Generalized Decomposition and Cross Entropy Methods for Many-Objective Optimization

Ioannis Giagkiozis, Robin C. Purshouse, and Peter J. Fleming,
Department of Automatic Control and Systems Engineering
The University of Sheffield
Research Report No. 1029
November 2012

***Abstract*—Decomposition-based algorithms for multi-objective optimization problems have increased in popularity in the past decade. Although their convergence to the Pareto optimal front (PF) is in several instances superior to that of Pareto-based algorithms, the problem of selecting a way to distribute or guide these solutions in a high-dimensional space has not been explored. In this work, we introduce a novel concept which we call *generalized* decomposition. Generalized decomposition provides a framework with which the decision maker (DM) can guide the underlying evolutionary algorithm toward specific regions of interest or the entire Pareto front with the desired distribution of Pareto optimal solutions. Additionally, it is shown that generalized decomposition simplifies many-objective problems by unifying the three performance objectives of multi-objective evolutionary algorithms – convergence to the PF, evenly distributed Pareto optimal solutions and coverage of the entire front – to only one, that of convergence. A framework, established on generalized decomposition, and an estimation of distribution algorithm (EDA) based on low-order statistics, namely the cross-entropy method (CE), is created to illustrate the benefits of the proposed concept for many objective problems. This choice of EDA also enables the test of the hypothesis that low-order statistics based EDAs can have comparable performance to more elaborate EDAs.**

***Index Terms*—Generalized decomposition, cross entropy method, MACE, many-objective optimization, multiobjective optimization, decomposition methods, scalarising functions.**

I. Introduction

MULTI-objective problems arise naturally in many disciplines, for example in control systems [1], finance [2] and biology [3]. A multi-objective problem (MOP) is defined as,

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

where k is the number of objective functions and $\mathbf{x}$ is the vector of decision variables defined in the domain $\mathbf{S} \subseteq \mathbb{R}^n$. It should be clarified what we mean by the notation $\min_{\mathbf{x}}$ is minimization over $\mathbf{x}$ which is different to the $\min$ operator which returns the minimum element of a set. We follow this convention because it leads to a more compact description. There is an implicit assumption that the scalar objective functions are competing, since if this assumption is not true then (1) degenerates to a single objective problem, or if some of the k objectives are competing and some are harmonious then the *effective* number of objectives will be less than k [4]. MOPs for 2 or 3 objectives have been heavily studied, however there is the need for algorithm frameworks that can deal with higher dimensional problems, i.e. more than 3 objectives. These problems are so-called many-objective problems (MAPs), for brevity we refer to multi and many-objective problems simply as MAPs.

The problem that is apparent in MAPs is that there is no *natural* way of *ordering* the obtained solutions; this ordering is crucial for fitness assignment. However MOEAs base their *decision* as to the direction of search on the assigned fitness of various solutions in the population. This is a very well known problem in MAPs and has been addressed with varying degrees of success by a number of researchers over the past three decades [5]–[7]. In general there are two approaches employed to resolve this issue: Pareto-based and decomposition-based methods. In both methodologies there is the assumption that the relative importance of the objectives is unknown. In the case that this information is given by the decision maker (DM) then a decomposition method can be used to create a scalar objective function, see Section III.

Pareto-based methods use the Pareto-dominance relations [8], to induce partial ordering in the objective space. These relations, were initially introduced by Edgeworth [9] and later expanded by Pareto [10]. For example for two vectors $\mathbf{a}, \mathbf{b} \in \mathbb{R}^n$, $\mathbf{a} \preceq \mathbf{b}$ if all the elements in $\mathbf{a}$ are smaller or equal ($\leq$) to the corresponding elements in $\mathbf{b}$ and at least one element in $\mathbf{a}$ is strictly ($<$) smaller than its corresponding element in $\mathbf{b}$. This partial ordering, induced by the $\preceq$ relation, is denoted as $\mathbf{a} \preceq \mathbf{b}$, and, in the context of a minimization problem this expression is read as: the vector $\mathbf{a}$ dominates $\mathbf{b}$. For a more complete treatment of Pareto-dominance relations the reader is referred to [8]. However such relations are of limited utility when the number of dimensions is increased [11]. This is primarily because the number of non-dominated solutions increases as the dimensionality of the problem increases, and for dimensions greater than around ten, almost all the solutions are non-dominated [12]. Hence this type of partial ordering becomes of limited use in high dimensions since, if all the generated solutions are non-dominated, the EA has no objective measure on which to base its selection process.

Decomposition-based methods employ a scalarizing func-

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

tion to aggregate all the objectives into a single objective function. Such methods have been used predominantly in non-linear mathematical programming, where the main algorithm is based on some variant of gradient search [8], [13]. However multi-objective evolutionary algorithms (MOEAs) have also employed decomposition with varying degrees of success, for example [14]–[16]. Arguably, decomposition methods have not been explored to sufficient depth for MAPs. For example, a popular hypothesis, that is employed by several MOEAs, is that an even distribution of weighting vectors will result in *well* distributed Pareto optimal points [7]. However, with the help of a novel concept which we call *generalized* decomposition, we show that this assumption is fundamentally flawed and we provide an exact solution to this issue, subject to some prior information. It is interesting to note that recently several researchers have taken an interest in the selection of weighting vectors in decomposition-based methods. For instance [17] identify two issues with the way that set of weighting vectors are selected in MOEA/D [7]: (i) it is not possible to select an arbitrary number of weighting vectors, which can be problematic for many-objectives, and, (ii) the number of weighting vectors situated on the *boundary* tends to be large. The boundary in this context is understood to mean: weighting vectors with many components equal to zero. Weighting vectors on the boundary produce subproblems that completely disregard some of the objective functions which in general is undesirable [17]. The suggestion is to use *uniform design* to select the set of weighting vectors instead of a set of evenly distributed weighting vectors. However, as is shown in this work, an even or *uniform* distribution of weighting vectors does not produce evenly distributed Pareto optimal solutions, hence what is proposed in [17] does not address the more pressing issue, that of finding the distribution of weighting vectors that would lead to a Pareto set whose points have a *desirable* distribution on the PF. This distribution can be defined in numerous ways, and depends mostly on the preferences of the DM. This issue is further discussed in Section III-C.

An interesting adaptive method to select the set of weighting vectors is presented in [18], [19]. The main idea is to identify the Pareto front geometry and then distribute a set of points on that surface in such a way so as to maximize the hypervolume indicator [20]. Subsequently, the points found in the previous step, are used to identify a weighting vectors that, upon minimization of the resulting subproblems, would result in similar points on the Pareto front. The idea seems hopeful, however, there are three major difficulties with this approach. First, the authors assume that the Pareto front can be parameterized using the following,

$$f_1^{p_1} + f_2^{p_2} = 1, \tag{2}$$

where, $p_i \in \mathbb{R}_{++}$ and the fact that (2) equals to one means that the objective functions are normalized in the range, $[0, 1]$. The problem is that (2) is nonconvex but the authors of [18], [19] ignored this issue and used the Newton method to solve for the p_i parameters. Therefore, if there is noise in the Pareto optimal points used in identifying the p_i parameters or the Pareto front geometry has, $p_i \neq p_j$, $i \neq j$, this method will fail. This can be seen in [19] whereby a front described by: $f_1^2 + f_2 = 1$ is generated and the estimate using the Newton method is: $f_1^{1.445} + f_2^{1.445} = 1$. Therefore, the first part of the suggested method can mislead the entire procedure in [18], [19]. The second problem, is that the weighting vectors that correspond to points on the identified Pareto front are formulated in a similar fashion to (2), hence the issue of nonconvexity of the problem formulation emerges again and the resulting weighting vectors will not produce subproblems that converge to the reference points. Lastly, the hypervolume indicator [20], which is used to ascertain the quality of the *reference* points on the PF, has exponential complexity in the number of objectives [21], [22], which limits the method to approximately 4-objective problems, since the hypervolume must be calculated several times on every iteration of the algorithm [19].

Most tantalizingly, in a recent publication Gu et al. [23] discuss a solution for identifying a weighting vector set using a set of evenly distributed Pareto optimal solutions. However, the proposed method in the above mentioned work is limited for the weighted sum method and the Chebyshev scalarizing function [23]. For example if weakly Pareto optimal solutions are to be avoided, the *modified* Chebyshev scalarizing function [8, pp. 101] can be used. However there is no clear way in identifying the required set of weighting vectors using the proposed methodology in [23].

Evolutionary algorithms (EAs) have found numerous applications in MAPs [12]. This is because most EAs are population-based, in the sense that at each iteration an entire population of solutions is evaluated. This feature is quintessential to MAPs since, in a posteriori optimization, an entire family of solutions is required to describe the trade-off surface. This trade-off surface in objective space is also called the Pareto front (PF). Another important reason for EA applicability is that they impose almost no constraints on the problem structure; for example, continuity and differentiability are not required for EA operation. Due to these factors MAP research is vibrant in the EA community, something that can be attested by the number of EAs available for MAPs, e.g. [7], [12], [24]. Specifically EAs are comprised of a number of algorithm families, such as genetic algorithms (GAs) [25] and evolution strategies (ES) [26], as well as differential evolution (DE) [27] and others. Most of the aforementioned algorithm families are inspired by some naturally occurring process, such as DNA recombination and mutation [25]. However this presents certain difficulties. For example, it is very hard to analyse the behaviour of MOEAs analytically, thus their performance on a problem cannot be guaranteed prior to application. This is why EAs are usually evaluated experimentally using some test problem sets [28]–[30].

More recently, a new family of algorithms has emerged, namely estimation of distribution algorithms (EDAs). EDAs stand in the middle ground between Monte-Carlo simulation and EAs. In EDAs, a probabilistic model is built, based on elite individuals, which subsequently is sampled producing a

new population of *better*[1] individuals. From the EA point of view, EDAs can be traced back to recombination operators based on density estimators that use good performing individuals in the population as sample [31]. A positive aspect of EDAs is that it is straightforward to fuse prior information into the optimization procedure, thus reducing the time to convergence if such information is available. Also, the amount of heuristics, compared with other EAs, is reduced easing the task of mathematical analysis of these algorithms. This is an important aspect which has been overlooked, due to inherent difficulties, in most heuristics for optimization. Studies of this kind are usually applied to algorithms that are not used in practice [32], [33], therefore the practical value of such studies is limited. However EDAs are not a panacea since they heavily depend on the quality and complexity of the underlying probabilistic model [34]. For instance, a simple EDA based on low-order statistics, i.e. an EDA that does not account for variable dependencies, can be easily misled if, in fact, such dependencies exist in the underlying problem. To overcome such difficulties researchers proposed ever more elaborate models [34], which of course increase the complexity of the algorithm and in some instances the identification of the optimal model is of comparable complexity to that of the optimization problem necessitating the use of heuristics [35]. Acknowledging this problem has led some researchers to suggest hybridization of EDAs based on simple probabilistic models with some form of clustering [36]. This course is further supported by more recent studies [37].

For these reasons we have selected an optimization algorithm, the so-called Cross Entropy method (CE), as the main algorithm in our generalized decomposition-based framework. The CE-method was introduced by Rubinstein [38], initially as a rare event estimation technique and subsequently as an algorithm for combinatorial and continuous optimization problems. The most alluring feature of CE is that, for a certain family of instrumental densities, the updating rules can be calculated analytically, and thus are extremely efficient and fast. Also the theoretical background of CE is enabling theoretical studies of this method which can provide sound guidelines about the applicability of this algorithm to problems.

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

- A generalization of decomposition methods is presented, that is applicable to a wide range of EAs and for all scalarizing functions that are convex with respect to the weighting vectors. Using the presented methodology, the spread of the resulting PF can be directly controlled. Additionally, it is shown how generalized decomposition can be used to refine the search of a MOEA in regions that are of particular interest to the DM, thus introducing preference articulation for decomposition methods.
- Using generalized decomposition, the CE-method is extended to MAPs and is shown to perform very well compared to two other EAs, namely MOEA/D, RM-MEDA, and random search.

[1]Or more precisely, individuals that are more likely to be better than their predecessors.

The remainder of this paper is structured as follows. In Section II we elaborate on the ensuing problems in Pareto-based methods for many objective problems. In Section III generalized decomposition is described along with the benefits that this method can bring to currently existing MOEAs. Following this, in Section IV the CE-method is presented along with its form for continuous optimization problems. A many-objective optimization framework based on generalized decomposition and the CE-method is presented in Section V. The algorithms in our comparative studies in Section VII are described in Section VI. In Section VIII we illustrate how generalized decomposition can be used for preference articulation. Lastly in Section IX we summarize and conclude this work.

II. Pareto-Based Methods And Many-Objective Problems

The concept of Pareto-dominance is of limited use as a fitness assignment scheme for many-objective problems. Of course, Pareto-optimal solutions in any number of dimensions will still be the minimal elements of the feasible set in objective space. In [39] a very interesting geometric argument is presented that should clarify this point. In what follows we elaborate on the above mentioned argument.

Consider the simplest multi-objective case, namely a 2-objective problem. Every point in objective space defines 4 regions, (i) a region that contains solutions that are clearly better, (ii) a region that contains solutions that are clearly worse and (iii, iv) two regions where the solutions are incomparable to the point in question. Now, for 3-objective problems there are 8 such regions (2^3), however there is only 1 region which contains clearly better solutions and 1 region with clearly worse solutions. So, there are $2^3 - 2 = 6$ regions that contain solutions incomparable to the point in question. In general the following is true, for k-dimensional problems, there is always 1 region with clearly better solutions, 1 region with clearly worse solutions and $2^k - 2$ regions containing incomparable solutions. Now, assuming that there is no bias towards any of these regions in the problem (objective function), the probability that a solution is generated in any one of these regions by a stochastic process (algorithm) is approximately proportional to the volume of these regions divided by the volume of the entire feasible set in objective space[2]. However, for increasing number of dimensions, the likelihood that a solution will be generated within the region of solutions that are clearly better becomes almost insignificant the closer the point is to the Pareto front [39]. For example, for 11 dimensions there are $2\,048$ regions, hence for the above problem, the probability for a solution to be generated, that dominates the current point, is approximately $p = 1/2\,048 = 4.88^{-4}$ if the point in question is exactly in the *middle* of the feasible objective set. To contrast this, the probability that a non-comparable solution is generated is $2\,046/2\,048 \approx 0.99$.

However, we have simplified the problem greatly, that is we have assumed no bias and that the point is significantly away from the Pareto front so that the volume of all the regions

[2]We assume that the feasible objective set is bounded.

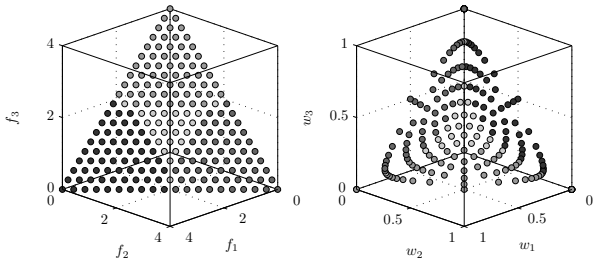

Fig. 1. Left: An affine Pareto front. Right: The corresponding optimal weighting vectors. Different shades of grey aid in identifying corresponding regions in the Pareto front and the associated weighting vectors.

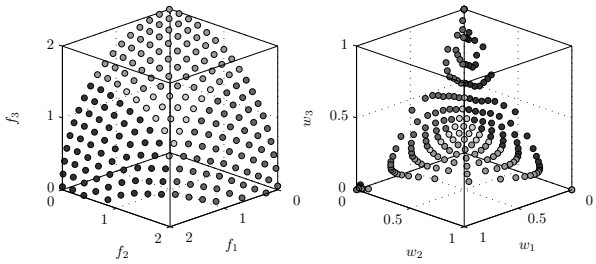

Fig. 2. Left: A concave Pareto front. Right: The corresponding optimal weighting vectors.

is approximately the same. Naturally, this is highly unlikely to be the case, so let us consider a more realistic scenario. Let the problem in question be bounded, so assuming we know the boundaries, we can shift it so that its forward image (objective space) is the nonnegative orthant. The only case that the problem will be easier to solve, is when there is bias towards the Pareto front, but this is not usually encountered in practice. The contrary is a much more common situation, namely that there is "resistance" in finding better solutions. This combined with the fact that as a solution approaches toward the Pareto front the region that contains clearly better solutions is becoming very small, the probability that a worse solution is generated is increasing toward $p \to 1$ (no-bias towards worse solutions) and the probability of generating a better solution is diminishing toward $p \to 0$. Regarding the regions that contain incomparable solutions, their volume is *exchanged* with the region that contains clearly worse solutions. Therefore it becomes increasingly more difficult to find solutions in the desirable direction.

