\right\},$$
$$\text{if } VaR^{t+1}_\alpha < 0, \text{ then } M(\mathbf{x}) = 0,$$
$$\alpha\in(0,1), \tag{27}$$

where α is the probability of a return smaller than y. In essence VaR quantifies the potential loss in a portfolio with probability α. Also if $M(\mathbf{x})$ becomes negative, this translates to positive returns ($y > 0$) in the *worst case* scenario, which

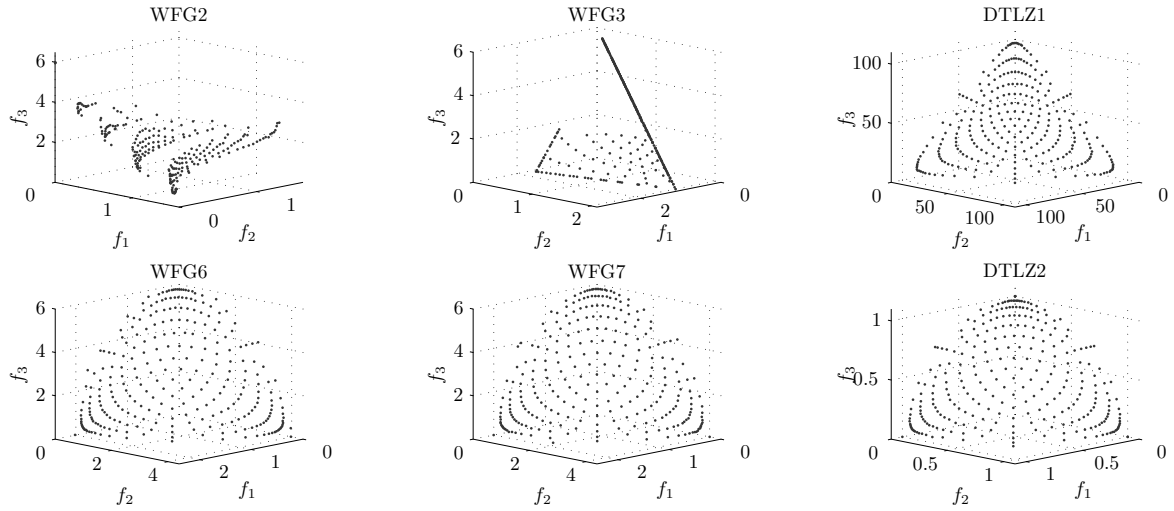

Fig. 12. Pareto front solutions found by MOEA/D for the 3-objective problem set.

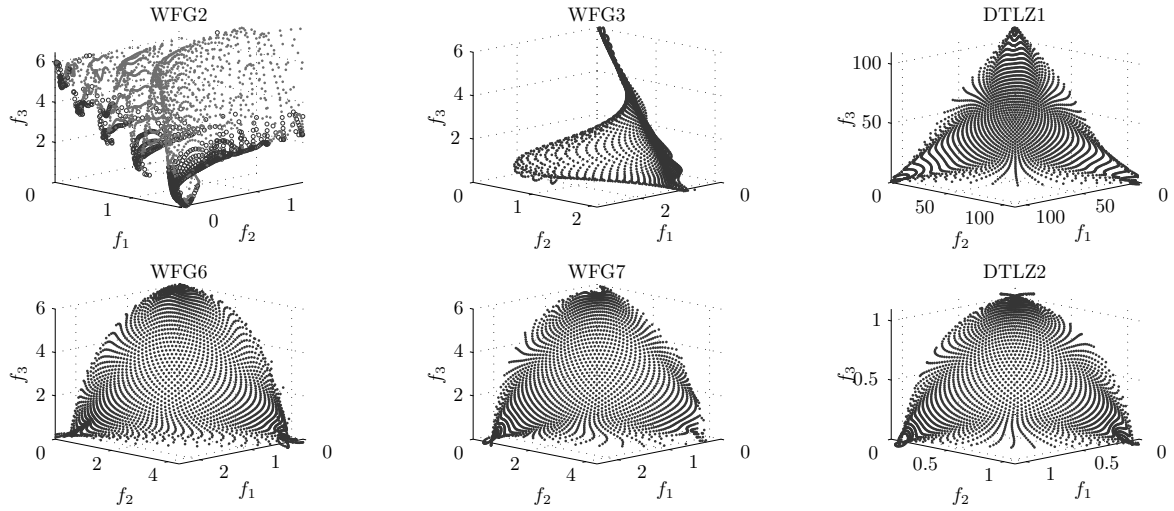

Fig. 13. Estimated solutions $\mathcal{P}_E$ ($|\mathcal{P}_E| = 3\,003$) from the non-dominated solutions found by MOEA/D for the 3-objective problem set. The non-dominated solutions in the WFG2 test problem are the represented by the darker points on the plot.

