Information Sciences 282 (2014) 363–387

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Generalized decomposition and cross entropy methods for many-objective optimization

I. Giagkiozis [a,b,*], R.C. Purshouse [b], P.J. Fleming [b]

[a] School of Mathematics and Statistics (SoMaS), University of Sheffield, Sheffield S3 7RH, UK
[b] Department of Automatic Control and Systems Engineering, University of Sheffield, Sheffield S1 3JD, UK

ARTICLE INFO

Article history:
Received 29 November 2012
Received in revised form 11 April 2014
Accepted 20 May 2014
Available online 8 June 2014

Keywords:
Generalized decomposition
Many-objective optimization
Decomposition method
Cross entropy method
Convex optimization

ABSTRACT

Decomposition-based algorithms for multi-objective optimization problems have increased in popularity in the past decade. Although convergence to the Pareto optimal front (PF) for such algorithms can often be superior to that of Pareto-based alternatives, the problem of effectively distributing Pareto optimal solutions in a high-dimensional space has not been solved. 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 search algorithm toward specific regions of interest, or the entire Pareto front, with the desired distribution of Pareto optimal solutions. The method simplifies many-objective problems by unifying the three performance objectives of an *a posteriori* multi-objective optimizer – 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, is created to illustrate the benefits of the proposed concept for many-objective problems. The algorithm – MACE-gD – is shown to be highly competitive with the existing best-in-class decomposition-based algorithm (MOEA/D) and a more elaborate EDA method (RM-MEDA).

1. 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:

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

where k is the number of objective functions and $\mathbf{x}$ is the vector of decision variables defined in a feasible domain $\mathbf{S} \subseteq \mathbb{R}^n$. In the event that there is *conflict* between the objectives, such that improved performance in one objective can only be obtained at the expense of reduced performance in another objective, then Eq. (1) admits no single *optimal* solution; rather a family of

* Corresponding author at: Department of Automatic Control and Systems Engineering, University of Sheffield, Sheffield S1 3JD, UK. Tel.: +44 01142225616.
E-mail addresses: i.giagkiozis@sheffield.ac.uk (I. Giagkiozis), r.purshouse@sheffield.ac.uk (R.C. Purshouse), p.fleming@sheffield.ac.uk (P.J. Fleming).

http://dx.doi.org/10.1016/j.ins.2014.05.045
0020-0255/

Pareto optimal solutions exists, representing different performance trade-offs for the problem at hand. Given that we would like to reveal this set of trade-offs to the decision maker (DM) then the task of the optimizer is to find a set of solutions that represent this *Pareto front* (PF) in objective-space. This type of optimization is generally referred to as *a posteriori*, since the DM applies his or her preferences for a particular trade-off between objectives *after* the full set of trade-offs have been revealed [4, pp. 63].

MOPs for 2 or 3 objectives have been heavily studied in the literature and effective optimizers are available for these types of problems – for example [5]. However it is now known that the Pareto-based algorithms designed for these types of problems experience a failure mode for problems with 4 or more objectives [6]. These types of problems are typically referred to as *many-objective problems* (MAPs). For brevity, hereafter we refer to multi and many-objective problems simply as MAPs.

Evolutionary algorithms (EAs) have long been regarded as a suitable choice of method for the *a posteriori* optimization of MAPs [7]. EAs maintain a family of solutions during the optimization process and therefore have the potential to maintain a representative set of trade-off solutions simultaneously, with the potential to exploit the synergies of a parallel search across all possible trade-offs. The algorithms that have been designed with this purpose in mind are known as *multi-objective evolutionary algorithms* (MOEAs). 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, see [8]. However all MOEAs require the performance of a solution to be represented as a scalar fitness value, upon which the MOEAs can base their decision as to the direction of search. This is a very well known problem in MAPs and has been investigated by many researchers over the past three decades – seminal examples include [7,9,10]. There are two major classes of approaches for resolving this issue: Pareto-based and decomposition-based methods.