The difference with decomposition-based algorithms is that, for each subproblem a complete ordering of the objective space is defined, irrespective of its dimension. This in effect reduces[3] the rate of decrease of the probability that a better solution is generated [39]. To see this consider a scenario that the weighted sum method is used (see (3)). In this scenario the weighting vector represents the normal of a hyperplane that separates the feasible objective space in two regions. One region containing better solutions and one with worse solutions. Solutions above the hyperplane are considered to be *worse* while solutions below the hyperplane are taken to

[3]This statement is true only for the weighted sum scalarizing function. For other scalarizing functions a more elaborate formulation is required, however there are indications that a similar statement may be established [39].

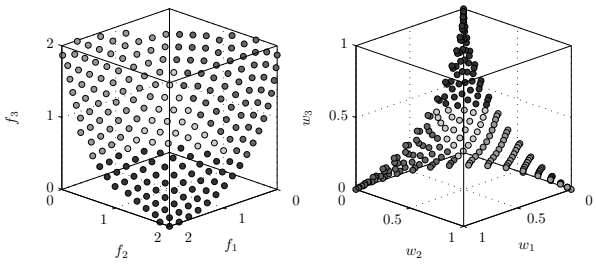

Fig. 3. Left: A convex Pareto front. Right: The corresponding optimal weighting vectors.

be better with respect to the particular subproblem [39]. An intuitive way that explains why this is the case is if we consider the effect of a scalarizing function to the objective space. A scalarizing function projects the entire objective space onto a line[4], therefore some regions that contain incomparable solutions in the Pareto sense, now become solutions that are either better or worse for the particular subproblem. Admittedly this is not an entirely desirable behaviour, however the algorithm is provided with an unambiguous direction of search. It should be noted that by using a decomposition-based method, the problem does not become any easier to solve. The major difference between decomposition-based and Pareto-based algorithms is that the former provide unambiguous information about the quality of the produced solutions at every iteration while the latter cannot always guarantee such information because the likelihood of generating incomparable solutions in high dimensions is high [12]. However it is easy to reduce the above argument into a zugzwang between Pareto-based methods and decomposition-based methods. This is accomplished by the simple observation that the *clearly better* regions in the Chebyshev scalarizing function (see (5)) are identical to the regions generated by Pareto dominance based methods, while the incomparable and clearly worse regions in Pareto-based methods are mapped to *clearly worse* regions by the Chebyshev scalarizing function. Namely, if we require a decomposition method that can guarantee the generation of Pareto optimal solutions, then, we have to use the Chebyshev scalarizing function but in so doing we give up the favourable convergence rates[5] achieved when using the weighted sum method, and vice versa. There are ways that different scalarizing functions can be used to adaptively resolve this issue while preserving the guarantees that the Chebyshev function provides however this requires further investigation.

III. Generalized Decomposition

A. Decomposition Methods

Decomposition methods, or so-called scalarizing functions, have been employed in several MOEAs, for example [14]–[16]. These methods transform (1) to a single-objective problem by combining the objective functions to form a single scalar objective function. The potential of such methods for extending MOEAs to MOPs is obvious considering the basis

[4]In this case a segment of a ray, since the objective space is bounded.

[5]Or more correctly the potential for favourable convergence rates.

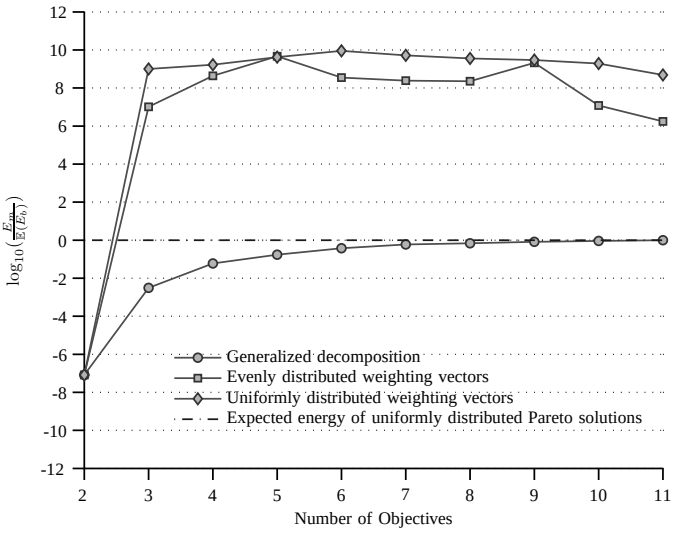

Fig. 4. Logarithm of the energy ratio of generalized decomposition, evenly distributed weighting vectors [7], and uniformly distributed weighting vectors [14].

of almost every, if not all, optimization algorithms is a method that can address only single objective problems. Therefore decomposition methods present a clear path in extending such algorithms to MOPs.

Arguably the simplest scalarizing function is the weighted sum method [40]:

$$\begin{aligned} &\min_{\mathbf{x}} \mathbf{w}^T \mathbf{F}(\mathbf{x}) \\ &\sum_{i=1}^{k} w_i = 1, \text{ and } w_i \geq 0, \forall\, i \in \{1, \ldots, k\}, \end{aligned} \tag{3}$$

where $\mathbf{w} = (w_1, \ldots, w_k)$. However it has been shown that for complicated Pareto fronts, an algorithm based on (3) is unable to discover all Pareto optimal solutions [8]. Although, with some modifications this simple decomposition can produce respectable results, for example see [7].

A more sophisticated decomposition is based on the weighted metrics method [40]:

$$\min_{\mathbf{x}} \left(\sum_{i=1}^{k} w_i |f_i(\mathbf{x}) - z_i^\star|^p \right)^{p^{-1}}, \tag{4}$$

here as in (3), it is assumed that $w_i \geq 0$ and that $\sum_{i=1}^{k} w_i = 1$, and $p \in [1, \infty)$. Also $\mathbf{z}^\star$ is the *ideal* vector, which is equal to the minimum values for all the objectives independently. When $p \to \infty$ the well known Chebyshev decomposition is obtained:

$$\min_{\mathbf{x}} \| \mathbf{w} \circ |\mathbf{F}(\mathbf{x}) - \mathbf{z}^\star| \, \|_\infty. \tag{5}$$

The $\circ$ operator denotes the Hadamard product which is element-wise multiplication of vectors or matrices of the same size. This decomposition is quite interesting due the fact that there are theoretical results stating that for any Pareto optimal solution $\tilde{\mathbf{x}}$ there exists a *convex* weighting vector $\mathbf{w}$ for which the solution of (5) is $\tilde{\mathbf{x}}$ [8]. Note that by, convex weighting vector, $\mathbf{w}$, we mean a vector $\mathbf{w} \in \mathbf{conv}\, C$, where $C = \{\mathbf{e}_i : i = 1, \ldots, k\}$ and $\mathbf{e}_i$ is a vector whose components are all equal to zero, except its i^{th} component that is equal to one. Also $\mathbf{conv}\, C$ is the *convex hull* of the set C which is defined in (37). For further details see Appendix B. This means that all Pareto optimal solutions can be found using the Chebyshev decomposition. This result is very encouraging, however it does not suggest a way to choose the weighting vectors $\mathbf{w}$ in order for a representative and evenly spread PF to be obtained.

B. Optimal Choice of Weighting Vectors

The guarantee that all Pareto optimal solutions can be obtained by the Chebyshev decomposition, for some convex weighting vector $\mathbf{w}$, is well known and has been exploited on numerous occasions in past research. For example Jaszkiewicz [14] suggests that a uniformly sampled set of weighting vectors $\mathbf{w}$ should produce uniformly distributed Pareto optimal solutions along the entire PF. Later Zhang et al. [7] argue that choosing at each iteration a new random weighting vector is too ambitious, since only an approximation of the PF is necessary. Instead the authors suggest that a set of evenly spaced weighting vectors should produce *well* distributed Pareto optimal solutions. Their main argument was that this should be the case since the various subproblems obtained using different weighting vectors are a continuous function of the weights [7]. This seems to be the case, however there is nothing to suggest that this *continuous* function is also linear in the parameters $\mathbf{w}$, which is the only case for which their assumption would hold, up to a multiplicative constant. Namely, an evenly distributed set of weighting vectors would produce well distributed Pareto optimal solutions only in the case that the function $g_\infty(\cdot)$ defined as:

$$\begin{aligned} \min_{\mathbf{x}} g_\infty(\mathbf{x}, \mathbf{w}^s, \mathbf{z}^\star) &= \| \mathbf{w}^s \circ |\mathbf{F}(\mathbf{x}) - \mathbf{z}^\star| \, \|_\infty \\ \forall\, s &= \{1, \ldots, N\}, \\ \text{subject to } \mathbf{x} &\in \mathbf{S}, \end{aligned} \tag{6}$$

is linear in the weights $\mathbf{w}$, which is obviously not the case. The parameter N in (6) is the size of the population which is equal to the number of subproblems to be solved and $\mathbf{w}^s$ is the weighting vector of the s^{th} subproblem.

Therefore, the assumption that, well distributed Pareto optimal points will result from decomposing an MAP into a set of scalar subproblems with the aid of evenly spaced weighting vectors $\mathbf{w}$, is not entirely valid. An illustration of this can be seen in *Fig. (1) – Fig. (3)*[6] where to the left we depict a PF and to the right we calculate the weighting vectors that would produce these Pareto optimal solutions, assuming that the algorithm is successful in minimizing all subproblems. This calculation was performed with what we call *generalized decomposition*, which is given by the solution of the program in (7). The insight in this formulation is that by using (7) we can *solve* the inverse problem, i.e. given a point $\mathbf{F}(\tilde{\mathbf{x}})$ in objective space we want to find a unique convex weighting vector $\tilde{\mathbf{w}}$ for which the following would be true

[6]An affine function is a linear function plus a shift, namely $\mathbf{y} = \alpha\mathbf{x} + \mathbf{c}$, is an affine function.

$\|\tilde{\mathbf{w}} \circ \mathbf{F}(\tilde{\mathbf{x}})\|_\infty \leq \|\mathbf{w} \circ \mathbf{F}(\tilde{\mathbf{x}})\|_\infty$ for all convex vectors $\mathbf{w}$. This means, that for all possible subproblems defined by the set of weighting vectors $\mathbf{w} \in \mathcal{W}$, the Pareto optimal solution $\mathbf{F}(\tilde{\mathbf{x}})$ is *closest* to the subproblem defined by the weighting vector $\tilde{\mathbf{w}}$. Closest in this context means that the Pareto optimal solution, $\mathbf{F}(\tilde{\mathbf{x}})$, minimizes the subproblem defined by $\tilde{\mathbf{w}}$. Additionally, $\mathcal{W}$ is the set of all k dimensional convex vectors. The ability to obtain the weighting vector $\tilde{\mathbf{w}}$ for a particular point on the Pareto front can be exploited in several ways as explained later in this section. To obtain the $\tilde{\mathbf{w}}$ vectors, the following program is to be solved for every Pareto optimal point of interest:

$$\begin{aligned} &\min_{\mathbf{w}} \|\mathbf{w} \circ \mathbf{F}(\mathbf{x})\|_\infty, \\ &\text{subject to } \sum_{i=1}^{k} w_i = 1, \\ &\text{and } w_i \geq 0,\ \forall\, i \in \{1, \ldots, k\}. \end{aligned} \tag{7}$$

Also to obtain the optimal weighting vectors for the weighted metrics scalarizing function for p other than infinity, all that is required is to change the norm in (7) to reflect that change. If the scalar objective functions $(f_1(\mathbf{x}), \ldots, f_k(\mathbf{x}))$, that comprise the objective vector $\mathbf{F}(\mathbf{x})$, are non-negative for all $\mathbf{x} \in \mathcal{S}$ then the problem formulated in (7) is a disciplined convex program [41], hence it is also a convex program. So a unique solution is guaranteed and can be obtained by solving (7) using some interior-point method [42]. On a side note the non-negativity constraint on the scalar objective functions can be relaxed in the case that all scalar functions are bounded from below and these lower bounds are known. In which case $\mathbf{F}(\mathbf{x})$ is replaced by,

$$\tilde{\mathbf{F}}(\mathbf{x}) = \left(f_1 - b_1, \ldots, f_k - b_k\right), \tag{8}$$

where b_i are the respective lower bounds for the scalar objective functions f_i. For details on the formalism of disciplined convex programming, the interested reader is referred to [41]–[43].

The general idea is that the generation of weighting vectors greatly influences the convergence and spread of the resulting Pareto front. However, this selection has been either arbitrary [14], or based on invalid assumptions [7]. Additionally, the method presented by MOEA/D (see Section VI-A) to generate weighting vectors is limiting in the sense that for high dimensional problems the choice of the size of the population is restrictive. For example, for $H = 10$, where H can be interpreted as the number of divisions per dimension for the weighting vectors, and for 11 objectives the population size must be equal to $92\,378$. This H setting is less than half of that used by Zhang et al. [7] for 3-objective problems. This restriction can prove problematic in certain situations, for example if a different choice of population size is more natural or if there are computational and memory constraints.

C. The Effect of Weighting Vector Choice

Assuming that our definition of *well* distributed PF solutions is a Pareto optimal set uniformly distributed along the trade-off surface, the following experiment illustrates the benefits of using generalized decomposition. It should be noted that the generalized decomposition framework is fully capable of accommodating any other definition of well distributed Pareto optimal solutions. A commonly used measure of evenly distributed points on the unit hypersphere is the Coulomb potential [44], or Riesz kernel [45], defined as:

$$\begin{aligned} E(\mathbf{Z}; s) &= \sum_{1 \leq i \leq j \leq N} \|\mathbf{z}_i - \mathbf{z}_j\|^{-s},\ s > 0 \\ \mathbf{z} &\in \mathbb{R}^k,\ \text{and, } \mathbf{Z} = \{\mathbf{z}_i : i \in \{1, \ldots, N\}\}, \end{aligned} \tag{9}$$

and for $s = 2$, (9) is equivalent, up to a multiplicative constant, to the Coulomb potential energy. The set $\mathbf{Z}$ in the present work is the set of objective vectors $\mathbf{z}$. Intuitively, when (9) is minimized then the mean nearest neighbour distance of the set of points $\mathbf{z}$ is maximized, subject to the constraints imposed by the geometry of the PF. For some examples on the distribution of solutions using (9) the reader is referred to [44]. We illustrate the fluctuation of energy for an increasing number of dimensions, when the weighting vectors are chosen either according to the suggestions in [14] or [7], see *Fig. (4)*. It should be noted that these schemes for weight vector selection are predominantly used in several algorithms. The results in *Fig. (4)* have been obtained in the following way:

- For 2 to 11 dimensions and for a concave PF, N number of objective vectors are selected according to generalized decomposition and the methods described in [14] and [7]. The number of selected objective vectors used in every instance can be seen in Table I. This choice is motivated by the fact that H is the number of subdivisions per dimension, so the point density of objective vectors for a constant H should represent the PF equally well, in all dimensions. The H parameter has been set to 7 because for 11 objectives the number of objective vectors, N, increases quite rapidly for a higher value of H. For instance, for $H = 8$ and $H = 9$ the number of objective vectors becomes $N = 19\,448$ and $N = 43\,758$ respectively. This increases the computational resources required for the experiment significantly.
- For each problem instance, a set of weighting vectors was generated according to the proposed methods in [14] and [7]. For generalized decomposition the weighting vectors are generated using a reference Pareto front with the desired distribution. For example, in 2 dimensions the first quadrant of a unit circle is uniformly sampled and then the optimal weighting vectors are estimated by solving (7). Also the expected energy $\mathbb{E}(E_b)$ is calculated using $N \times 50$ independent uniformly distributed samples on the PF. Details on the generation of a uniformly sampled concave PF can be found in Appendix A.
- Subsequently, using the inverse relationship to (7),

TABLE I
THE NUMBER OF OBJECTIVE VECTORS, N, FOR CONSTANT H USED IN THE EXPERIMENT SEEN IN *Fig. (4)*.

Obj. #	2	3	4	5	6	7	8	9	10	11
H	7	7	7	7	7	7	7	7	7	7
N	7	28	84	210	462	924	1716	3003	5005	8008

namely:

$$\begin{aligned} &\min_{\mathbf{F}(\mathbf{x})} \|\mathbf{F}(\mathbf{x}) \circ \tilde{\mathbf{w}}\|_\infty, \\ &\text{subject to } \sum_{i=1}^{k} f_i = 1, \\ &\text{and } f_i \geq 0 \,, \forall\, i \in \{1, \ldots, k\}. \end{aligned} \tag{10}$$

the Pareto optimal solutions $\mathbf{F}(\mathbf{x})$ that minimize every subproblem $\tilde{\mathbf{w}}$ are calculated. However, as can be seen in (10), the inverse problem to (7) can be solved only for an affine Pareto front. Although, in the case of a concave PF, the affine PF obtained by (10) can be projected onto the unit hypersphere and the obtained solutions will still be optimal for their corresponding weighting vectors.