means there is no risk in the investment, as far as VaR is concerned, so $M(\mathbf{x})$ is assigned to 0. A negative value could be assigned, however this has the potential to reduce the portfolio diversification which is generally undesirable [59]. For example, if a security has never had extreme variations in its price, then it would appear that it is *safe*, so if $M(\mathbf{x})$ is allowed to be negative (i.e. guaranteed positive returns), then a clear strategy would be to allocate a big proportion of the capital to this security. However, this will reduce the portfolio diversification and increase its sensitivity to the aforementioned security. So, should this security exhibit a large negative *swing*, the entire portfolio would follow. An even more conservative approach would be to assign a lower bound on VaR for securities whose historic price has never exhibited extreme variations. Since VaR can fail to account for risk due to lack of portfolio diversification [59] and the variance of the portfolio is insensitive to extreme events, the two objectives $V(\mathbf{x})$ and $M(\mathbf{x})$ complement each other well.

B. Decision Making Procedure

Given a Pareto set approximation, $\mathcal{P}$, and using the Pareto estimation method, the decision maker has the opportunity to request a solution that is not present in the original Pareto set approximation. To illustrate this consider the following scenario. Let us assume that the decision maker is interested in a solution, $\tilde{\mathbf{z}} \notin \mathcal{P}$, that is within the convex hull of the following solutions, $\mathbf{z}_1, \mathbf{z}_2, \mathbf{z}_3 \in \mathcal{P}$. Without the PE method, a solution to this would be to re-start the optimization process, use another optimization algorithm or involve the decision maker in the optimization procedure using some preference articulation method, for instance [1]. All these alternatives have a high cost in function evaluations and are not guaranteed to produce the desired results. However, while the PE method cannot guarantee positive results either, it does enable the analyst to try and satisfy the DMs' request at a much lower cost. A way to leverage the Pareto estimation method could be the following:

- Request from the decision maker to specify the regions

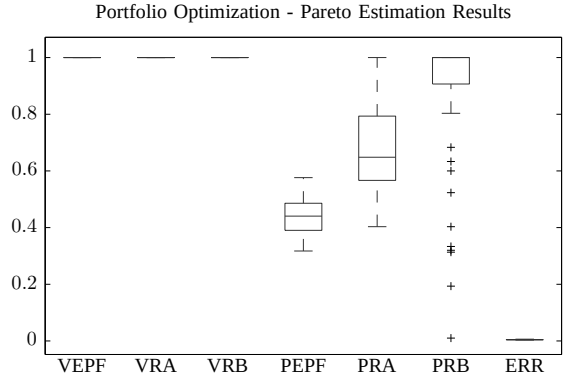

Fig. 14. **VEPF:** Number of valid solutions generated by the PE method when considering the entire Pareto front. **VRA:** Number of valid solutions generated by the PE method for region A. **VRB:** Number of valid solutions generated by the PE method for region B. **PEPF:** Number of Pareto optimal solutions generated by the PE method when considering the entire Pareto front. **PRA:** Number of Pareto optimal solutions generated by the PE method for region A. **PRB:** Number of Pareto optimal solutions generated by the PE method for region B. **ERR:** Neural network generalization error calculated using cross validation.

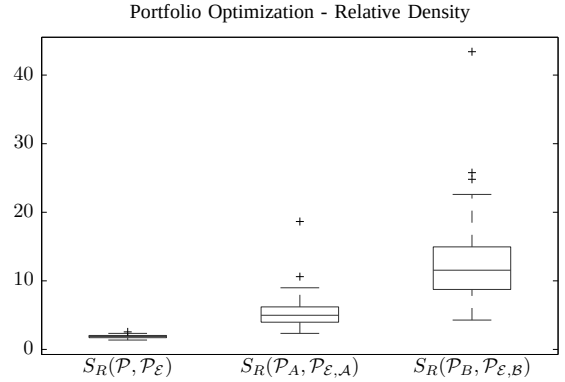

Fig. 15. Mean distance to nearest neighbour ratio of: (i) $S_R(\mathcal{P}, \mathcal{P}_\mathcal{E})$ entire Pareto front approximation produced by NSGA-II, $\mathcal{P}$, divided by the set obtained by the PE method, $\mathcal{P}_\mathcal{E}$, for the entire PF, (ii) $S_R(\mathcal{P}_A, \mathcal{P}_{\mathcal{E},\mathcal{A}})$ the Pareto optimal solutions in the neighbourhood of region A, $\mathcal{P}_A$, divided by the set of solutions obtained by the PE method in region A, $\mathcal{P}_{\mathcal{P},\mathcal{A}}$, (iii) $S_R(\mathcal{P}_B, \mathcal{P}_{\mathcal{E},\mathcal{B}})$ the Pareto optimal solutions in the neighbourhood of region B, $\mathcal{P}_B$, divided by the set of solutions obtained by the PE method in region B, $\mathcal{P}_{\mathcal{P},\mathcal{B}}$.

of interest by selecting points from the obtained Pareto set.