Pareto-based methods use the Pareto-dominance relations [4], to induce partial ordering in the objective space. These relations, were initially introduced by Edgeworth [11] and later expanded by Pareto [12]. 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 ($\leqslant$) 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 [4]. However such relations are of limited utility when the number of dimensions[1] is increased [6]. 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 will tend to be non-dominated [13]. Hence this type of partial ordering appears to be 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 function 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 [4,14]. However multi-objective evolutionary algorithms have also employed decomposition, for example [15–17]. A central issue in decomposition-based methods is how to select a set of weighting vectors that will provide a *well* distributed set of Pareto optimal points. A popular assumption is that an even distribution of weighting vectors will result in *well* distributed Pareto optimal points [10]. However, with the help of a novel concept which we call *generalized* decomposition [18], we will demonstrate that this assumption is flawed and provide an exact solution to this issue, subject to having some prior information about the problem.

This problem with decomposition methods has motivated researchers to employ adaptive approaches for the selection of weighting vectors in decomposition-based algorithms. An interesting adaptive method to select the set of weighting vectors is presented in [19,20]. 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 [21]. Subsequently, the points found in the previous step, are used to identify 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 $p_i \in \mathbb{R}_{++}$,[2] and the fact that Eq. (2) equals to one means that the objective functions are normalized in the range, $[0, 1]$. However the problem of solving for the p_i parameters in Eq. (2) is nonconvex. Nevertheless in [19,20] this issue was not addressed and the Newton method was used. The Newton method however can only perform local search thus will be unable to identify the correct p_i parameters. The effects of this difficulty are seen in [20] 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 [19,20]. The second problem is that the weighting vectors that correspond to points on the identified Pareto front are formulated in a similar fashion to Eq. (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 [21], which is used to ascertain the quality of the *reference* points on the PF, has exponential complexity in the number of objectives [22,23], which limits the method to approximately 4-objective problems, since the

[1] Unless stated otherwise with: *number of dimensions* we refer to the number of scalar objective functions, k.

[2] $\mathbb{R}_+$ refers to the set of non-negative real numbers and $\mathbb{R}_{++}$ to the set of positive real numbers.

hypervolume must be calculated several times on every iteration of the algorithm [20]. Another interesting method is due to Gu et al. [24] and others. Although these methods appear to be promising there is evidence that adaptive schemes for the selection of the weighting vectors convert the optimization problem to a varying one which can have a potentially high impact on the convergence rate of the algorithm [25].

Despite the successes of MOEAs, particularly on problems with 2 or 3 objectives, their stochastic nature does present certain difficulties. For example, it is very hard to analyse the behavior 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 [26–28]. 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*[3] 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 a sample [29]. 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 [30,31]. However EDAs are not a panacea since they heavily depend on the quality and complexity of the underlying probabilistic model [32]. 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 [32], 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 [33]. Acknowledging this issue has led some researchers to suggest hybridization of EDAs based on simple probabilistic models with some form of clustering [34]. This course is further supported by more recent studies [35].

For these reasons we have selected an EDA, the so-called Cross Entropy method (CE), as the main optimization algorithm in our generalized decomposition-based framework. CE was introduced by Rubinstein [36], initially as a rare event estimation technique and subsequently as an algorithm for combinatorial and continuous optimization problems. The most attractive 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 remainder of this paper is structured as follows. In Section 2 generalized decomposition is described along with the benefits that this method can bring to currently existing MOEAs. Following this, in Section 3 the EDA employed in our framework, the CE-method, is presented along with its form for continuous optimization problems. A many-objective optimization framework based on generalized decomposition and CE is described in Section 4. The algorithms in our comparative studies in Section 6 are described in Section 5. In Section 7 we illustrate how generalized decomposition can be used for preference articulation. Lastly in Section 8 we summarize and conclude this work.

2. Generalized decomposition

Generalized decomposition (gD) was first introduced in [18], as a way to optimally select the weighting vectors in decomposition-based algorithms, subject to the Pareto front geometry being known *a priori*. In this work we show that, even if this requirement is not fulfilled, the performance of gD can still be orders of magnitude better with regard to the quality of distribution of Pareto optimal points as measured by the Riesz kernel [37], when compared with two highly regarded methods.

2.1. Decomposition methods

Decomposition methods, have been employed in several MOEAs, for example [15–17]. These methods transform Eq. (1) to a set of single-objective subproblems to be solved simultaneously with the help of a scalarizing function and a set of weighting vectors. The potential of such methods for extending MOEAs to MOPs is obvious considering that the basis of almost every, if not all, optimization algorithms is a method that can address only single objective problems.