- Lastly, the log ratio of the energy of obtained solutions for every method, E_m, and the expected energy, $\mathbb{E}(E_b)$, is calculated for all objectives in Table I.

In *Fig. (4*) it can be seen that the energy *signature* of generalized decomposition asymptotically converges to $\mathbb{E}(E_b)$, which is the expected energy of uniformly distributed solutions on the convex PF. Therefore, generalized decomposition successfully captures the underlying distribution of the target PF, so it is only a matter of convergence of the underlying algorithm to that front in order to obtain an approximation of that PF with the desired distribution. Conversely, solutions obtained using the scalarisation method employed by MOEA/D [7] or MOGLS [14], have radically different energy levels signifying a distribution of Pareto optimal solutions very different to that of the uniform. Additionally, since (9) penalizes solutions that are clustered, we can see that for 3 or more dimensions the other methods produce significantly more clustered solutions in comparison to generalized decomposition. These results do not provide superiority information of one method over all others. They do however furnish evidence that given prior information about the definition of what well distributed Pareto optimal solutions on the PF means to the DM, generalized decomposition can identify this and produce solutions distributed accordingly. Therefore, for a MOEA that is based on generalized decomposition, the three performance objectives that an EA, when applied to an MAP, has to achieve, namely – convergence, well distributed solutions along the PF and coverage of the entire PF – degenerate to only one, that of convergence. This, of course, is subject to prior knowledge of the PF shape and a definition of what *well* distributed Pareto optimal solutions mean to the DM. In Section VIII we present how this feature of generalized decomposition can be used for preference articulation.

IV. Cross Entropy Method

The cross entropy method (CE) was introduced by Rubinstein [38], for single objective continuous and discrete optimization problems. In its original form, CE was based on Kullback-Leibler cross-entropy, importance sampling and the Boltzmann distribution for continuous problems, while Markov chains are employed in the discrete case [38]. It is interesting to note that in this form CE is similar, in principle, to probability collectives (PC), a method introduced by Wolpert et al. [46] for distributed control and optimization.

In CE, the optimization problem is cast as a rare event estimation and, subsequently, an adaptive technique, with the aid of importance sampling, is applied to update the parameters of an instrumental density. The derived problem is called the *associated stochastic problem* (ASP). The method then uses the ASP to implicitly solve the original optimization problem. Generally speaking there are two steps involved in this iterative procedure,

- Generate a population[7] based on a prior distribution g. The distribution g is uniquely defined by a parameter vector v. In the initial iterations of the algorithm it is usually the uniform distribution, unless there is prior information available.
- Update the parameter vector v to create the posterior distribution using an elite subset, $\mathcal{E}$, of the previous population.

Since its introduction, several studies expanding on the initial methodology have been presented. Most notably, the minimum cross-entropy (MCE) method [47], where a non-parametric instrumental distribution is used. Albeit, MCE is computationally more demanding compared with CE. Another interesting approach,presented by Botev in [48], to extend CE is termed generalized cross entropy (GCE). In GCE, quite elegantly, the ASP is transformed to a convex program with the help of the χ^2 directed divergence. GCE overcomes the specification bias by using non-parametric density estimation. However, the computational cost of GCE is prohibitive when used in an optimization setting.

Let us assume that the optimization problem to be minimized is single objective:

$$\min_{\mathbf{x}} f(\mathbf{x}) \tag{11}$$

where $\mathbf{x}$ is the decision variable vector and $f(\mathbf{x}^\star) = \gamma^\star$ is the minimum. Assuming $\mathbf{x}^\star$ is *rare*[8] in $\mathbf{S}$, (11) can be interpreted in a different way, i.e. as a rare event estimation. Therefore (11) can be restated as follows,

$$\mathbb{E}_{\mathbf{u}} I_{f(\mathcal{X}) \leq \gamma} = \mathbf{P}_{\mathbf{u}}(f(\mathcal{X}) \leq \gamma) = \ell, \tag{12}$$

where ℓ is the probability of the *rare event*, I is the indicator function and $\mathbb{E}_{\mathbf{u}}$ is the expectation of a quantity distributed according to the density $g(\cdot; \mathbf{u})$. Also $\mathcal{X}$ is a random variable associated with the decision variable vector $\mathbf{x}$. For notational compactness we define $H(\mathcal{X}; \gamma) \equiv I_{f(\mathcal{X}) \leq \gamma}$,

$$H(\mathcal{X}; \gamma) = \begin{cases} 1 & f(\mathcal{X}) \leq \gamma \\ 0 & f(\mathcal{X}) > \gamma. \end{cases} \tag{13}$$

Now to estimate ℓ for some $\tilde{\gamma}$ that $\|\tilde{\gamma} - \gamma^\star\| \leq \epsilon$, with ϵ small, we have to solve $\mathbf{P}_{\mathbf{u}}(H(\mathcal{X}; \tilde{\gamma}))$ which is non-trivial if our initial assumption is true, i.e. that the probability $\mathbf{P}_{\mathbf{u}}(H(\mathcal{X}; \tilde{\gamma}))$ is small when $\mathcal{X} \sim g(\cdot; \mathbf{u})$. In the trivial case

[7]Note that the terms *population* and *samples* are used interchangeably in this work; unless stated otherwise.

[8]By rare in this context we mean that for, $C = \{\mathbf{x} : \|\mathbf{x}^\star - x\|_2 \leq \varepsilon, \varepsilon > 0\}$ and ε *small*, then the probability, $\mathbf{P}(\mathbf{x} \in C) = \int_C u(\mathbf{x}) d\mathbf{x} \ll 1$, where, u, is a density function.

that the aforementioned assumption is not true, ℓ can be estimated using the *crude Monte Carlo* (CMC) estimator,

$$\hat{\ell} = \frac{1}{N}\sum_{i=1}^{N} H(\mathcal{X};\gamma). \tag{14}$$

If, however, our prior assumption holds that the indicator function $I_{f(\mathcal{X})\leq\rho}$ in (14) will most likely be identically 0 for all $\mathcal{X}_i$, then a different approach is necessary. An alternative to CMC is the *importance sampling* (IS) estimator which is defined as follows,

$$\hat{\ell} = \frac{1}{N}\sum_{i=1}^{N} W(\mathcal{X}_i;\mathbf{u},\mathbf{v})H(\mathcal{X};\gamma), \tag{15}$$

where $W(\mathcal{X};\mathbf{u},\mathbf{v}) = \frac{g(\cdot;\mathbf{u})}{g(\cdot;\mathbf{v})}$ is the *likelihood ratio* (LR). Now the problem is to find the IS density $g(\cdot;\mathbf{v})$ that would minimize the variance of the estimator; theoretically the zero variance density is:

$$g^{\star}(\mathbf{x}) = \frac{f(\mathbf{x};\mathbf{u})H(\mathcal{X};\gamma)}{\ell}. \tag{16}$$

However (16) involves the quantity which we are trying to estimate (ℓ), hence its practical value is limited, although we could, up to a multiplicative constant, attempt to minimize the "distance" of $g(\cdot;\mathbf{v})$ from $g^{\star}(\cdot)$. For this purpose, a convenient measure of "distance" is the Kullback-Leibler *distance* (KL), defined as:

$$\mathcal{D}(g,h) = \int g(\mathbf{x})\ln\left(\frac{g(\mathbf{x})}{h(\mathbf{x})}\right)\,d\mathbf{x} \tag{17}$$

and upon expansion,

$$\begin{aligned}\mathcal{D}(g,h) = &\int g(\mathbf{x})\ln g(\mathbf{x})\,d\mathbf{x}\\ &-\int g(\mathbf{x})\ln h(\mathbf{x})\,d\mathbf{x}.\end{aligned} \tag{18}$$

Since the first term in (18) is constant, we only need to minimize the second term which is equivalent to maximizing $\int g(\mathbf{x})\ln h(\mathbf{x})\,d\mathbf{x}$. Therefore the optimal parameter vector $\mathbf{v}^{\star}$, in the minimum variance sense, is obtained by the solution of the following program:

$$\mathbf{v}^{\star} = \max_{\mathbf{v}} \mathbb{E}_{\tilde{\mathbf{v}}} H(\mathcal{X};\gamma)W(\mathcal{X};\mathbf{u},\tilde{\mathbf{v}})\ln g(\mathcal{X};\mathbf{v}), \tag{19}$$

where $\mathcal{X}$ is independent and identically distributed (i.i.d) according to $g(\cdot;\tilde{\mathbf{v}})$. However $\mathbf{P}_{\mathbf{u}}(H(\mathcal{X};\gamma))$ is still a rare event. In CE this is overcome by substitution of γ with $\bar{\gamma}\geq\gamma$ equal to the ρ-quantile of $f(\mathcal{X})$ under $\mathbf{v}$. The program in (19) is solved for decreasing levels of $\bar{\gamma}$ until $\bar{\gamma}\leq\gamma$. So (19), in the CE method, becomes:

$$\mathbf{v}_t = \max_{\mathbf{v}} \mathbb{E}_{\mathbf{v_{t-1}}} H(\mathcal{X};\gamma_{t-1})W(\mathcal{X};\mathbf{u},\mathbf{v}_{t-1})\ln g(\mathcal{X};\mathbf{v}), \tag{20}$$

whose stochastic counterpart is,

$$\mathbf{v}_t = \max_{\mathbf{v}} \frac{1}{N}\sum_{i=1}^{N} H(\mathcal{X}_i;\gamma_{t-1})W(\mathcal{X}_i;\mathbf{u},\mathbf{v}_{t-1})\ln g(\mathcal{X}_i;\mathbf{v}), \tag{21}$$

where $\mathcal{X}_1,\ldots,\mathcal{X}_N$ is drawn from $g(\cdot;\mathbf{v}_{t-1})$. Typically (21) is convex and if the instrumental densities $g(\cdot;\cdot)$ are chosen from the *natural exponential family* (NEF) [49], then, (21) can be solved analytically [47] by solving the following system of equations:

$$\max_{\mathbf{v}} \frac{1}{N}\sum_{i=1}^{N} H(\mathcal{X}_i)W(\mathcal{X}_i;\mathbf{u},\mathbf{v}_{t-1})\nabla_{\mathbf{v}}\ln g(\mathcal{X}_i;\mathbf{v}) = 0. \tag{22}$$

This is a major strength in CE,that is, the fact that the updating rules for the instrumental densities can be obtained analytically translates to a much lower computational overhead. Briefly, some distributions in the NEF family are the Gaussian, Poisson and the gamma distributions [50].

The procedure described by (20)-(22) will generate a monotonically nonincreasing sequence of γ values: $\{\gamma_t : t = 1,2,\ldots\}$, with the corresponding instrumental densities converging to the optimal parameter $\mathbf{v}$ for which the event $\mathbf{P}_{\mathbf{u}}(H(\mathcal{X};\tilde{\gamma}))$ is increasingly easier to estimate, i.e. becomes more *likely* under the density $g(\cdot;\mathbf{v})$.

A. CE Method for Continuous Optimization

The procedure described so far is directly applicable to optimization problems, the only difference being that the level γ is either the *a priori* minimum of the objective function $f(\cdot)$ or, if this information is not available, it is allowed to decrease *ad infinitum*. In practice, for bounded problems, the sequence $\{\gamma_t \,|\, t = 1,2,\ldots\}$ converges to a value close to the minimum, hence the stopping criterion can be set to $|\gamma_t - \gamma_{t-1}| \leq \delta$ for some small δ.

A commonly used candidate for the instrumental densities is the normal distribution,

$$g(x;\mu,\sigma) = \frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right), \tag{23}$$

and its truncated equivalent for problems with boundary constraints. We should mention that the updating rules derived using (22) are identical for the regular and truncated Gaussian [48].

It is suggested in [47] that for the optimization case, IS is not very useful since the initial parameter $\mathbf{u}$ in the density $g(\cdot;\mathbf{u})$ is actually arbitrary, under the assumption that we do not possess any information about the location of the optimum. However, such information may be available, hence maintaining the IS estimator allows prior information to be exploited. This can be achieved by setting the parameters $\mathbf{u}$ according to the information available, which should, in turn, help steer the search towards optimal solutions faster. On the downside, if the prior information is not correct, this biasing can lead the optimization procedure astray.

The CE method for single objective problems can be summarized as follows:

- **Step 1** Initialize $\mathbf{v}_0$ to the uniform distribution and set $t = 1$.
- **Step 2** Sample the distribution $g(\cdot;\mathbf{v}_{t-1})$ to generate a random sample of size N and evaluate the objective function $f(\cdot)$.
- **Step 3** Select the top ρN performing samples and use them to estimate $\mathbf{v}_t$. Solving (22) we obtain the updating

rules for the normal distribution $\mathbf{v}_t = \{\mu_t, \sigma_t\}$:

$$\hat{\mu}_t = \frac{\sum_{i=1}^{\rho N} W(\mathcal{X}_i; \mathbf{u}, \mathbf{v}_{t-1})\mathcal{X}_i}{\sum_{i=1}^{\rho N} W(\mathcal{X}_i; \mathbf{u}, \mathbf{v}_{t-1})}, \tag{24}$$

$$\hat{\sigma}_t = \left(\frac{\sum_{i=1}^{\rho N} W(\mathcal{X}_i; \mathbf{u}, \mathbf{v}_{t-1})(\mathcal{X}_i - \hat{\mu})^2}{\sum_{i=1}^{\rho N} W(\mathcal{X}_i; \mathbf{u}, \mathbf{v}_{t-1})} \right)^{\frac{1}{2}}, \tag{25}$$

where ρ is some small value, e.g. 0.1. The updating rules in (24) and (25) could lead to premature convergence [47], so a *smoothed* version is usually employed:

$$\begin{aligned} \mu_t &= \alpha\hat{\mu}_t + (1-\alpha)\mu_{t-1} \\ \sigma_t &= \beta_t\hat{\sigma}_t + (1-\beta_t)\sigma_{t-1}, \end{aligned} \tag{26}$$

where α and β_t are smoothing parameters with $\alpha \in (0.7, 1)$ and β_t is calculated as:

$$\begin{aligned} \beta_t &= \beta - \beta\left(1 - \frac{1}{t}\right)^q, \\ \beta &\in (0.7, 1), \\ q &\in (5, 9). \end{aligned} \tag{27}$$

Step 4 If the stopping condition is not met go to **Step 2**, otherwise output the current μ_t as the estimate of the location of the optimum.

V. Generalized Decomposition-Based Many Objective Cross-Entropy

The proposed algorithm is based on the CE method, see Section IV, and the newly introduced concept of generalized decomposition, as described in Section III. However we introduce two versions: many-objective CE (MACE) and MACE based on generalized decomposition (MACE-*g*D). The difference between the versions is that the weighting vectors $\mathbf{w}$ in MACE are generated according to the suggestions in [7] to enable a clearer comparison with the MOEA/D framework and evaluate the benefits and potential shortcomings of generalized decomposition. Therefore MACE employs a set of evenly spaced weighting vectors to further test validity of our hypothesis that this scheme does not result in an *even* distribution of Pareto optimal solutions on the PF, see Section III-C. We show how such issues can be overcome using MACE-*g*D and present a method that can prove invaluable when the optimization problem has many objectives. The general idea is that we can generate a set of weighting vectors near regions that are of interest, thus avoiding a waste of resources in search of Pareto optimal solutions away from such regions. The *main* algorithm in MACE and MACE-*g*D is the CE method for continuous optimization problems, as described in Section IV-A. An overview of MACE-*g*D can be seen in Algorithm 1. In line *1*, the optimal weighting vectors are obtained according to prior information about the shape of the PF and the desired distribution of Pareto optimal solutions. This procedure is comprised of two steps, namely:

Step 1 Generate a set of solutions according to the PF shape of the given problem. For example, for a concave PF this reference front could be the one depicted in *Fig. (2)*. The generation of this target front is mostly a matter of preference. To insulate the DM from different objective function scales, it is advisable that the objective functions are normalized in the range $[0, 1]$. This can be achieved if the ideal vector $\mathbf{z}^\star$ is known *a priori* or an adaptive method is used during the optimization, such as in [7]. Note that this method can be used only for bounded objective functions, since generalized decomposition in its current formulation, only applies to such functions.

Step 2 Solve (7) for every point in the reference PF generated in **Step 1** to obtain the optimal weighting vectors $\mathbf{w}$.