- For each region select 3 points $\mathbf{z}_1, \mathbf{z}_2, \mathbf{z}_3 \in \mathcal{P}$ that fully enclose the preferred region on the Pareto front. For 2-objective problems, 2 points would suffice.
- Project the points on to CH_I. Let these points be $\mathbf{w}_1, \mathbf{w}_2, \mathbf{w}_3$.
- Generate points within the $\mathbf{conv}\{\mathbf{w}_1, \mathbf{w}_2, \mathbf{w}_3\}$, namely the convex hull of the set of points $\{\mathbf{w}_1, \mathbf{w}_2, \mathbf{w}_3\}$. A way to achieve this is to create a set of evenly spaced weighting vectors, as described in [50]. Let W be an $N \times k$ matrix of N evenly spaced weighting vectors and $k = 3$ in this example, then:

$$\tilde{W} = W \cdot \begin{pmatrix} \mathbf{z}_1 \\ \mathbf{z}_2 \\ \mathbf{z}_3 \end{pmatrix}, \tag{28}$$

  where the resulting matrix, $\tilde{W}$, will be comprised of points within the $\mathbf{conv}\{\mathbf{w}_1, \mathbf{w}_2, \mathbf{w}_3\}$ by definition [60].
- Use the general version of the Pareto estimation method (see Section IV-C) to identify the mapping $\tilde{F}_\mathcal{P}$.
- Use the points in $\tilde{W}$ as input to the identified mapping, $\tilde{F}_\mathcal{P}$, to obtain a set of decision vectors, $\mathcal{D}_\mathcal{E}$, that will generate Pareto optimal points in the convex hull of the region enclosed by $\mathbf{z}_1, \mathbf{z}_2, \mathbf{z}_3$.
- Finally, using the objective function verify that the set, $\mathcal{D}_\mathcal{E}$, does indeed produce Pareto optimal solutions.

Following the above mentioned procedure any region of interest on the Pareto front can be further explored without incurring the usually high cost of restarting the optimization algorithm.

C. Portfolio Optimization Experiments

To evaluate the Pareto estimation method on the portfolio optimization problem defined in (VI-A), NSGA-II was used with $N = 300$ for 350 generations, totaling a $105\,000$ function evaluations. This procedure was performed for 50 independent runs. Furthermore the dimension of the decision vector was set to $n = 20$, which was comprised of 20 randomly selected securities. The historic data used for the calculation of the objective function are daily opening prices for the past $3\,000$ trading days and where obtained from Yahoo! Finance [61]. Subsequently the PE method was used to obtain more Pareto optimal solutions for the entire Pareto front using the method described in Section IV-C and two pre-specified regions using the procedure described in Section VI-B. The number of requested solutions for the entire Pareto front where $3\,003$ and for regions A and B 300 additional points where generated within the aforementioned regions. These results are shown in *Fig. 16*.

In *Fig. 14* the statistics of the output of the PE method are shown. Notice that for all regions, namely the entire Pareto front and the regions A and B, all generated solutions are valid. Furthermore the ratio of Pareto optimal solutions to dominated solutions for the case of the entire Pareto front seems to be lower when compared to regions A and B. However its' median is approximately 0.41, which translates to 4 Pareto optimal solutions for every 10 generated solutions. This seems to be a fairly good ratio, since for only $3\,003$ function evaluations an additional 1231 Pareto optimal solutions are generated. Also, notice that for regions A and B (see *Fig. 16*) this ratio is significantly higher. This is potentially due to the size of the requested region and the quality of the model in these parts of the PF. However the important benefit of the PE method is seen from the accuracy in location of the generated solutions in the above mentioned regions. So, for a cost of 300 extra objective function evaluations the decision maker has obtained more than 160 additional Pareto optimal solutions in regions A and B, which greatly increase the chance that a specific solution would satisfy his or her preferences assuming

that the regions where selected according to Section VI-B. Furthermore, the mean nearest neighbour distance in the entire Pareto front as well as for the regions A and B is shown in *Fig. 15*, and, although the increase in density of Pareto optimal solutions for the entire Pareto front is modest (1.8 to 2.7 times larger density), the increase in density in regions A and B is phenomenal. In real terms, and given the fact that the solutions are very well distributed within the above regions (see *Fig. 16*), this increase in the density of Pareto optimal solutions means that for any desired solutions within these regions the DM will be able to find one that is 4 to 6 and 9 to 15 times[12] closer to the exact location of the preferred Pareto optimal solution within region A and B respectively.

VII. Discussion

This study has shown that the question posed in Section III-C is far from impossible to answer. In fact it can be answered with relative precision, as is strongly indicated by the results for the selected test problems, shown in *Fig. 4*, *Fig. 5* and even more so for the portfolio optimization problem, whose results are shown in *Fig. 14* and *Fig. 15*. However, the Pareto estimation method is not without its problems. For instance, since the quality of the produced solutions depends on the employed modeling method, which in turn depends on the quality of the produced Pareto set approximation, it is to be expected that when both these factors are satisfied to a higher degree; better results are to follow. This is related to the observation in [40], about the connectedness of the Pareto optimal set in decision space for continuous multi-objective problems. Namely, if the Pareto set approximation is not *close* to the true Pareto set, this argument need not necessarily hold. For instance such a Pareto set approximation need not necessarily be piecewise continuous, in decision space, as the Karush-Kuhn-Tucker conditions would not obtain for the aforementioned PS.

As mentioned in Section III-A, there are many alternative methods for identifying the mappings used in the Pareto estimation method, however since the cost of more elaborate methods renders them prohibitive for repetitive testing as the one performed in Section V, it is difficult to quantify the benefits in using more sophisticated identification methods and even more difficult to discern if the results are due to the affinity of the modeling method to the particular problem set. However, when applying the Pareto estimation method to a specific real-world problem, the analyst has several options on how to proceed to identify the required mappings used in PE. An excellent work that addresses modeling issues and proposes a comprehensive approach based on neural networks is [49], wherein the entire procedure is systematized for producing high quality models. Although it should be noted that, based on the results in this work, the radial basis function neural network proposed in Section IV-B, has more than acceptable performance given the small amount of data that is usually available in a Pareto set approximation, therefore it is an excellent starting point.

[12]These numbers refer to the 25th to 75th percentile in *Fig. 15*.

Another aspect that has become evident, especially when comparing the results produced using the Pareto sets produced by NSGA-II and MOEA/D is that the distribution of the Pareto optimal solutions on the Pareto front, disregarding their convergence, seems to be an important factor determining the quality of the model. So, it would appear if some active learning method as in [62] could be used, the results could potentially be improved. However, the problem of direct control of the distribution of Pareto optimal points in the PS is a very difficult one.

Lastly, the modeling employed in the Pareto estimation method operates under the assumption that the mapping from objective to decision space is a bijection, which seems to be limiting if in fact the objective function, $\mathbf{F}$, is many-to-one. However, careful consideration of this issue shows that this is not limiting to the Pareto estimation method, to the contrary, it can be rather helpful. This is based on the fact that, a many-to-one objective function when *viewed* from the objective space to the decision space, for every objective vector there are one or more decision vectors to be found. This means that the probability of finding one decision vector for a specific objective vector is increased, which is to the benefit of the modeling method as there are many alternatives. Also, given the way multi-objective evolutionary algorithms operate, that is they distribute Pareto optimal solutions across the entire Pareto front, this one-to-many relationship would be impossible to discern as MOEAs do not preserve solutions that result in identical objective vectors. So it would be highly unlikely for a Pareto set approximation to have such *alternatives* as this in clash with MOEA objectives.

VIII. Conclusion

Multiobjective optimization problem solvers seek to generate a satisfactory set of Pareto-optimal solutions to enable a decision-maker to select suitable solution. Here, a novel methodology that increases the density of available Pareto optimal solutions has been described. Using this method, the number of available solutions along the trade-off surface is increased, thereby greatly enhancing the ability of the DM to identify a suitable solution with accuracy.

This is accomplished by identifying the mapping of a transformed set, derived from an approximation of the Pareto optimal set, to the corresponding decision vectors. This mapping was identified with the aid of a radial basis function neural network which was subsequently used to infer a number of Pareto optimal solutions $\mathcal{P}_{\mathcal{E}}$. The proposed method was presented in two forms. The first is a general formulation that is widely applicable to any multi-objective optimization algorithm. This formulation was applied to a Pareto-based algorithm, NSGA-II, with a ten-fold increase in Pareto optimal solutions. The second form of the proposed method applies to decomposition-based algorithms. This form is motivated by the fact that by using the weighting vectors $\mathbf{w}$ in place of $\tilde{\mathcal{P}}$ we avoid the operations required to generate that set. Both versions of the proposed method were experimentally tested against a set of well-known test problems and the results strongly indicate that the suggested methodology shows great promise.