Algorithm 1 MACE-*g*D

1: $\{\mathbf{w}_1, \ldots, \mathbf{w}_N\} \leftarrow gD(\text{PF Shape})$
2: $\mathcal{M}^{(1)} \leftarrow \min \mathbf{x} + \mathcal{U}(0,1)(\max \mathbf{x} - \min \mathbf{x})$
3: $\mathcal{S}^{(1)} \leftarrow C(\max \mathbf{x} - \min \mathbf{x})$
4: $\mathbf{X}^{(1)} \leftarrow \mathcal{N}(\mathcal{M}, \mathcal{S})$
5: $\mathbf{E} \leftarrow \mathbf{F}(\mathbf{X}^{(1)})$
6: $\mathbf{z}^\star \leftarrow \min\{\mathbf{E}_{f_1}, \ldots, \mathbf{E}_{f_k}\}$
7: $t \leftarrow 1$
8: **repeat**
9: **for** $i \leftarrow 1, N$ **do**
10: $\mathbf{V}^{(t)} \leftarrow g_\infty(\mathbf{X}^{(t)}, \mathbf{w}_i, \mathbf{z}^\star)$
11: $Q \leftarrow Sort(\mathbf{V}^{(t)})$
12: $\mathcal{E} \leftarrow Q_{1,\ldots,\rho N}$
13: $\mathcal{M}_i^{(t)} \leftarrow \alpha\hat{\mu}_t + (1-\alpha)\hat{\mu}_{t-1}$
14: $\mathcal{S}_i^{(t)} \leftarrow \beta_t\hat{\sigma}_t + (1-\beta_t)\hat{\sigma}_{t-1}$
15: $\hat{\mathbf{x}}_i^{(t)} \leftarrow \mathcal{N}(\mathcal{M}_i^{(t)}, \mathcal{S}_i^{(t)})$
16: $\hat{\mathbf{V}}_i^{(t)} \leftarrow g^{tce}(\hat{\mathbf{x}}_i^{(t)}, \mathbf{w}_i, \mathbf{z}^\star)$
17: **if** $\hat{\mathbf{V}}_i^{(t)} \leq \mathbf{V}_i^{(t)}$ **then**
18: $\mathbf{V}_i^{(t+1)} \leftarrow \hat{\mathbf{V}}_i^{(t)}$
19: $\mathbf{x}_i^{(t+1)} \leftarrow \hat{\mathbf{x}}_i^{(t)}$
20: $\mathbf{z}^\star \leftarrow \min\left(\mathbf{z}^\star, \mathbf{F}\left(\mathbf{x}_i^{(t)}\right)\right)$
21: **end if**
22: **end for**
23: $t \leftarrow t + 1$
24: **until** $t \leq MaxGenerations$
25: $\mathbf{x} \leftarrow \mathcal{M}^{(t)}$

The reference Pareto front used in this work for the WFG4–9 test problems in Section VII-C is a uniformly distributed set on a concave front using the method described in Appendix A. For the test problem WFG3, since the front is a line in any number of dimensions, an evenly spaced set of points were selected along this line and lastly for the WFG2 problem the optimal weighting vectors are evaluated using a random sample from the true PF.

Next, in lines *2–4*, the starting population $\mathbf{X}^{(1)}$ is initialized by sampling the almost uniform distribution $\mathcal{N}(\mathcal{M}, \mathcal{S})$. In this work, for notational compactness, $\mathcal{N}(\mathcal{M}, \mathcal{S})$ has the following

meaning:

$$\begin{pmatrix} \mathcal{N}(\mu_{1,1},\sigma_{1,1}) & \cdots & \mathcal{N}(\mu_{1,n},\sigma_{1,n}) \\ \vdots & \ddots & \vdots \\ \mathcal{N}(\mu_{N,1},\sigma_{N,1}) & \cdots & \mathcal{N}(\mu_{N,n},\sigma_{N,n}) \end{pmatrix}, \quad (28)$$

where n is the number of decision variables and N the size of the population, which is the same as the number of subproblems and $\mathcal{N}$ is the truncated normal distribution in the domain of definition of the corresponding decision variables. The matrix, $\mathcal{M}^{(t)}$ contains the current estimate of the decision variables and $\mathcal{S}^{(t)}$ contains the standard deviation parameters. The $\mathcal{M}^{(t)}$ matrix is initialized at random within the decision variables' domain of definition or using some alternative method, for example Latin hypercube sampling. The $\mathcal{S}^{(t)}$ matrix is initialized to some sufficiently large value so that the truncated normal distributions tend to be approximately equal to the uniform distribution at the first iteration, given that no prior information is available. For this reason we use $C = 10$, see line *3*.

Next, the objective function, $\mathbf{F}(\cdot)$ is evaluated for the initial population $\mathbf{X}^{(1)}$ and the ideal vector $\mathbf{z}^{\star}$ is estimated using the minimum of the individual objectives in E.

The main loop of the MACE-*g*D algorithm is in lines *8–24*. At each iteration and for every subproblem, $\mathbf{w}_i$, the entire population is evaluated using the Chebyshev decomposition. The population performance, $\mathbf{V}^{(t)}$ is sorted in an ascending order[9] and the solutions in the ρ-percentile, $\mathcal{E}$, are used to update the instrumental density parameters of the i^{th} subproblem, $\mathcal{M}_i^{(t)}$ and $\mathcal{S}_i^{(t)}$. Next, a new solution, $\hat{\mathbf{x}}_i^{(t)}$, is sampled from the parametric density using the updated parameters. This new solution is evaluated and if its performance is superior to the previous solution it is retained, see lines *17–20*. The algorithm terminates once the maximum function evaluations are reached. Finally, the PF approximation set is the matrix $\mathcal{M}^{(t)}$.

MACE and MACE-*g*D have similarities with MOEA/D [7] and derivatives [51]–[53]. However there are fundamental differences which have been motivated by the results in Section III-C. Namely, MACE and MACE-*g*D do not have a mating restriction, and there is no neighbourhood in weighting vector space. In fact only the top performing individuals for every subproblem are used, irrespective of their *origin* (see Algorithm 1), namely the distribution that generated them. In contrast to that, MOEA/D derivatives insist on using a neighbourhood based on the distance of the weighting vectors. This choice seems reasonable when the relative location of the Pareto optimal solutions resulting from the set of subproblems is unknown. However, even if the Pareto front geometry is unknown *a priori*, this information can be extracted using generalized decomposition. For example, assuming an affine Pareto front geometry the neighbourhood can be calculated in objective space. The weighting vectors can be calculated using (7) and the neighbourhood structure can be as calculated for the above Pareto front. Here the assumption of an affine Pareto front is only limiting if the real Pareto front is discontinuous. However, this is also problematic for MOEA/D as defined in [7]. In any other case, the relative distance of the Pareto optimal solutions will be correct.

[9]For maximization problems, $\mathbf{V}^{(t)}$ is sorted in descending order.

VI. Benchmark Algorithms

The goal of the comparative studies in this work is not to proclaim a *best* algorithm among variants of MACE and the aforementioned frameworks. Our main aim is to explore the potential of generalized decomposition versus what is considered to be standard practice in decomposition-based MOEAs. The additional benefit is that the generalized decomposition framework seems very suitable for the extension of EDAs to MAPs, something that enables us to evaluate whether the performance of the CE method is comparable with established MOEAs. Therefore, our selection of MOEA/D as a benchmark algorithm is only natural since this algorithm framework has become a baseline for comparison of decomposition-based MOEAs. Also the good performance of RM-MEDA against other EDAs makes it a suitable candidate to evaluate the main EDA in our MACE and MACE-*g*D algorithms.

A. Multi-Objective Evolutionary Algorithm based on Decomposition

As mentioned in Section I, decomposition methods were usually applied in conjunction with gradient search methods, although there are examples of EAs based on this type of fitness assignment. One notable framework based on decomposition, introduced by Zhang et al. [7], is the MOEA/D algorithm. The original version of MOEA/D was a decomposition-based algorithm consisting of mating restriction and an archive preserving the best-so-far solution for every subproblem. The use of scalarizing functions to extend an EA to MAPs has the following benefits:

- Diversity preserving operators and *elite* preserving strategies, become, to an extent, redundant if the choice of weighting vectors and decomposition method is suitable for the problem in question.
- The computational cost tends to be lower compared to Pareto-based algorithms [7].

MOEA/D depends on one of several available decomposition techniques, - weighted sum, Chebyshev [8] and a penalty-based variant of the normal boundary intersection [7], [54] decompositions - with each having its own strengths and weaknesses. The minimization problem from Section 1, when using the Chebyshev decomposition is restated according to (6). In MOEA/D the vectors $\mathbf{w}^i$ are N evenly distributed weighting vectors. A MAP is decomposed to N subproblems using $\mathbf{w}^i$. Each individual in the population is assigned to a single subproblem, and so N is also the size of the population. For example, for a 2-objective problem, the weighting vectors are defined as:

$$w_1^i = \frac{i}{H},\ w_2^i = 1 - w_1^i,\ i \in \{0,\ldots,H\}, \quad (29)$$

where the H parameter controls the number of subdivisions per dimension and $\mathbf{w}^i = \{w_1^i, w_2^i\}$. The argument is that since g_∞ is a continuous function of $\mathbf{w}$, N evenly distributed weighting vectors should result in N evenly distributed Pareto

optimal solutions, assuming that the objectives are normalized [7]. However this argument is only valid in the case that a boundary intersection (BI) approach is used, such as the normal boundary intersection method (NBI) [54]. In NBI the following program is to be solved:

$$\begin{aligned} &\min_{\mathbf{x}} g_{nbi}(\mathbf{x}; \mathbf{w}^i, \mathbf{z}^\star) = d \\ &\text{subject to } \mathbf{z}^\star - \mathbf{F}(\mathbf{x}) = d \cdot \mathbf{w}^i, \end{aligned} \tag{30}$$

where Zhang et al. [7] suggest a penalty function approach to handle the equality constraint. Thus (30) is transformed to:

$$\begin{aligned} \min_{\mathbf{x}} g_{nbi}(\mathbf{x}; \mathbf{w}^i, \mathbf{z}^\star) &= d_1 + p d_2 \\ d_1 &= \frac{\|(\mathbf{z}^\star - \mathbf{F}(\mathbf{x}))^T \mathbf{w}^i\|_2}{\|\mathbf{w}^i\|_2}, \\ d_2 &= \|\mathbf{F}(\mathbf{x}) - (\mathbf{z}^\star - d_1 \mathbf{w}^i)\|_2, \end{aligned} \tag{31}$$

where p is a tunable parameter which controls the relative importance of convergence, d_1, and position, d_2, in the penalty function. It was shown that MOEA/D using (31) has the potential to produce truly evenly distributed Pareto optimal solutions [7]. Unfortunately (31) has three significant drawbacks. First, the normal-boundary intersection method does not guarantee that the solutions to the subproblems will be Pareto optimal [54]. Second, NBI has to be solved using a penalty method which introduces one more parameter that has to be tuned for every test problem separately, and lastly it is unclear how this decomposition method can be scaled for MAPs. A description of the MOEA/D algorithm follows:

- **Step 1** Generate N equally spaced $\mathbf{w}^i$ vectors according to (29). Create a matrix B containing the nearest neighbours of each $\mathbf{w}^i$ and initialize the ideal weighting vector $\mathbf{z}^\star$ to a large value.
- **Step 2** Evaluate the decision variable vectors $\mathbf{X}$ using the objective function.
- **Step 3** Update the ideal vector $\mathbf{z}^\star = \min(\mathbf{z}^\star, \mathbf{F}(\mathbf{x}))$.
- **Step 4** For each individual $i \in \{1, \ldots, N\}$ execute the following procedure:
- **Step 4.1** Apply genetic operators, crossover and mutation, using individuals in the neighbourhood of each solution. The choice of individuals is random among neighbouring solutions.
- **Step 4.2** Evaluate the newly generated solution using (6).
- **Step 4.3** Update the ideal vector $\mathbf{z}^\star$.
- **Step 4.4** If the new solution is superior to the previous in the archive, then swap the old solution to the i^{th} subproblem with the new solution. Otherwise, retain the old solution.
- **Step 4.5** Check if the new solution is better for any of the neighbouring subproblems and substitute if that is the case.
- **Step 5** If the termination criteria are met, output the non-dominated solutions. Otherwise, proceed to **Step 4**.

In this work the MATLAB code provided by the authors of MOEA/D is used [7].

B. Regularity Model-Based Estimation of Distribution Algorithm

The second algorithm that we employ in our comparative studies, see Section VII, is the regularity model-based multi-objective estimation of distribution algorithm (RM-MEDA) proposed by Zhang et al. [55]. The main idea in RM-MEDA is that, for continuous MAPs, the Pareto set can be viewed as a $(k-1)$-dimensional piecewise continuous manifold. So, for two dimensions, the PF can be described with line segments, for three dimensions with planes etc.

Zhang et al. [55] used inductively the Karush-Kuhn-Tucker condition [8] for continuous multi-objective problems, asserting that the PF of a problem with k objectives is defined by a $(k-1)$ dimensional manifold in the decision variable space. This assertion allowed Zhang et al. [55] to approximate this $(k-1)$ dimensional manifold with K piecewise continuous manifolds. To accomplish this task, a $(k-1)$ dimensional local principal component analysis algorithm was used to partition the population into K disjoint clusters and then the centroid and its variance were estimated. The problem with this approach is that there is no objective measure to choose the number of clusters K for an unknown problem, hence the practitioner must heavily depend on the *smoothness* of the objective function in the decision space. In contrast, if it is known *a priori* that the MAP fulfils the smoothness criteria then RM-MEDA will be able to exploit that structure and thus converge much faster.

In [55] RM-MEDA was evaluated against PCX-NSGA-II [56], GDE3 [57] and MIDEA [58], on average, outperforming the aforementioned algorithms on variants of the ZDT[10] test problems [30]. However the performance of RM-MEDA comes at the expense of increased computational cost due to the necessity of computing a local principal component analysis at each iteration. The implementation of RM-MEDA that is employed in this work is the publicly available version in MATLAB code provided by the authors [55].

C. Random Search

Random search is regarded as the absolute baseline algorithm in MOEAs. In random search, absolutely no prior assumptions are made about the problem and, during the optimization, the search is not affected by the *fitness* of the previous samples. Random search with memory, that is an algorithm that samples uniformly the decision variable space but does not revisit solutions previously sampled, enjoys asymptotical convergence [59]. However, since there is no mechanism to *steer* the search, the time to convergence is proportional to the problem complexity. Conversely, due to its simplicity and inability to *learn*, it cannot be misled by the problem. The random search algorithm employed in the current work is in its most basic form. The objective function is evaluated for $25\,000$ uniformly sampled decision variable combinations, then the non-dominated solutions are extracted and a randomly selected subset is chosen for evaluation using the methodology described in Section VII.

[10] Zitzler, Deb, Thiele (ZDT)

TABLE II
VALUE OF THE H PARAMETER IN MOEA/D AND MACE AND THE CORRESPONDING POPULATION SIZE N. THE POPULATION SIZE IS THE SAME FOR ALL ALGORITHMS. $|\mathcal{P}^\star|$ IS THE SIZE OF THE PARETO FRONT REFERENCE SET, SOLUTIONS IN THIS SET ARE UNIFORMLY DISTRIBUTED ALONG THE PF.

Obj. #	2	3	4	5	6	7	8	9	10	11
H	101	20	10	7	6	5	5	5	5	5
N	101	210	220	210	252	210	330	495	715	1001
$\|\mathcal{P}^\star\|$	500	1000	1500	2000	2500	3000	3500	4000	4500	5000

TABLE III
SETTINGS FOR MACE AND MACE-*g*D.

ρ	α	β	q
0.1	0.9	0.9	7

VII. COMPARATIVE STUDIES

A. Performance Indicator

The main performance metric for the comparative studies in this work is the generational distance (GD) indicator. This metric has been chosen since we are mainly interested in the convergence properties of the studied algorithms.

- Generational Distance (GD), introduced in [60], is defined thus:

$$D(A,\mathcal{P}^\star)=\frac{\sum\limits_{s\in A}\min\{\|\mathcal{P}_1^\star-s\|_2,\ldots,\|\mathcal{P}_N^\star-s\|_2\}}{|A|} \tag{32}$$

  where $|\mathcal{P}^\star|$ is the cardinality of the set $\mathcal{P}^\star$. The GD metric measures the distance of the elements in the set A from the nearest point of the reference PF. A is an approximation of the true Pareto front and $\mathcal{P}^\star$ is the reference Pareto optimal set.

B. Experiment Description

In Section III, it was explained that the three objectives that MOEAs have to achieve – namely convergence, diversity and PF coverage – can be reduced to only one, convergence, in the generalized decomposition framework. Therefore, the most important quantity of interest becomes some measure of convergence to the PF. For this reason, the GD metric was used, see (32).

The problem set chosen to perform the experiments is the WFG toolkit [28], specifically problems WFG2–WFG9, since they contain several challenging problems, are scalable and the PFs are known *a priori*. For all test instances we used 32 decision variables and the k parameter is calculated as: $k=4+2\cdot(M-1)$, the only exception being the 2-objective instances of the test problems where it is set to 4; M is the number of objectives. The neighbourhood size T in MOEA/D was selected to be 10% of the population size N, since, according to [12], this appears to be a setting producing good results for MAPs. The population size was the same for all the algorithms, see Table II. The parameters of the CE method are the same in MACE and MACE-*g*D and have been selected according to the suggestions in [47], see Table III. Lastly, the reference Pareto fronts used in MACE-*g*D to produce the *optimal* weighting vectors for the test instances WFG2 and WFG3 were generated by a random sample of the true Pareto set and, for the problems WFG4–WFG9, the method described in Appendix A was employed for generating a concave Pareto optimal set.

In practice such information is usually not available before the application of the optimization algorithm. This problem can be addressed using an identification method to determine the PF shape during the optimization; the methodology to be adopted will be investigated in future research.

Finally, as is probably evident from the selection of the reference PF for the generation of the weighting vectors in MACE-*g*D, we assume that the DM is interested in a PF that is uniformly distributed on that front. This is due to several considerations. First, if we follow the method usually applied in MOEA benchmarking for generating the reference PF of concave geometry, say for 3 dimensions, i.e. generate a set of evenly distributed weighting vectors and then project onto the first octant of the unit sphere, then for higher dimensions, due to the curvature of the hypersphere this will induce a large bias in the reference set. Namely, the density of Pareto optimal solutions will be higher near the edges of the PF compared to the density near the centre. Conversely, to produce a truly even distribution of Pareto optimal solutions in high dimensions is still an unresolved issue for an arbitrary number of points, even for PFs that have simple geometry, see [44], [45].

C. Experiment Results

A summary of the GD-metric performance of the algorithms is presented in Tables IV–XI. The values in bold indicate the best performing algorithm for the particular instance of a test problem. We used the Kruskal-Wallis test at the 95% confidence level to verify whether the mean performance of the studied algorithms is different. For each algorithm and for each problem instance we used the Wilcoxon two-sided rank sum test for $\alpha=0.05$ (95% confidence level). Every time an algorithm outperforms another in the test group, for a test instance, a 1 was added to its rank. Since we have 5 algorithms, the maximum rank for an algorithm is 4. A rank of 4 means that the algorithm in question performs better than all other algorithms for that particular test instance. In the case that no algorithm is clearly better we have a tie thus both algorithms are displayed in bold in Table IV–Table XI. An algorithm with a rank of 4 is denoted with a (1), one with a rank of 3 with a (2) and so forth, with (1) denoting the best performing algorithm and (5) the worst performer. These values are recorded to the right of the GD-metric performance in Tables IV–XI.

Table IV presents the results of the algorithms for 2–11 objective instances of the WFG2 test problem. WFG2 has the following features – it is non-separable, unimodal with respect to all objectives except the last which is multi-modal, there is no bias in the parameters and the PF geometry is piecewise convex. In this problem, MACE-*g*D performance is significantly better than the other algorithms for MOPs having more than 4 objectives. We attribute this performance to the fact that, for PFs that have a convex geometry, the optimal

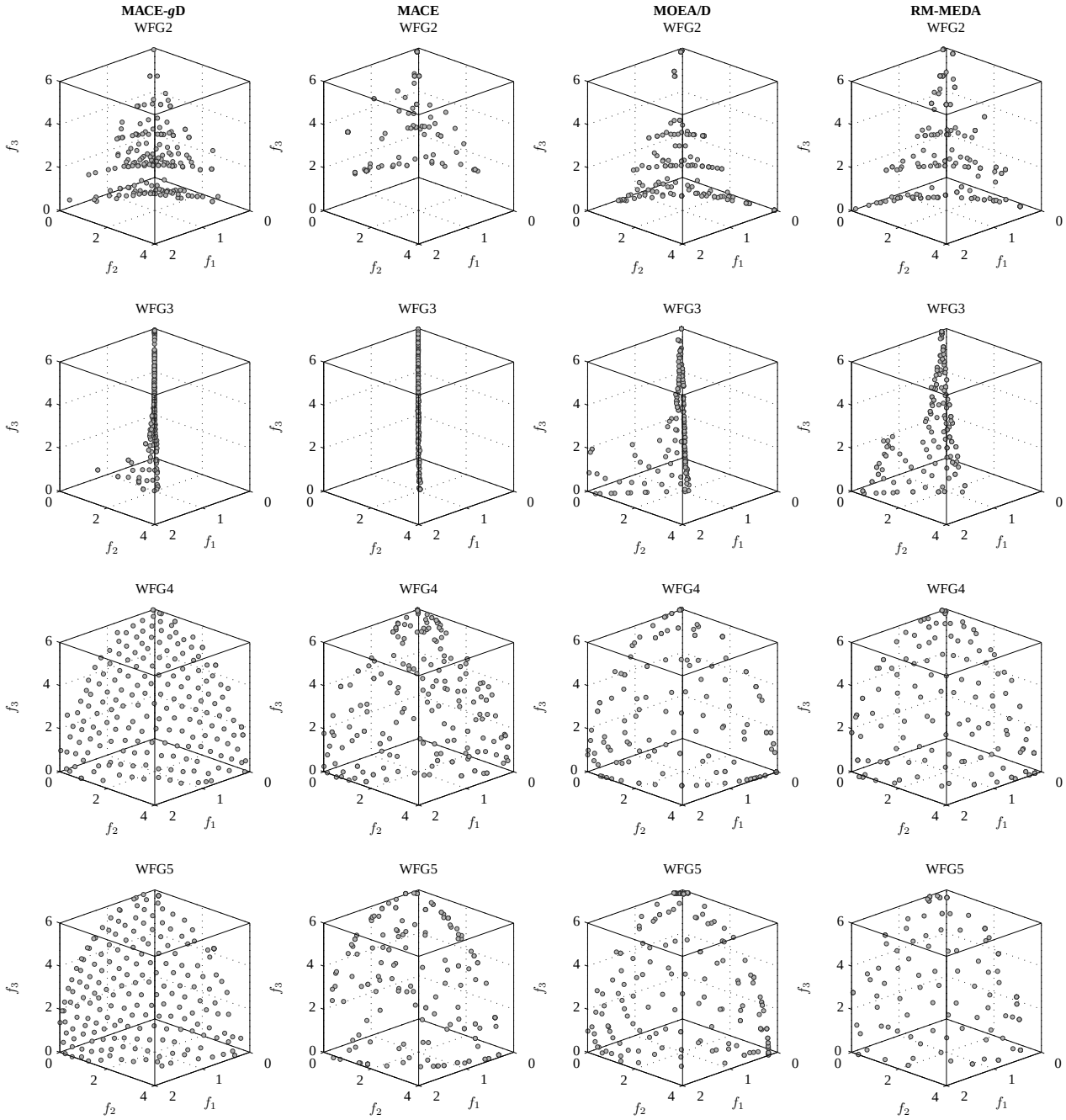

Fig. 5. MACE-**g**D, MACE, MOEA/D and RM-MEDA Pareto front for 3-objective instances of the WFG2–WFG5 test problems.

TABLE IV
GD-METRIC PERFORMANCE OF THE STUDIED ALGORITHMS ON THE WFG2 PROBLEM FOR 2–11 OBJECTIVES.

	WFG2				
Obj. #	MACE	MACE-*g*D	MOEA/D	RM-MEDA	RAND
2	0.0816 (3)	0.1027 (4)	0.0656 (2)	**0.0279** (1)	0.1687 (5)
3	**0.0353** (1)	0.0386 (2)	0.0444 (3)	0.0794 (4)	0.1929 (5)
4	0.0712 (2)	**0.0485** (1)	0.1283 (4)	0.1274 (3)	0.1998 (5)
5	0.0718 (2)	**0.0471** (1)	0.1717 (4)	0.1674 (3)	0.2125 (5)
6	0.0573 (2)	**0.0423** (1)	0.1489 (3)	0.1979 (4)	0.2228 (5)
7	0.0650 (2)	**0.0487** (1)	0.1081 (3)	0.2152 (4)	0.2335 (5)
8	0.0525 (2)	**0.0379** (1)	0.0806 (3)	0.2434 (4)	0.2649 (5)
9	0.0471 (2)	**0.0286** (1)	0.0791 (3)	0.2563 (4)	0.2638 (5)
10	0.0495 (2)	**0.0168** (1)	0.0658 (3)	0.2694 (4)	0.2785 (5)
11	0.0453 (2)	**0.0108** (1)	0.0814 (3)	0.2793 (4)	0.2867 (5)

TABLE V
GD-METRIC PERFORMANCE OF THE STUDIED ALGORITHMS ON THE WFG3 PROBLEM FOR 2–11 OBJECTIVES.

	WFG3				
Obj. #	MACE	MACE-*g*D	MOEA/D	RM-MEDA	RAND
2	**0.0133** (1)	0.0194 (3)	0.0190 (3)	0.0215 (4)	0.2108 (5)
3	0.0699 (2)	**0.0231** (1)	0.1553 (3)	0.2419 (4)	0.2899 (5)
4	0.0841 (2)	**0.0338** (1)	0.2422 (3)	0.3474 (5)	0.3204 (4)
5	0.1023 (2)	**0.0230** (1)	0.3137 (3)	0.3885 (5)	0.3311 (4)
6	0.1146 (2)	**0.0209** (1)	0.2701 (3)	0.4091 (5)	0.3312 (4)
7	0.1033 (2)	**0.0340** (1)	0.2122 (3)	0.4346 (5)	0.3321 (4)
8	0.0921 (2)	**0.0290** (1)	0.1912 (3)	0.4356 (5)	0.3350 (4)
9	0.0848 (2)	**0.0237** (1)	0.1728 (3)	0.4342 (5)	0.3364 (4)
10	0.0760 (2)	**0.0135** (1)	0.1512 (3)	0.4314 (5)	0.3371 (4)
11	0.0702 (2)	**0.0117** (1)	0.1317 (3)	0.4283 (5)	0.3379 (4)

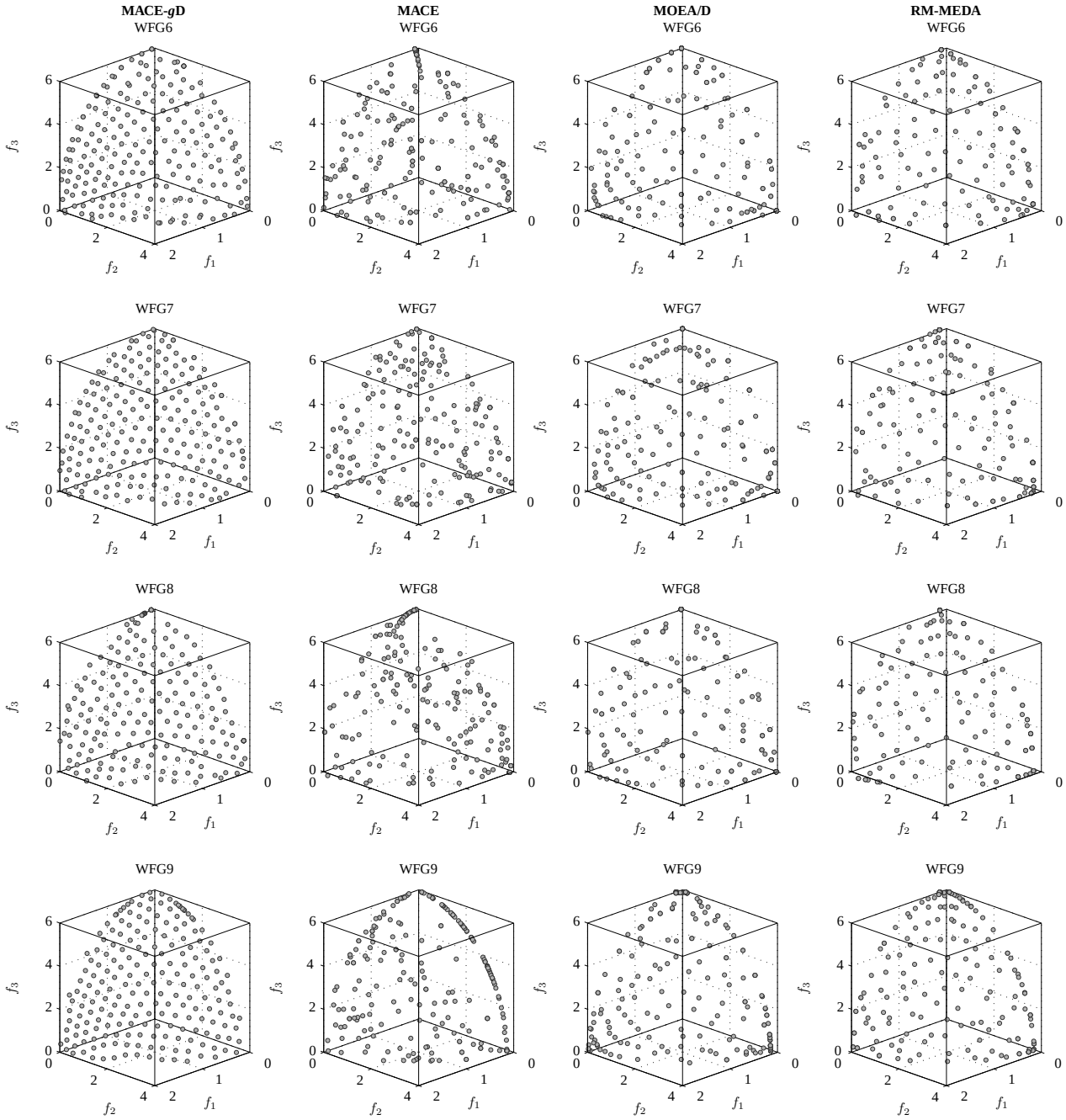

Fig. 6. MACE-**g**D, MACE, MOEA/D and RM-MEDA Pareto front for 3 objective instances of the WFG6–WFG9 test problems.

TABLE VI
GD-METRIC PERFORMANCE OF THE STUDIED ALGORITHMS ON THE WFG4 PROBLEM FOR 2–11 OBJECTIVES.

	WFG4				
Obj. #	MACE	MACE-*g*D	MOEA/D	RM-MEDA	RAND
2	0.0345 (3)	0.0344 (3)	**0.0211** (1)	0.0392 (4)	0.1161 (5)
3	0.0617 (3)	0.0522 (2)	**0.0316** (1)	0.0939 (4)	0.1302 (5)
4	0.0749 (3)	0.0740 (2)	**0.0655** (1)	0.1336 (4)	0.1358 (5)
5	0.1438 (3)	**0.1048** (1)	0.1653 (5)	0.1464 (4)	0.1407 (2)
6	**0.1358** (1)	0.1414 (2)	0.1959 (5)	0.1668 (4)	0.1549 (3)
7	0.2349 (4)	0.1997 (3)	0.2739 (5)	0.1898 (2)	**0.1770** (1)
8	0.3176 (4)	0.2351 (3)	0.3371 (5)	0.2172 (2)	**0.2025** (1)
9	0.3995 (5)	0.3028 (3)	0.3958 (4)	**0.2495** (1)	0.2568 (2)
10	0.3791 (4)	0.3265 (3)	0.4001 (5)	0.2718 (2)	**0.2577** (1)
11	0.4839 (5)	0.3875 (3)	0.4644 (4)	**0.3162** (1)	0.3540 (2)

TABLE VII
GD-METRIC PERFORMANCE OF THE STUDIED ALGORITHMS ON THE WFG5 PROBLEM FOR 2–11 OBJECTIVES.

	WFG5				
Obj. #	MACE	MACE-*g*D	MOEA/D	RM-MEDA	RAND
2	0.0393 (2)	0.0523 (4)	**0.0276** (1)	0.0433 (3)	0.1947 (5)
3	0.1052 (3)	0.0962 (2)	**0.0321** (1)	0.2168 (5)	0.2114 (4)
4	0.1533 (2)	0.1845 (3)	**0.0655** (1)	0.2652 (5)	0.2268 (4)
5	**0.1537** (2)	0.2221 (3)	**0.1540** (2)	0.2604 (5)	0.2307 (4)
6	0.1579 (2)	0.2313 (3)	**0.1558** (1)	0.2556 (5)	0.2346 (4)
7	**0.1872** (1)	0.2286 (2)	0.2455 (4)	0.2588 (5)	0.2372 (3)
8	0.2620 (3)	**0.2340** (1)	0.3262 (5)	0.2646 (4)	0.2441 (2)
9	0.3357 (4)	0.2685 (2)	0.4007 (5)	0.2748 (3)	**0.2598** (1)
10	0.3497 (4)	0.2789 (2)	0.3813 (5)	0.2911 (3)	**0.2706** (1)
11	0.4479 (4)	0.3203 (3)	0.4792 (5)	0.3096 (2)	**0.3036** (1)

TABLE VIII
GD-METRIC PERFORMANCE OF THE STUDIED ALGORITHMS ON THE WFG6 PROBLEM FOR 2–11 OBJECTIVES.