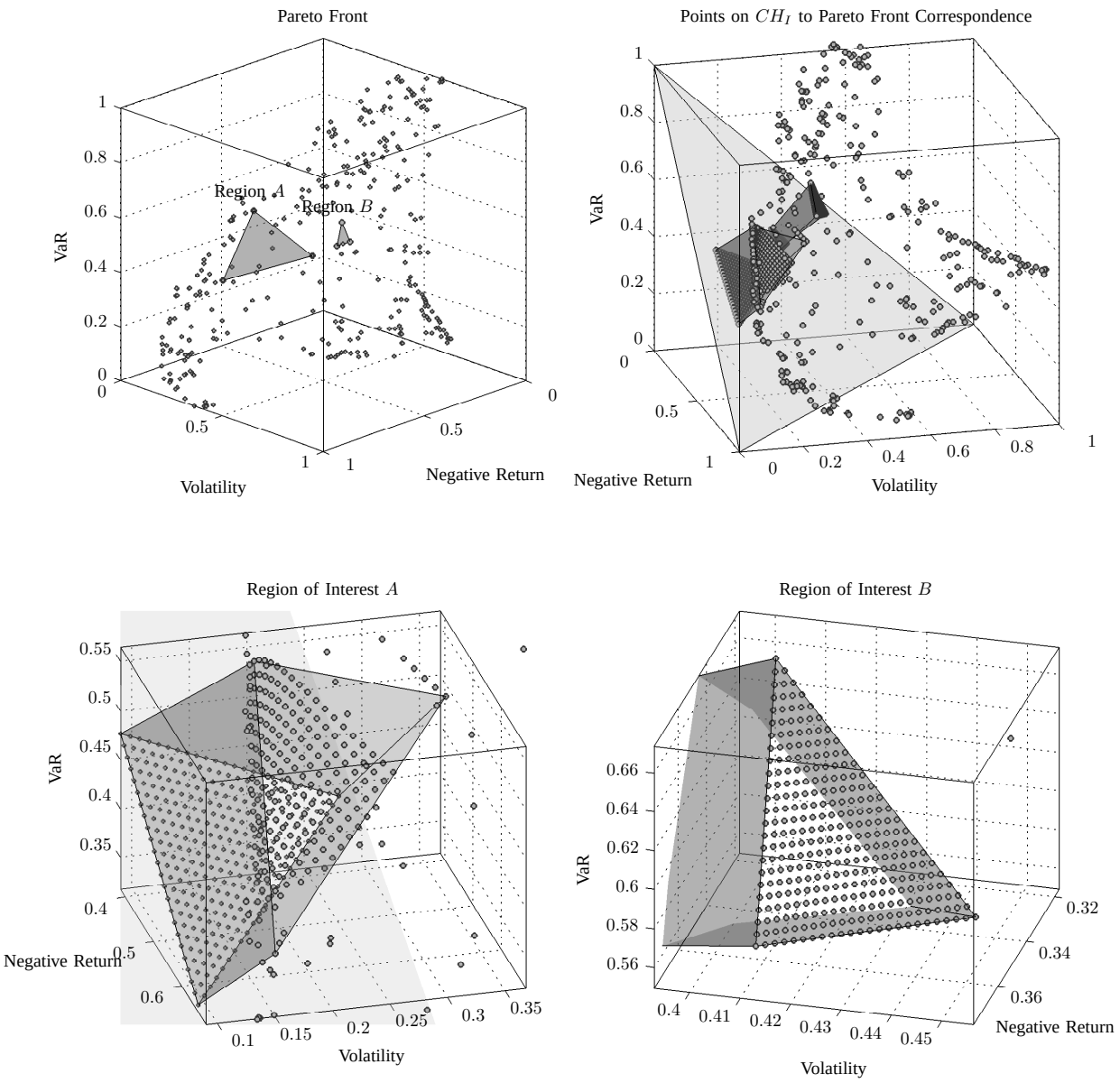

Fig. 16. **Top Left:** Portfolio optimization Pareto front and the two regions of interest. **Top Right:** The Pareto estimation method applied to identify more solutions in region A and B, the correspondence of points on the CH_I to the generated Pareto optimal solutions is marked by the shaded regions. **Bottom Left:** A closer view of the generated Pareto optimal solutions for region A. **Bottom Right:** A closer view of the generated Pareto optimal solutions for region B. Note that for illustration purposes, in the bottom and top right figures the Pareto front has been shifted by 0.1 in all dimensions.

Furthermore, the results in Section V-A and Section V-B, suggest that the choice of weighting vectors in MOEA/D is not optimal, i.e. an even distribution of Pareto optimal points is not produced by the algorithm. By even distribution we mean that the Pareto optimal points have a distribution that minimizes the s-energy which has been shown to solve the best packing problem for a sphere, see [63], [64] for further details. This effect is transferred to the results of the proposed method that used an approximation of the PF produced by MOEA/D, see *Fig. 12* and *Fig. 13*. This issue has been successfully addressed and will be reported in a future work. In contrast with MOEA/D, the Pareto-based method produced much more uniform results, see *Fig. 9*. However, there are obvious *edge* effects, which are explained by the fact that we generate solutions only within the CH_I, see Section IV-A and *Fig. 3*. This can be averted if Pareto estimation is used for specific regions, as is seen in Section VI-B.

Finally, although the concepts presented in this work need to be further developed, we believe that they can alter the definition of what we currently consider to be a well dis-

tributed approximation of the PF. This is primarily due the fact that, if an inverse mapping can be identified, then the main issue becomes that of the *optimal* allocation of Pareto optimal solutions on the PF such that the process of identifying a suitable solution is facilitated. By optimal allocation we mean an approximation set of the PF that provides the most information about the underlying PF. This, still unknown distribution, need not necessarily be an *even* distribution of Pareto optimal solutions. This issue is deferred to future research along with the exploration of the applicability of the presented method for many-objective optimization problems.

ACKNOWLEDGMENT

The authors would like to thank Ricardo H.C. Takahashi for the most interesting discussions and his invaluable perspective with respect to the present work, during his visit to the University of Sheffield, while supported by a Marie Curie International Research Staff Exchange Scheme Fellowship within the 7th European Community Framework Programme.

REFERENCES

[1] C. Fonseca, P. Fleming *et al.*, "Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization," in *Conference on Genetic Algorithms*, vol. 423, 1993, pp. 416–423.

[2] N. Srinivas and K. Deb, "Muiltiobjective Optimization Using Nondominated Sorting in Genetic Algorithms," *Evolutionary Computation*, vol. 2, no. 3, pp. 221–248, 1994.

[3] E. Zitzler, M. Laumanns, L. Thiele *et al.*, "SPEA2: Improving the Strength Pareto Evolutionary Algorithm," in *EUROGEN*, vol. 3242, no. 103, 2001, pp. 1–21.

[4] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

[5] K. Miettinen, *Nonlinear Multiobjective Optimization*. Springer, 1999, vol. 12.

[6] H. Ishibuchi, N. Tsukamoto, and Y. Nojima, "Evolutionary Many-Objective Optimization: A Short Review," in *IEEE Congress on Evolutionary Computation*, june 2008, pp. 2419 –2426.

[7] R. Purshouse and P. Fleming, "On the Evolutionary Optimization of Many Conflicting Objectives," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 770 –784, dec. 2007.

[8] Q. Zhang, A. Zhou, and Y. Jin, "RM-MEDA: A Regularity Model-Based Multiobjective Estimation of Distribution Algorithm," *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 1, pp. 41–63, 2008.

[9] C. Vira and Y. Haimes, *Multiobjective Decision Making: Theory and Methodology*. North-Holland, 1983, no. 8.

[10] F. Edgeworth, *Mathematical Psychics: An Essay on the Application of Mathematics to the Moral Sciences*. CK Paul, 1881, no. 10.

[11] V. Pareto, "Cours D'Économie Politique," 1896.

[12] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[13] E. Zitzler, L. Thiele, M. Laumanns, C. Fonseca, and V. da Fonseca, "Performance Assessment of Multiobjective Optimizers: An Analysis and Review," *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 2, pp. 117–132, 2003.

[14] C. Fonseca and P. Fleming, "Multiobjective optimization and multiple constraint handling with evolutionary algorithms. i. a unified formulation," *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, vol. 28, no. 1, pp. 26–37, 1998.

[15] ——, "Multiobjective optimization and multiple constraint handling with evolutionary algorithms. ii. application example," *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, vol. 28, no. 1, pp. 38–47, 1998.

[16] P. Fleming and R. Purshouse, "Evolutionary algorithms in control systems engineering: a survey," *Control Engineering Practice*, vol. 10, no. 11, pp. 1223–1241, 2002.

[17] J. Shoaf and J. Foster, "The Efficient Set GA for Stock Portfolios," in *IEEE International Conference on Evolutionary Computation*, may 1998, pp. 354 –359.

[18] R. Armananzas and J. Lozano, "A Multiobjective Approach to the Portfolio Optimization Problem," in *IEEE Congress on Evolutionary Computation*, vol. 2, sept. 2005, pp. 1388 – 1395 Vol. 2.

[19] R. Subbu, P. Bonissone, N. Eklund, S. Bollapragada, and K. Chalermkraivuth, "Multiobjective Financial Portfolio Design: A Hybrid Evolutionary Approach," in *IEEE Congress on Evolutionary Computation*, vol. 2, sept. 2005, pp. 1722 – 1729 Vol. 2.

[20] M. Tapia and C. Coello, "Applications of multi-objective evolutionary algorithms in economics and finance: A survey," in *IEEE Congress on Evolutionary Computation, CEC*, vol. 2007, 2007, pp. 532–539.

[21] Y.-S. Ong, P. Nair, and K. Lum, "Max-Min Surrogate-Assisted Evolutionary Algorithm for Robust Design," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 4, pp. 392 – 404, aug 2006.

[22] T. Goel, R. Vaidyanathan, R. Haftka, W. Shyy, N. Queipo, and K. Tucker, "Response Surface Approximation of Pareto Optimal Front in Multi-Objective Optimization," *Computer Methods in Applied Mechanics and Engineering*, vol. 196, no. 4, pp. 879–893, 2007.

[23] D. R. Jones, M. Schonlau, and W. J. Welch, "Efficient Global Optimization of Expensive Black-Box Functions," *Journal of Global Optimization*, vol. 13, pp. 455–492, 1998. [Online]. Available: http://dx.doi.org/10.1023/A:1008306431147

[24] K. Liang, X. Yao, and C. Newton, "Evolutionary Search of Approximated N-Dimensional Landscapes," *International Journal of Knowledge Based Intelligent Engineering Systems*, vol. 4, no. 3, pp. 172–183, 2000.