WFG6					
Obj. #	MACE	MACE-*g*D	MOEA/D	RM-MEDA	RAND
2	**0.0162** (2)	0.0226 (3)	0.0293 (4)	**0.0164** (2)	0.2465 (5)
3	0.0489 (2)	0.0499 (3)	**0.0318** (1)	0.1417 (4)	0.2666 (5)
4	0.0782 (2)	0.0836 (3)	**0.0624** (1)	0.2441 (4)	0.2865 (5)
5	0.1459 (2)	**0.1182** (1)	0.1644 (3)	0.2532 (4)	0.2940 (5)
6	0.1960 (3)	**0.1491** (1)	0.1962 (3)	0.2574 (4)	0.2936 (5)
7	0.2531 (3)	**0.1897** (1)	0.2506 (2)	0.2608 (4)	0.2881 (5)
8	0.3094 (4)	**0.2215** (1)	0.3234 (5)	0.2759 (2)	0.2885 (3)
9	0.3890 (5)	**0.2716** (1)	0.3520 (4)	0.2888 (2)	0.2951 (3)
10	0.3762 (5)	**0.3004** (1)	0.3758 (5)	0.3078 (3)	0.3032 (2)
11	0.4632 (5)	0.3577 (3)	0.4233 (4)	0.3257 (2)	**0.3201** (1)

TABLE IX
GD-METRIC PERFORMANCE OF THE STUDIED ALGORITHMS ON THE WFG7 PROBLEM FOR 2–11 OBJECTIVES.

WFG7					
Obj. #	MACE	MACE-*g*D	MOEA/D	RM-MEDA	RAND
2	0.0075 (2)	0.0144 (3)	**0.0040** (1)	0.0158 (4)	0.1707 (5)
3	0.0363 (3)	0.0309 (2)	**0.0261** (1)	0.1159 (4)	0.1889 (5)
4	0.0819 (3)	0.0740 (2)	**0.0732** (1)	0.1742 (4)	0.1998 (5)
5	0.1374 (2)	**0.1086** (1)	0.1760 (3)	0.1915 (4)	0.2013 (5)
6	0.1541 (2)	**0.1434** (1)	0.2150 (5)	0.2050 (4)	0.2046 (4)
7	0.2587 (4)	**0.1889** (1)	0.2839 (5)	0.2191 (3)	0.2142 (2)
8	0.3269 (4)	0.2282 (2)	0.3704 (5)	0.2432 (3)	**0.2270** (1)
9	0.3954 (4)	0.2838 (3)	0.4359 (5)	0.2632 (2)	**0.2508** (1)
10	0.3803 (4)	0.3092 (3)	0.4052 (5)	0.2844 (2)	**0.2633** (1)
11	0.4812 (4)	0.3704 (3)	0.4875 (5)	**0.3115** (1)	0.3153 (2)

weighting vector set, see *Fig. (3)*, is clustered near the centre region. So, using an even distribution of weighting vectors, the effective number of Pareto optimal solutions for which these vectors are optimal is reduced. This is especially true in higher dimensions, since the features seen in *Fig. (3)* are only accentuated. However, the MACE algorithm that utilized the same weighting vector selection as MOEA/D, outperforms the latter algorithm for all instances except the 2-objective case. This, in combination with the fact that MOEA/D consistently outperforms RM-MEDA, except for the 2-objective instance, leads to the hypothesis that Pareto-based algorithms potentially are not very well suited for problems with convex PF geometries in high dimensions. This hypothesis is further supported by the fact that RM-MEDA uses a variant of non-dominated sorting [55]. So, for high dimensions, the closer the estimated PF is to the true PF, the fewer are the solutions that are part of the first and second non-dominated fronts, which means that the availability of *good* solutions to the model creation process is reduced in RM-MEDA. Therefore, the closer the algorithm is to the actual PF, the more difficult it becomes for further progress to be achieved.

The results for the WFG3 instances are given in Table V. The WFG3 problem is non-separable, unimodal with no bias in the parameters and its PF geometry is affine degenerate, i.e. the front is always a line for any number of dimensions. In this problem as well, the MACE-*g*D algorithm has the superior performance, except for the 2-objective instance, where the performance of all algorithms is comparable. However MACE has statistically better performance for 2 objectives. We believe that MACE-*g*D outperforms other approaches on the WFG3 problems mainly due to the PF geometry. Since the PF geometry is affine, if we have the optimal weighting vectors then the algorithm directly attempts to converge to this location, while other algorithms are exploring the search space under the assumption that the front is some hyper-surface which is to be populated with solutions. This focus illustrates the potential advantages of generalized decomposition. Also encouraging is the fact that MACE performs very well, which means that, if the information about the geometry of the PF is not very accurate, the algorithm can still achieve acceptable results. Additionally the results of RM-MEDA on WFG3 further support our previous hypothesis about its selection scheme, notably its performance is much degraded compared to WFG2. Lastly, a curiosity is that for increasing number of dimensions, MACE-*g*D is not only better compared with other algorithms but the GD metric becomes smaller, something that is counter intuitive. However, the explanation is rather simple, namely, since WFG3 is a line in any number of dimensions, the necessity of employing a larger population is diminished. Since the population size is increased, and we know exactly the optimal weighting vectors, the density of solutions along the WFG3 PF is effectively increased, hence the decrease in the mean of the GD metric. In Table VI the results for the WFG4 problem are presented. WFG4 is a separable problem, multi-modal with no bias and its PF geometry is concave. In this problem the major influence on algorithm performance seems to be the fact that this problem is multi-modal. From the MACE and MACE-*g*D perspective, the fact that the instrumental densities used are Gaussian appears to have a significant effect. Namely, the multi-modal nature of the problem is misleading to all of the algorithms. However, the more elaborate model employed in RM-MEDA helps the algorithm scale much better compared with the other algorithms. This conclusion is based on the performance of random search on this problem and the fact that RM-MEDA follows this much more *smoothly* relative to all other algorithms. For example, for the 11 objective instance, while random search achieves a mean value for the GD-metric of 0.3540, MACE-*g*D, MOEA/D and MACE have much worse performance. The positive effect of generalized decomposition, however, is clearly visible when comparing MACE-*g*D to MACE. For instances with 2–4 objectives, MOEA/D exhibits the best performance, however it is closely followed by

TABLE X
GD-METRIC PERFORMANCE OF THE STUDIED ALGORITHMS ON THE WFG8 PROBLEM FOR 2–11 OBJECTIVES.

WFG8					
Obj. #	MACE	MACE-*g*D	MOEA/D	RM-MEDA	RAND
2	0.0598 (2)	0.0697 (3)	**0.0582** (1)	0.0875 (4)	0.2043 (5)
3	0.0857 (3)	0.0797 (2)	**0.0562** (1)	0.1671 (4)	0.2147 (5)
4	0.1201 (3)	0.1165 (2)	**0.0790** (1)	0.2596 (5)	0.2436 (4)
5	0.1453 (2)	**0.1349** (1)	0.1966 (3)	0.2982 (5)	0.2635 (4)
6	0.1835 (2)	**0.1528** (1)	0.1961 (3)	0.3005 (5)	0.2657 (4)
7	0.2524 (2)	**0.1888** (1)	0.2804 (4)	0.3002 (5)	0.2652 (3)
8	0.3214 (4)	**0.2237** (1)	0.3594 (5)	0.3134 (3)	0.2703 (2)
9	0.3762 (4)	**0.2706** (1)	0.3929 (5)	0.3246 (3)	0.2852 (2)
10	0.3698 (4)	0.2995 (2)	0.4050 (5)	0.3401 (3)	**0.2912** (1)
11	0.4669 (5)	0.3601 (3)	0.4658 (4)	0.3560 (2)	**0.3254** (1)

MACE-*g*D and MACE. This leads to the hypothesis that a more elaborate EDA coupled with generalized decomposition could potentially overcome the difficulties present in problems similar to WFG4. Table VII presents the results for the WFG5 problem. WFG5 is a unimodal, separable and deceptive problem with no bias and a concave PF. It is most interesting that for this test problem, contrary to what we anticipated, RM-MEDA performs consistently worse than random search, the only exception being the 2-objective test instance. However for more than 9 objectives, random search out-performs the other algorithms. Also, when compared with RM-MEDA, both MACE and MACE-*g*D perform significantly better for all instances with 2–10 objectives, a fact that supports the theory presented in [36] that EDAs using low order statistics with some form of clustering have potential. Of course, clustering is not used in these versions of the MACE algorithm; this is the subject of future research. Another important feature is that MOEA/D strongly outperforms all algorithms on this test problem for 2–6 objectives although its performance is heavily degraded for larger numbers of objectives, performing much worse than random search. This rapid relative degradation in performance is not seen in MACE. We believe that this phenomenon has to do with the control parameters in MOEA/D, leading us to the conclusion that MACE, MACE-*g*D and RM-MEDA are more robust with respect to their controlling parameters. This is in accord with recent studies that show that the sweet spot of configuration parameters *shrinks* with an increase in problem dimension [61].

Table VIII presents the results of the GD-metric performance for the WFG6 test problem. WFG6 is a non-separable, unimodal problem with no bias and concave PF geometry. These results further strengthen the hypothesis that the CE method performs very well on unimodal problems. Generally, the performance of MACE and MACE-*g*D over all test problems that are unimodal is similar, see Table VII-Table X. The exception to this is WFG3. However the geometry of WFG3 is influencing the performance of the algorithms greatly, so that MACE-*g*D, which has prior information of the *correct* direction of search can exploit this feature. In WFG6, RM-MEDA performs worse than random search for all instances except the 2-objective one. We believe that this phenomenon has to do with the fact that this problem non-separable, as is the case for WFG2–3 and WFG8–9, see Table IV–Table V and Table X–Table XI. For 2–3 objectives MOEA/D has superior performance to all algorithms and for 4–10 objectives MACE-*g*D is the top performer. It is interesting to note that, in that range of objectives, MACE and MOEA/D exhibit similar performance, which further suggests that the decomposition method has a strong influence on algorithm performance.

TABLE XI
GD-METRIC PERFORMANCE OF THE STUDIED ALGORITHMS ON THE WFG9 PROBLEM FOR 2–11 OBJECTIVES.

	WFG9				
Obj. #	MACE	MACE-*g*D	MOEA/D	RM-MEDA	RAND
2	0.0223 (2)	0.0259 (3)	0.0286 (4)	**0.0179** (1)	0.1925 (5)
3	0.0390 (3)	**0.0366** (2)	**0.0365** (2)	0.0657 (4)	0.2410 (5)
4	0.0653 (3)	**0.0592** (1)	0.0607 (2)	0.1636 (4)	0.2764 (5)
5	0.1494 (3)	**0.0987** (1)	0.1468 (2)	0.2442 (4)	0.2982 (5)
6	0.1441 (3)	**0.1349** (1)	0.1369 (2)	0.2655 (4)	0.3073 (5)
7	0.2193 (2)	**0.1843** (1)	0.2270 (3)	0.2769 (4)	0.3070 (5)
8	0.3055 (4)	**0.2223** (1)	0.3122 (5)	0.2889 (2)	0.3058 (4)
9	0.3657 (4)	**0.2742** (1)	0.3685 (5)	0.3039 (2)	0.3110 (3)
10	0.3514 (4)	**0.2999** (1)	0.3547 (5)	0.3214 (3)	0.3199 (2)
11	0.4473 (4)	0.3488 (3)	0.4506 (5)	0.3416 (2)	**0.3346** (1)

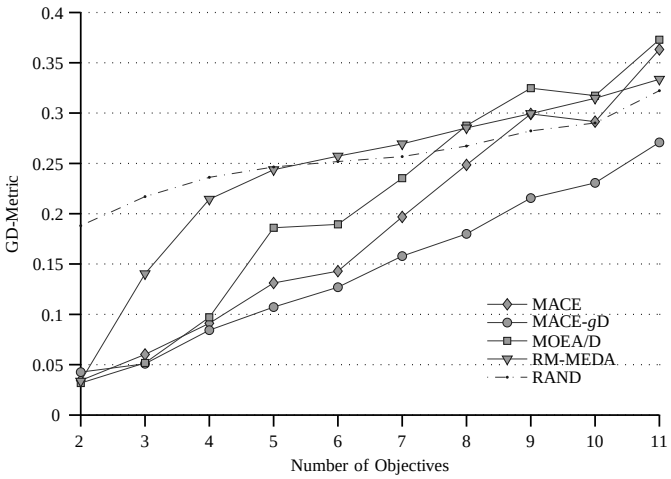

Fig. 7. Mean GD-metric performance of studied algorithms over WFG2–9 for 2–11 objectives.

Table IX and Table X correspond to the mean GD-metric value of the compared algorithms for the problems WFG7 and WFG8. The demonstrated performance is similar to the results reported in Table IV–Table VIII.

Lastly, Table XI presents the results for the WFG9 test problem which is non-separable, multi-modal and deceptive. WFG9 has also parameter dependent bias and its PF geometry is concave. Based on what we have observed in Table VI, also a multi-modal problem, the results here are counter-intuitive, especially given the fact that WFG9 is not only multi-modal but it is also deceptive. For this reason we anticipated that RM-MEDA would be the top performer. Instead, for more than ~ 6 objectives the performance of RM-MEDA is very close to that of random search and worse in the last two instances, i.e. for 10 and 11 objectives. In contrast, for 3–7 objectives MACE, MACE-*g*D and MOEA/D have relatively similar performance – with MACE-*g*D in the lead. For 8–10 objectives this lead is significantly increased and this is attributed to generalized decomposition, since the performance of the CE method for multi-modal problems is moderate, or so it would seem.

D. Sensitivity of MACE and MACE-gD to the ρ Parameter

Although a complete sensitivity analysis of algorithm performance with respect to all control parameters in the MACE and MACE-*g*D algorithms is beyond the scope of this work, it is important that we investigate how convergence is affected by the ρ parameter. This parameter controls the percentage of the individuals in the previous generation that are used in the updating process of the μ and σ parameters of the instrumental densities in the CE method. Intuitively, since every instrumental density is sampled only once for every subproblem, this parameter controls the amount of information

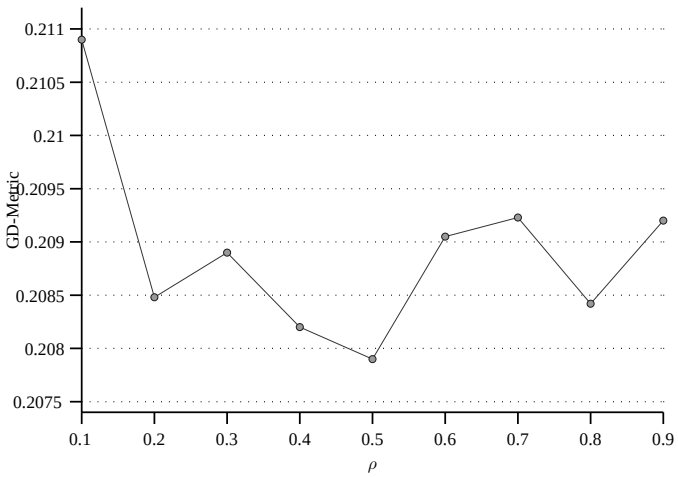

Fig. 8. Mean GD-metric performance of MACE, over all objectives for the WFG9 test problem.