[25] Y. Jin, M. Olhofer, and B. Sendhoff, "A Framework for Evolutionary Optimization with Approximate Fitness Functions," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 5, pp. 481 – 494, oct 2002.

[26] J. Knowles, "Parego: A hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 1, pp. 50–66, 2006.

[27] Y. Jin and J. Branke, "Evolutionary Optimization in Uncertain Environments - A Survey," *IEEE Transactions on Evolutionary Computation*, vol. 9, no. 3, pp. 303–317, 2005.

[28] M. Farina, "A Neural Network Based Generalized Response Surface Multiobjective Evolutionary Algorithm," in *IEEE Congress on Evolutionary Computation*, vol. 1. IEEE, 2002, pp. 956–961.

[29] M. Emmerich, A. Giotis, M. Özdemir, T. Bäck, and K. Giannakoglou, "Metamodel - Assisted Evolution Strategies," in *Parallel Problem Solving from Nature - PPSN VII*, ser. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2002, vol. 2439, pp. 361–370.

[30] Y. Ong, P. Nair, and A. Keane, "Evolutionary Optimization of Computationally Expensive Problems via Surrogate Modeling," *AIAA journal*, vol. 41, no. 4, pp. 687–696, 2003.

[31] G. E. P. Box and K. B. Wilson, "On the Experimental Attainment of Optimum Conditions," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 13, no. 1, pp. pp. 1–45, 1951. [Online]. Available: http://www.jstor.org/stable/2983966

[32] C. Poloni, A. Giurgevich, L. Onesti, and V. Pediroda, "Hybridization of a multi-objective genetic algorithm, a neural network and a classical optimizer for a complex design problem in fluid dynamics," *Computer Methods in Applied Mechanics and Engineering*, vol. 186, no. 24, pp. 403 – 420, 2000.

[33] S. Adra, T. Dodd, I. Griffin, and P. Fleming, "Convergence Acceleration Operator for Multiobjective Optimization," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 4, pp. 825–847, 2009.

[34] Y. Jin, "A Comprehensive Survey of Fitness Approximation in Evolutionary Computation," *Soft Computing - A Fusion of Foundations, Methodologies and Applications*, vol. 9, pp. 3–12, 2005.

[35] Y. Ong, P. Nair, A. Keane, and K. Wong, "Surrogate-Assisted Evolutionary Optimization Frameworks for High-Fidelity Engineering Design Problems," *Knowledge Incorporation in Evolutionary Computation*, pp. 307–332, 2004.

[36] K. Deb and A. Srinivasan, "Innovization: Innovating Design Principles Through Optimization," in *Conference on Genetic and Evolutionary Computation*. ACM, 2006, pp. 1629–1636.

[37] S. Bandaru and K. Deb, "Automated Innovization for Simultaneous Discovery of Multiple Rules in Bi-Objective Problems," in *Evolutionary Multi-Criterion Optimization*. Springer, 2011, pp. 1–15.

[38] ——, "Automating Discovery of Innovative Design Principles Through Optimization," *KanGAL Report*, no. 2010001, 2010.

[39] O. Schütze, S. Mostaghim, M. Dellnitz, and J. Teich, "Covering pareto sets by multilevel evolutionary subdivision techniques," in *Evolutionary Multi-Criterion Optimization*, ser. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2003, vol. 2632, pp. 10–10.

[40] Y. Jin and B. Sendhoff, "Connectedness, Regularity and the Success of Local Search in Evolutionary Multi-Objective Optimization," in