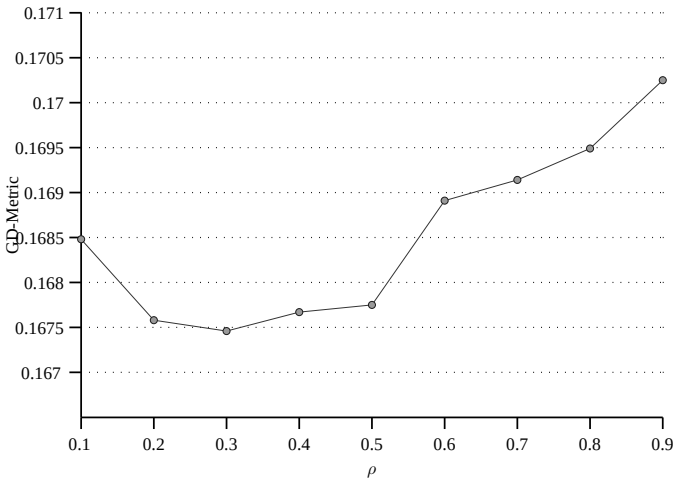

Fig. 9. Mean GD-metric performance of MACE-*g*D, over all objectives for the WFG9 test problem.

sharing between different subproblems. In that context it is similar to the T parameter in MOEA/D. However the *neighbourhood* for the MACE algorithms does not depend on the closeness of weighting vectors but depends only on the similarity of performance of different subproblems. Hence, it is not fixed as it is in MOEA/D.

To test how the GD metric performance of MACE and MACE-*g*D is affected for various values of ρ, 50 independent trials were performed for $\rho = \{0.1, 0.2, \ldots, 0.9\}$ on the WFG9 problem. All other parameters are identical to those employed in Section VII-C. The results can be seen in *Fig. (8)* – *Fig. (10)*. In *Fig. (8)* and *Fig. (9)* the mean performance of the two algorithms over 2–11 objectives for different values of the ρ parameter is illustrated. The fact that the mean performance of MACE-*g*D, see *Fig. (9)*, is better when compared with MACE, see *Fig. (8)*, is expected, given the results in Table XI. MACE and MACE-*g*D exhibit similar variation in terms of their GD metric performance for the selected range of ρ. Namely the absolute value of the difference of the best performance less the worse one as seen in *Fig. (8)* and *Fig. (9)* is 2.79×10^{-3} and 2.96×10^{-3} for MACE and MACE-*g*D respectively. A comparison of these values with the absolute performance of the above algorithms shown in *Fig. (10)*, suggests that MACE and MACE-*g*D are relatively robust to variations in the ρ parameter. Specifically, the mean performance over all objectives of MACE and MACE-*g*D for the WFG9 problem is 0.2109 and 0.1685 respectively which means that for $\rho \in \{0.1, \ldots, 0.9\}$ the variation in performance with respect to the GD metric of MACE and MACE-*g*D is 1.32% and 1.75% respectively. However their behaviour is qualitatively different.

MACE performs relatively better for all values of $\rho > 0.2$ with no consistent degradation or improvement past this threshold. Therefore any value for ρ that is greater than 0.2 should produce acceptable results. In contrast to MACE, the performance of MACE-*g*D varies in a much more coherent manner for different values of ρ, and, in general for $\rho < 0.5$ it performs consistently better than for $\rho > 0.5$. The lack of *coherency* in the improvement (or degradation) in GD performance for MACE could suggest that the algorithm is not affected as much as MACE-*g*D, by the ρ parameter. The question is: why is MACE less susceptible to variations in ρ? Our hypothesis is that, since the weighting vectors in MACE are selected in the same fashion as in MOEA/D, subproblems are aggregated in a very small region of the PF, therefore sharing information with neighbouring solutions is less disruptive, for instance, see *Fig. (2)*. Conversely, the weighting vectors in MACE-*g*D are distributed according to a uniformly distributed Pareto front, so that, as we increase ρ, the less likely it is to obtain *local* information from faraway solutions. Hence the convergence rate of the algorithm is somewhat inhibited for large ρ.

Additionally, the GD-performance of MACE-*g*D appears to be a quasi-convex function of ρ, see *Fig. (9)*. We believe this is due to the presence of two competing trends in MACE-*g*D. First, as we increase ρ, more samples are used in the updating rules in (24) and (25), hence better estimates are obtained. However, past a certain value for ρ, which for the selected problem set appears to be somewhere between $(0.5, 0.6)$, the GD-metric performance starts to degrade. This degradation is due to the second trend. As we increase ρ, samples obtained by disparate subproblems are used in the updating process, hence convergence to the PF becomes slower. This is consistent with the hypothesis that generalized decomposition successfully captures the density of the PF reference set used to generate the optimal weighting vectors.

In *Fig. (10)* the mean GD performance is illustrated over all ρ values for increasing number of objectives. Again, this result is consistent with the experiments in Section VII-C. Additionally, it seems that the linear scaling of performance of the MACE-*g*D algorithm as seen in *Fig. (7)*, is persistent for a range of ρ values.

VIII. Preference Articulation

Apart from convergence in MOEA algorithms, which is a relatively well defined concept, there can be no consensus on the meaning of a *well* distributed Pareto set. Apart from the

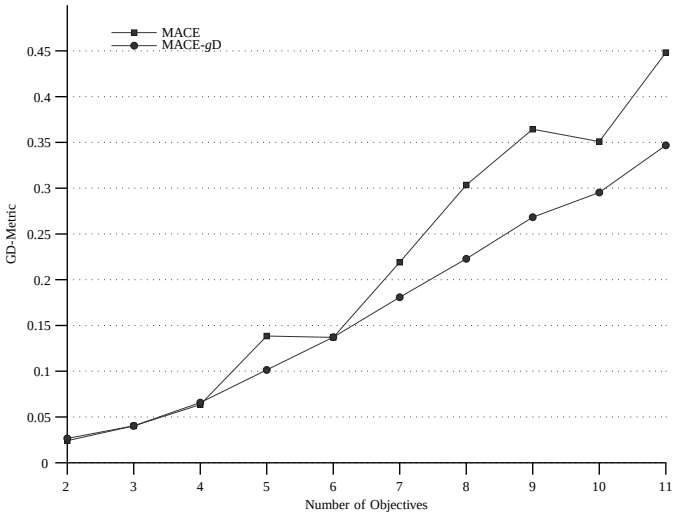

Fig. 10. Mean GD-metric performance of MACE and MACE-*g*D, over all ρ values for the WFG9 test problem.

theoretical difficulties, a proper definition of a well distributed PF cannot be given, mainly because it is contingent on the preferences of the decision maker (DM). Of what use would a Pareto optimal set be, if the solutions that are of interest to the DM are sparsely sampled, if at all.

Generalized decomposition can be employed very effectively to resolve this problem, given that some information is available *a priori* about the general shape of the PF. To illustrate this we used the 3-objective instances of WFG2–9 with an evenly distributed reference PF for the generation of weighting vectors in MACE-*g*D, see *Fig. (5*) and *Fig. (6*). As can be seen, the solutions produced by MACE-*g*D are more evenly distributed compared with MOEA/D or RM-MEDA. It should be noted that, apart from a different reference PF for the generation of weighting vectors, all algorithm parameters are identical with the ones used in Section VII-C. Furthermore, we also used a 3-objective DTLZ2 instance, a test problem with concave PF, and selected manually a set of regions on an artificially generated PF, see *Fig. (11*). These regions represent the desired parts of the PF, potentially because other parts are of no interest to the DM. The set of points seen in the left figure in *Fig. (11*) is the set,

$$C = C_1 \cup C_2 \cup C_3 \cup C_4,$$

and the sets C_1, C_2, C_3, C_4 are defined as follows,

$$C_1 = \{\mathbf{z} : (z_1 - c_1)^2 + (z_2 - c_2)^2 + (z_3 - c_3)^2 \geq r^2\},$$
$$r^2 = 0.65,\ \mathbf{c} = (0.33, 0.33, 0.33),$$

$$C_2 = \{\mathbf{z} : (z_1 - c_1)^2 + (z_2 - c_2)^2 + (z_3 - c_3)^2 \leq r^2\},$$
$$r^2 = 0.15,\ \mathbf{c} = (0.53, 0.23, 0.8),$$

$$C_3 = \{\mathbf{z} : (z_1 - c_1)^2 + (z_2 - c_2)^2 + (z_3 - c_3)^2 \leq r^2\},$$
$$r^2 = 0.1,\ \mathbf{c} = (0.23, 0.53, 0.8),$$

and,

$$C_4 = C_a \cap C_b,$$
$$C_a = \{\mathbf{z} : (z_1 - c_1)^2 + (z_2 - c_2)^2 + (z_3 - c_3)^2 \geq r_a^2\},$$
$$C_b = \{\mathbf{z} : (z_1 - c_1)^2 + (z_2 - c_2)^2 + (z_3 - c_3)^2 \leq r_b^2\},$$
$$r_a^2 = 0.2,\ r_b^2 = 0.27,\ \mathbf{c} = (0.63, 0.63, 0.38).$$

Subsequently (7) was solved to obtain the weighting vectors corresponding to these regions and using these weighting vectors MACE-*g*D was able to generate a PF that closely resembles the initially chosen regions, see *Fig. (11*). This concept extends directly to MAPs, however the results are much more difficult to visualise.

Additionally, although it is useful to know the geometry of the PF, it is sufficient if its general shape is known. The boundary for which the weighting vectors radically change position is the transition from concave geometry to convex geometry, see *Fig. (1*) – *Fig. (3*).

IX. Conclusion

A new concept was introduced and used in the solution of many-objective optimization problems (MAPs), namely generalized decomposition (*g*D). With the aid of *g*D, weighting vectors can be selected optimally to satisfy specific requirements in the distribution of the Pareto optimal solutions along the PF. This approach allows decomposition-based MOEAs to focus on only one performance objective, that of convergence to the PF. This can be a significant advantage over other MOEAs that have to tackle 3 performance objectives simultaneously, i.e. Pareto front coverage, even distribution of Pareto optimal solutions and convergence to the Pareto front. Based on *g*D and the CE method, a many-objective optimization framework was presented, MACE-*g*D. The performance of MACE-*g*D with respect to the GD-metric is competitive with that of MOEA/D and RM-MEDA, for the selected problem set. Additionally, a methodology for incorporating DM preferences is given, using the presented framework. As far as we are aware, there is no other method available that can address all of the aforementioned issues so successfully. Another benefit of *g*D-based algorithms is that since there is a clear way to distribute solutions on the Pareto front very precisely, the necessity of using elaborate archiving strategies and sharing is diminished. However, these benefits require that certain prior information is available. Specifically, the geometry of the Pareto front needs to be known *a priori*. This requirement can be alleviated to a certain degree, however, by adaptively identifying the shape of the PF of a problem during the optimization process. This adaptive Pareto front identification for use with *g*D seems to be a promising direction for future research.

Another result of this study is that the CE method appears to be a strong candidate as the main algorithm of choice for multiobjective optimization. This is fortunate since the CE method is based on sound theoretical principles which can facilitate further analysis of this method. Also, the hypothesis presented in [36], that EDAs based on low order statistics and clustering can be used as an alternative to complex probabilistic models, seems to be supported by the obtained results in Section VII-C. However, as no clustering method is

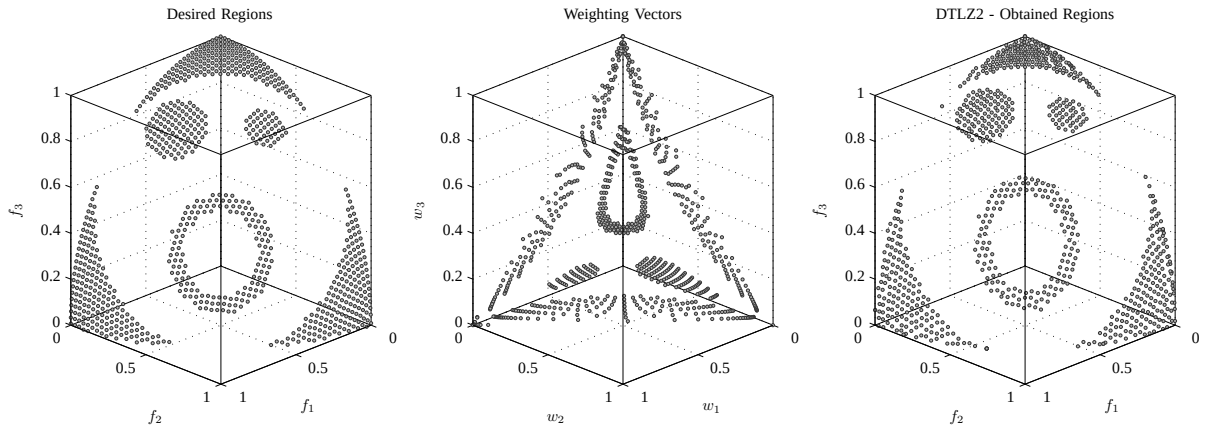

Fig. 11. Left: Preferred regions of the Pareto front. Middle: Weighting vectors corresponding to the preferred PF regions. Right: Obtained Pareto optimal solutions on a 3-objective instance of the DTLZ2.

employed in MACE-*g*D, this does not constitute solid proof but it is certainly a good indication.

In conclusion, it was shown that MACE-*g*D is a scalable framework for tackling many-objective problems, for example see *Fig. (7*), with respect to the GD-metric. Also, MACE-*g*D seems to be robust with respect to its main control parameter, ρ, see Section VII-D. Furthermore, the collective results of this work strongly suggest that the choice of weighting vectors in MOEAs based on decomposition can affect not only the distribution of Pareto optimal solutions on the PF but also the convergence of the algorithm. This issue is more evident in many-objective problems. Restriction of the search in objective space to a region that is of interest can be an effective approach in MAPs. Otherwise, the necessary increase in population size to obtain similar coverage in many-objectives as for 2 or 3-objective problems is computationally intractable. This restricted search is fully supported by the presented framework.

APPENDIX A
GENERATING AN N-DIMENSIONAL UNIFORMLY DISTRIBUTED CONCAVE OR CONVEX PARETO FRONT

A moderately efficient and highly convenient method for generating uniformly distributed points on the unit hypersphere of arbitrary dimension is presented by Marsaglia [62]. Let n be the dimension of a unit hypersphere. Then the method to uniformly distribute points on its' *surface* can be summarized as follows:

- Generate $\mathcal{X}_1, \mathcal{X}_2, \ldots, \mathcal{X}_n$ independent random deviates distributed according to $\mathcal{N}(0,1)$. $\mathcal{N}(0,1)$ is the normal distribution with mean 0 and variance 1.
- Calculate $S = \mathcal{X}_1^2 + \mathcal{X}_2^2 + \cdots + \mathcal{X}_n^2$, the point defined as:

$$U = \left(\frac{\mathcal{X}_1}{\sqrt{S}}, \frac{\mathcal{X}_2}{\sqrt{S}}, \ldots, \frac{\mathcal{X}_n}{\sqrt{S}} \right) \tag{33}$$

  is uniformly distributed on the n-dimensional hypersphere [62].

With this method, points on the unit hypersphere can be sampled that are uniformly distributed; however these points are not Pareto optimal. To obtain a concave Pareto front with uniformly distributed points all that is necessary is to select the points that all their components are non-negative. That is all points U for which the following is true $U \in \mathbb{R}^n_+$. Conversely to obtain a Pareto front with convex geometry , it is sufficient to select all the generated points $U \in \mathbb{R}^n_-$.

However there is a limitation to the described method. Namely since only a subset of the generated solutions U is used, for higher dimensions in order to obtain the same number of Pareto optimal points it is required that the number of uniformly distributed solutions in U is constantly increased. The required number of points in U, such that a specific number of Pareto optimal solutions is obtained, can be derived from the following relation that follows directly from geometric considerations,

$$|\mathcal{P}| \approx \frac{1}{2^k}|U|, \tag{34}$$

where $\approx$ becomes an equality in the limit as $|U| \to \infty$. For example, to obtain approximately 100 uniformly distributed solutions for a concave PF in 11 dimensions, then it is required that $204\,800$ uniformly distributed vectors U are generated on the 11 dimensional unit hypersphere. This translates to $\sim 2.2 \times 10^6$ samples from the normal distribution $\mathcal{N}(0,1)$. So this method can easily become impractical for higher dimensions.

APPENDIX B
CONVEX SETS AND FUNCTIONS

Some fundamental definitions about convex sets and functions are given below. For further details the reader is referred to [42] for an applications oriented presentation and [63] for a more theoretical approach.

A. Convex Sets

A set $C \subseteq \mathbb{R}^n$ is *convex* if for any $\mathbf{x}, \mathbf{y} \in C$ and any $\theta \in [0,1]$,

$$\theta \mathbf{x} + (1-\theta)\mathbf{y} \in C. \tag{35}$$

The combination of the points $\mathbf{x}, \mathbf{y}$ in (35), is called a *convex combination* and can be extended to multiple points in a

similar manner to the extension of affine combinations,

$$\sum_{i=1}^{d}\theta_i\mathbf{x}_i, \text{ with } \sum_{i=1}^{d}\theta_i = 1, \text{ and } \theta_i \geq 0, \text{ for all } i = 1,\ldots,d. \tag{36}$$

The set of all convex combinations of a convex set C is the *convex hull* of that set and is denoted as,

$$\mathbf{conv}\, C = \left\{\sum_{i=1}^{d}\theta_i\mathbf{x}_i : \mathbf{x}_i \in C,\ \sum_{i=1}^{d}\theta_i = 1, \theta_i \geq 0\right\}, \tag{37}$$

for $i = 1,\ldots,d$.