Congress on Evolutionary Computation, vol. 3, dec. 2003, pp. 910 – 1917.
[41] R. Jin, W. Chen, and T. Simpson, "Comparative studies of metamodelling techniques under multiple modelling criteria," *Structural and Multidisciplinary Optimization*, vol. 23, no. 1, pp. 1–13, 2001.
[42] I. Das, "Normal-Boundary Intersection: An Alternate Method for Generating Pareto Optimal Points in Multicriteria Optimization Problems," DTIC Document, Tech. Rep., 1996.
[43] H. Maier and G. Dandy, "Neural networks for the prediction and forecasting of water resources variables: a review of modelling issues and applications," *Environmental Modelling and software*, vol. 15, no. 1, pp. 101–124, 2000.
[44] A. Atiya, "Bankruptcy prediction for credit risk using neural networks: A survey and new results," *Neural Networks, IEEE Transactions on*, vol. 12, no. 4, pp. 929–935, 2001.
[45] B. Wong and Y. Selvi, "Neural network applications in finance: A review and analysis of literature (1990-1996)," *Information & Management*, vol. 34, no. 3, pp. 129–139, 1998.
[46] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural networks*, vol. 2, no. 5, pp. 359–366, 1989.
[47] C. Bishop, "Neural Networks for Pattern Recognition," 1995.
[48] D. Rumelhart, G. Hintont, and R. Williams, "Learning Representations by Back-Propagating Errors," *Nature*, vol. 323, no. 6088, pp. 533–536, 1986.
[49] S. A. Billings, H.-L. Wei, and M. A. Balikhin, "Generalized Multiscale Radial Basis Function Networks," *Neural Networks*, vol. 20, no. 10, pp. 1081 – 1094, 2007.
[50] Q. Zhang and H. Li, "MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 712–731, 2007.
[51] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable multi-objective optimization test problems," in *Proceedings of the 2002 Congress on Evolutionary Computation, 2002. CEC '02.*, vol. 1, may 2002, pp. 825 –830.
[52] S. Huband, P. Hingston, L. Barone, and L. While, "A Review of Multiobjective Test Problems and A Scalable Test Problem Toolkit," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 5, pp. 477–506, 2006.
[53] D. Hadka and P. Reed, "Diagnostic assessment of search controls and failure modes in many–objective evolutionary optimization," *Evolutionary Computation*, 2011.
[54] R. Purshouse, C. Jalbă, and P. Fleming, "Preference-Driven Co-Evolutionary Algorithms Show Promise for Many-Objective Optimisation," in *Evolutionary Multi-Criterion Optimization*. Springer, 2011, pp. 136–150.
[55] D. Van Veldhuizen, "Multiobjective Evolutionary Algorithms: Classifications, Analyses, and New Innovations," in *Evolutionary Computation*, 1999.
[56] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: A Comparative Case Study and the Strength Pareto Approach," *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 4, pp. 257–271, 1999.
[57] H. Markowitz, "Portfolio Selection," *The Journal of Finance*, vol. 7, no. 1, pp. 77–91, mar. 1952. [Online]. Available: http://www.jstor.org/stable/2975974
[58] J. Danielsson and C. G. D. Vries, "Value-at-Risk and Extreme Returns," *Annals of Economics and Statistics*, no. 60, pp. 239–270, dec. 2000.
[59] K. Kuester, S. Mittnik, and M. S. Paolella, "Value-at-Risk Prediction: A Comparison of Alternative Strategies," *Journal of Financial Econometrics*, vol. 4, no. 1, pp. 53–89, 2006.
[60] R. Rockafellar, *Convex Analysis*. Princeton University Press, 1970, vol. 28.
[61] "Yahoo! Finance," http://uk.finance.yahoo.com/, aug. 2012.
[62] D. Cohn, L. Atlas, and R. Ladner, "Improving Generalization with Active Learning," *Machine Learning*, vol. 15, pp. 201–221, 1994.
[63] A. Katanforoush and M. Shahshahani, "Distributing points on the sphere, i," *Experimental Mathematics*, vol. 12, no. 2, pp. 199–210, 2003.
[64] D. Hardin and E. Saff, "Discretizing Manifolds via Minimum Energy Points," *Notices of the AMS*, vol. 51, no. 10, pp. 1186–1194, 2004.

Ioannis Giagkiozis received the B.Sc. degree from TEI of Thessaloniki, Thessaloniki, Greece in 2009. He then obtained the M.Sc. degree in Control and Systems Engineering with Distinction from the University of Sheffield, Sheffield, U.K. in 2010, for which he was awarded the Nicholson Prize for most outstanding student. He joined the Department of Automatic Control and Systems Engineering, University of Sheffield, Sheffield, as a Research Associate in 2011 and is currently working towards a Ph.D. degree in multiobjective evolutionary algorithms.

His research interests are in many-objective optimization, estimation of distribution algorithms and applied convex optimization.

Peter Fleming received the B.Sc. and Ph.D. degrees from The Queen's University, Belfast, U.K. He joined the University of Sheffield as Professor of Industrial Systems and Control in 1991, having previously been with Syracuse University, NY, NASA, Langley, VA and the University of Wales, Bangor, U.K. Since 1993, he has been Director with Rolls-Royce University Technology Centre in Control and System Engineering, University of Sheffield, Sheffield, U.K. He was the head of Automatic Control and Systems Engineering from 1993 to 1999, Director of Research (Engineering) from 2001 to 2003, and Pro Vice-Chancellor for External Relations from 2003 to 2008. His control and systems engineering research interests include multicriteria decision making, optimization, grid computing, and industrial applications of modeling, monitoring, and control. He has over 400 research publications, including six books, and his research interests have led to the development of close links with a variety of industries in sectors such as aerospace, power generation, food processing, pharmaceuticals, and manufacturing.

Prof. Fleming is a Fellow of the Royal Academy of Engineering, both a Fellow of, and Adviser to, the International Federation of Automatic Control, a Fellow of the Institution of Electrical Engineers, a Fellow of the Institute of Measurement and Control, an Advisor to the International Federation of Automatic Control, and the Editor-in-Chief of the International Journal of Systems Science.