A function $f : \mathbb{R}^n \to \mathbb{R}$ is said to be convex if the domain of definition of f, denoted as $\mathbf{dom}\, f$, is a convex set and $\forall\, \mathbf{x}, \mathbf{y} \in \mathbf{dom}\, f$ and $\theta \in [0,1]$ we have,

$$f(\theta\mathbf{x} + (1-\theta)\mathbf{y}) \leq \theta f(\mathbf{x}) + (1-\theta)f(\mathbf{y}). \tag{38}$$

A function is strictly convex if the inequality in (38) is strict. Accordingly a function is concave if $-f$ is convex. A more interesting definition of convex and concave functions is formulated with the aid of the *epigraph* of a function, see Appendix B-B.

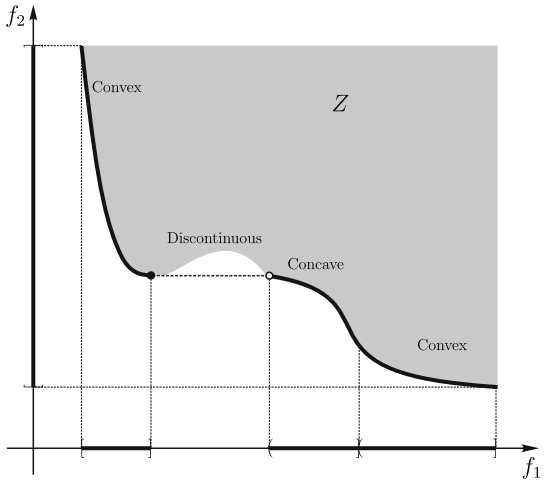

Fig. 12. A Pareto front which is partially convex, partially concave and discontinuous. Notice that the frame of reference, which in this case is f_1, used to determine the convex and concave parts is arbitrary, namely the same parts of the Pareto front would be partially convex and concave, even if f_2 was chosen as the reference. However, discontinuities on the PF are not always *visible* from all frames of reference, i.e. the projection of the PF on the f_2 axis is continuous, while the projection on the f_1 axis is discontinuous.

B. Epigraph

The *epigraph* of a function $f : \mathbb{R}^n \to \mathbb{R}$, which is the Greek word for *above the graph*, is defined as

$$\mathbf{epi}\, f = \{(\mathbf{x}, t) : \mathbf{x} \in \mathbf{dom}\, f, t \in \mathbb{R}, f(\mathbf{x}) \leq t\}, \tag{39}$$

consequently $\mathbf{epi}\, f \subset \mathbb{R}^{n+1}$. If the epigraph of a function is a convex set then the function is convex and vice versa. The *hypograph* of a function $f : \mathbb{R}^n \to \mathbb{R}$, meaning *below the graph*, is defined as,

$$\mathbf{hypo}\, f = \{(\mathbf{x}, t) : \mathbf{x} \in \mathbf{dom}\, f, t \in \mathbb{R}, f(\mathbf{x}) \geq t\}. \tag{40}$$

If a function is concave, its hypograph is a convex set. In general a function $f : \mathbb{R}^n \to \mathbb{R}$ with a convex domain of definition is:

- Convex, if and only if $\mathbf{epi}\, f$ is a convex set. If in addition $\mathbf{hypo}\, f$ is nonconvex then, f is strictly convex.
- Concave, if and only if $\mathbf{hypo}\, f$ is a convex set. If in addition $\mathbf{epi}\, f$ is nonconvex then, f is strictly concave.
- Convex and concave, if both $\mathbf{epi}\, f$ and $\mathbf{hypo}\, f$ are convex. A concave and convex function is affine.
- Nonconvex, if both $\mathbf{epi}\, f$ and $\mathbf{hypo}\, f$ are nonconvex.

C. Pareto Front Geometry

Assuming that the Pareto front can be represented by a piecewise continuous function, $g : \mathbb{R}^{k-1} \to \mathbb{R}$ and k the number of objectives, then there are three types of *geometries* and combinations thereof, that the PF can have. Namely the function, g, can have parts that are convex, concave, of affine. We refer to a Pareto front as,

- Convex, if $\mathbf{epi}\, g$ is a convex set.
- Concave, if $\mathbf{hypo}\, g$ is a convex set.
- Affine, if both $\mathbf{epi}\, g$ and $\mathbf{hypo}\, g$ are convex.
- Discontinuous, if $\mathbf{dom}\, g$ is nonconvex or g is discontinuous.
- Partially convex, if g is convex over a convex subset of $\mathbf{dom}\, g$.
- Partially concave, if g is concave over a convex subset of $\mathbf{dom}\, g$.
- Partially affine, if g is convex and concave over a convex subset of $\mathbf{dom}\, g$.
- Piecewise convex, if g partially convex over all convex subsets of $\mathbf{dom}\, g$.
- Piecewise concave, if g partially concave over all convex subsets of $\mathbf{dom}\, g$.
- Piecewise affine, if g partially affine over all convex subsets of $\mathbf{dom}\, g$.

ACKNOWLEDGMENT

The authors would like thank Jacob Mattingley for providing access to his tool CVXGEN [64]. In this work CVXGEN is employed to solve (7). The authors also gratefully acknowledge Ricardo H.C. Takahashi for useful discussions and for 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] P. Fleming and R. Purshouse, "Evolutionary Algorithms in Control Systems Engineering: A Survey," *Control Engineering Practice*, vol. 10, no. 11, pp. 1223–1241, 2002.

[2] M. Tapia and C. Coello, "Applications of Multi-Objective Evolutionary Algorithms in Economics and Finance: A Survey," in *IEEE Congress on Evolutionary Computation*, vol. 2007, 2007, pp. 532–539.

[3] N. Krasnogor, W. Hart, J. Smith, and D. Pelta, "Protein Structure Prediction with Evolutionary Algorithms," in *Proceedings of the Genetic and Evolutionary Computation Conference*, vol. 2, 1999, pp. 1596–1601.

[4] R. C. Purshouse and P. J. Fleming, "Conflict, Harmony, and Independence: Relationships in Evolutionary Multi-Criterion Optimisation," in *Conference on Evolutionary Multi-Criterion Optimization*. Berlin: Springer, 2003, pp. 16–30.

[5] D. Goldberg and J. Holland, "Genetic Algorithms and Machine Learning," *Machine Learning*, vol. 3, no. 2, pp. 95–99, 1988.

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

[7] 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.

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

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

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

[11] R. Purshouse and P. Fleming, "Evolutionary Many-Objective Optimisation: An Exploratory Analysis," in *IEEE Congress on Evolutionary Computation*, vol. 3. IEEE, 2003, pp. 2066–2073.

[12] 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.

[13] R. Takahashi, R. Saldanha, W. Dias-Filho, and J. Ramírez, "A New Constrained Ellipsoidal Algorithm for Nonlinear Optimization with Equality Constraints," *IEEE Transactions on Magnetics*, vol. 39, no. 3, pp. 1289–1292, 2003.

[14] A. Jaszkiewicz, "On the Performance of Multiple-Objective Genetic Local Search on the 0/1 Knapsack Problem - A comparative Experiment," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 4, pp. 402–412, 2002.

[15] E. Hughes, "Multiple Single Objective Pareto Sampling," in *Congress on Evolutionary Computation, 2003*, vol. 4. IEEE, 2003, pp. 2678–2684.

[16] ——, "MSOPS-II: A general-purpose Many-Objective optimiser," in *IEEE Congress on Evolutionary Computation*, sept. 2007, pp. 3944 – 3951.

[17] Y. yan Tan, Y. chang Jiao, H. Li, and X. kuan Wang, "MOEA/D + uniform design: A new version of MOEA/D for optimization problems with many objectives," *Computers & Operations Research*, no. 0, 2012.

[18] S. Jiang, Z. Cai, J. Zhang, and Y.-S. Ong, "Multiobjective Optimization by Decomposition with Pareto-Adaptive Weight Vectors," in *International Conference on Natural Computation*, vol. 3, july 2011, pp. 1260 –1264.

[19] S. Jiang, J. Zhang, and Y. Ong, "Asymmetric Pareto-adaptive Scheme for Multiobjective Optimization," in *Advances in Artificial Intelligence*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2011, vol. 7106, pp. 351–360.

[20] 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.

[21] L. While, P. Hingston, L. Barone, and S. Huband, "A Faster Algorithm for Calculating Hypervolume," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 1, pp. 29 – 38, feb. 2006.

[22] C. Fonseca, L. Paquete, and M. Lopez-Ibanez, "An Improved Dimension-Sweep Algorithm for the Hypervolume Indicator," in *IEEE Congress on Evolutionary Computation*, july 2006, pp. 1157 – 1163.

[23] F. Gu, H. Liu, and K. Tan, "A Multiobjective Evolutionary Algorithm Using Dynamic Weight Method," *International Journal of innovative Computing, Information and Control*, vol. 8, no. 5B, pp. 3677–3688, may 2012.

[24] J. Bader and E. Zitzler, "HypE: An Algorithm for Fast Hypervolume-Based Many-Objective Optimization," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 1, pp. 45–76, 2011.

[25] J. Holland, "Adaptation in natural and artificial systems," 1975.

[26] H. Schwefel, "Evolutionsstrategie und Numerische Optimierung," Ph.D. dissertation, Technische Universität Berlin, 1975.

[27] R. Eberhart and J. Kennedy, "A New Optimizer Using Particle Swarm Theory," in *International Symposium on Micro Machine and Human Science*. IEEE, 1995, pp. 39–43.

[28] 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.

[29] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable Multi-Objective Optimization Test Problems," in *Congress on Evolutionary Computation*, vol. 1, may 2002, pp. 825 –830.

[30] E. Zitzler, K. Deb, and L. Thiele, "Comparison of Multiobjective Evolutionary Algorithms: Empirical Results," *Evolutionary Computation*, vol. 8, no. 2, pp. 173–195, 2000.

[31] H. Mühlenbein and G. Paass, "From Recombination of Genes to the Estimation of Distributions I. Binary Parameters," *Parallel Problem Solving from Nature*, pp. 178–187, 1996.

[32] J. He and X. Yao, "Drift Analysis and Average Time Complexity of Evolutionary Algorithms," *Artificial Intelligence*, vol. 127, no. 1, pp. 57–85, 2001.

[33] T. Chen, K. Tang, G. Chen, and X. Yao, "Analysis of Computational Time of Simple Estimation of Distribution Algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 14, no. 1, pp. 1–22, 2010.

[34] M. Hauschild and M. Pelikan, "A survey of estimation of distribution algorithms," 2011.

[35] M. Pelikan, "Bayesian Optimization Algorithm," *Hierarchical Bayesian Optimization Algorithm*, pp. 31–48, 2005.

[36] L. Emmendorfer and A. Pozo, "Effective Linkage Learning Using Low-Order Statistics and Clustering," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 6, pp. 1233–1246, 2009.

[37] C. Echegoyen, Q. Zhang, A. Mendiburu, R. Santana, and J. Lozano, "On the Limits of Effectiveness in Estimation of Distribution Algorithms," in *IEEE Congress on Evolutionary Computation*. IEEE, 2011, pp. 1573–1580.

[38] R. Rubinstein, "The Cross-Entropy Method for Combinatorial and Continuous Optimization," *Methodology and Computing in Applied Probability*, vol. 1, no. 2, pp. 127–190, 1999.

[39] R. Takahashi, E. Carrano, and E. Wanner, "On a Stochastic Differential Equation Approach for Multiobjective Optimization up to Pareto-Criticality," in *Evolutionary Multi-Criterion Optimization*, ser. Lecture Notes in Computer Science. Springer Berlin, 2011, vol. 6576, pp. 61–75.

[40] K. Miettinen and M. Mäkelä, "On Scalarizing Functions in Multiobjective Optimization," *OR Spectrum*, vol. 24, no. 2, pp. 193–213, 2002.

[41] M. Grant, S. Boyd, and Y. Ye, "Disciplined Convex Programming," vol. 84, pp. 155–210, 2006.

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

[43] M. Grant and S. Boyd, "CVX: Matlab Software for Disciplined Convex Programming," 2008. [Online]. Available: http://cvxr.com/cvx/

[44] E. Saff and A. Kuijlaars, "Distributing Many Points on a Sphere," *The Mathematical Intelligencer*, vol. 19, no. 1, pp. 5–11, 1997.

[45] S. B. Damelin and P. J. Grabner, "Energy Functionals, Numerical Integration and Asymptotic Equidistribution on the Sphere," *Journal of Complexity*, vol. 19, no. 3, pp. 231 – 246, 2003.

[46] D. Wolpert, "Information Theory - The Bridge Connecting Bounded Rational Game Theory and Statistical Physics," *Complex Engineered Systems*, pp. 262–290, 2006.

[47] R. Rubinstein, "A Stochastic Minimum Cross-Entropy Method for Combinatorial Optimization and Rare-event Estimation," *Methodology and Computing in Applied Probability*, vol. 7, no. 1, pp. 5–50, 2005.

[48] Z. Botev, D. Kroese, and T. Taimre, "Generalized Cross-Entropy Methods with Applications to Rare-Event Simulation and Optimization," *Simulation*, vol. 83, no. 11, p. 785, 2007.

[49] P. De Boer, D. Kroese, S. Mannor, and R. Rubinstein, "A Tutorial on the Cross-Entropy Method," *Annals of Operations Research*, vol. 134, no. 1, pp. 19–67, 2005.

[50] C. N. Morris, "Natural Exponential Families with Quadratic Variance Functions," *The Annals of Statistics*, vol. 10, pp. 65–80, 1982.

[51] H. Ishibuchi, Y. Sakane, N. Tsukamoto, and Y. Nojima, "Effects of Using Two Neighborhood Structures on the Performance of Cellular Evolutionary Algorithms for Many-Objective Optimization," in *IEEE Congress on Evolutionary Computation*, may 2009, pp. 2508 –2515.

[52] H. Li and Q. Zhang, "Multiobjective Optimization Problems with Complicated Pareto Sets, MOEA/D and NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 2, pp. 284–302, 2009.

[53] A. Zhou, Q. Zhang, and G. Zhang, "A Multiobjective Evolutionary Algorithm Based on Decomposition and Probability Model," in *IEEE Congress on Evolutionary Computation*, june 2012, pp. 1–8.

[54] I. Das and J. Dennis, "Normal-Boundary Intersection: An Alternate Method for Generating Pareto Optimal Points in Multicriteria Optimization Problems," DTIC Document, Tech. Rep., 1996.

[55] 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.

[56] K. Deb, A. Sinha, and S. Kukkonen, "Multi-Objective Test Problems, Linkages, and Evolutionary Methodologies," in *Conference on Genetic and Evolutionary Computation*. ACM, 2006, pp. 1141–1148.

[57] S. Kukkonen and J. Lampinen, "GDE3: The Third Evolution Step of Generalized Differential Evolution," in *IEEE Congress on Evolutionary Computation*, vol. 1. IEEE, 2005, pp. 443–450.

[58] P. Bosman and D. Thierens, "The Naive MIDEA: A Baseline Multi-Objective EA," in *Evolutionary Multi-Criterion Optimization*. Springer, 2005, pp. 428–442.
[59] D. Wolpert and W. Macready, "No Free Lunch Theorems for Optimization," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 67–82, 1997.
[60] D. Van Veldhuizen, "Multiobjective Evolutionary Algorithms: Classifications, Analyses, and New Innovations," in *Evolutionary Computation*, 1999.
[61] 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.
[62] G. Marsaglia, "Choosing a Point from the Surface of a Sphere," *The Annals of Mathematical Statistics*, vol. 43, pp. 645–646, 1972.
[63] R. Rockafellar, *Convex Analysis*. Princeton University Press, 1970, vol. 28.
[64] J. Mattingley and S. Boyd, "CVXGEN: A Code Generator for Embedded Convex Optimization," *Optimization and Engineering*, pp. 1–27, 2012.

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.

Robin Purshouse received the MEng degree in Control Systems Engineering in 1999 and a Ph.D. in Control Systems in 2004 for his research on evolutionary many-objective optimisation. Commercial experience includes Logica plc (1999-2000), PA Consulting Group (2003-2007) and Rolls-Royce plc (2007-2008). He returned to academia in 2008 as a Research Fellow in the School of Health and Related Research at the University of Sheffield and was appointed as Lecturer in the Department of Automatic Control and Systems Engineering in 2010.

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.