Arguably the simplest scalarizing function is the weighted sum method [38]:

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

where $\mathbf{w} = (w_1,\ldots,w_k)$. However it has been shown that for complicated Pareto fronts, an algorithm based on Eq. (3) is unable to discover all Pareto optimal solutions [4]. In [39] further insight is given as to the reasons for this behavior. Although, with some modifications this simple decomposition can produce respectable results, for example see [10].

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

A more sophisticated decomposition is based on the weighted metrics method [38]:

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

here as in Eq. (3), it is assumed that $w_i \geqslant 0$ 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. For this decomposition 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 Eq. (5) is $\tilde{\mathbf{x}}$ [4]. A convex weighting vector, $\mathbf{w}$ is 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 ith component that is equal to one. Also $\mathbf{conv}\ C$ is the *convex hull* of the set C which is defined in Eq. (A.3). For further details see Appendix A. 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.

2.2. Optimal choice of weighting vectors – Generalized decomposition

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. Jaszkiewicz [15] suggested that a uniformly sampled set of weighting vectors $\mathbf{w}$ would produce a uniformly distributed of Pareto optimal solutions along the entire PF. Later Zhang et al. [10] proposed using a set of evenly spaced weighting vectors to produce *well* distributed Pareto optimal solutions on the basis that the various subproblems obtained using different weighting vectors are a continuous function of the weights. Whilst this seems to be the case, there is however nothing to suggest, critically, that this *continuous* function is also linear in the parameters $\mathbf{w}$. In fact 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 not the case. The parameter N in Eq. (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 sth subproblem.

This calculation was performed with what we call *generalized decomposition* [18], which is given by the solution of the program in Eq. (7). The insight in this formulation is that by using Eq. (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 $\|\tilde{\mathbf{w}} \circ \mathbf{F}(\tilde{\mathbf{x}})\|_\infty \leqslant \|\mathbf{w} \circ \mathbf{F}(\tilde{\mathbf{x}})\|_\infty$ would be true 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 $\tilde{\mathbf{w}}$. Here, $\mathcal{W}$ 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 for optimization purposes as we will show. To obtain the $\tilde{\mathbf{w}}$ vectors, the following program is to be solved for every Pareto optimal point of interest:

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

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 Eq. (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 Eq. (7) is a disciplined convex program [40], hence it is also a convex program. So a unique solution is guaranteed and can be obtained by solving Eq. (7) using any method presented in [41] for solving convex problems, for example, an interior-point method. 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:

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

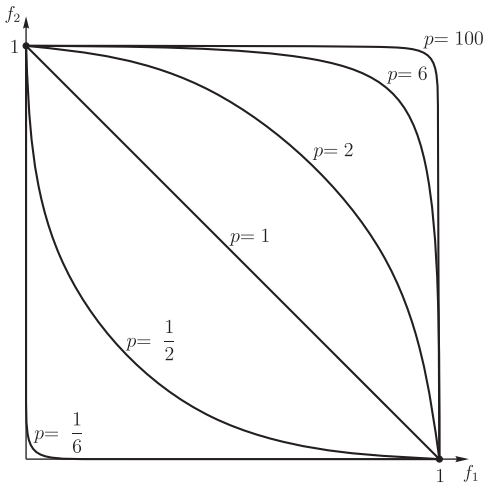

Fig. 1. Various Pareto front geometries that satisfy the following equation: $f_1^p + f_2^p = 1$ for $p = \{100, 6, 2, 1, \frac{1}{2}, \frac{1}{6}\}$.

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 [40–42].

In concluding this section, we hypothesise that the choice of weighting vectors will greatly influence the convergence and spread of the resulting Pareto front. However this choice has, to date, been either arbitrary or based on invalid assumptions.

2.3. The effect of weighting vector choice

Assuming that our definition of *well* distributed PF solutions is a Pareto optimal set evenly distributed along the trade-off surface, the following experiment illustrates the benefits of using generalized decomposition (whilst noting that the generalized decomposition framework is fully flexible for accommodating any other definition of well distributed Pareto optimal solutions).

In the experiment we define a number of Pareto front geometries and desired resolutions for representing the trade-off surface. For each of these configurations we use generalized decomposition to generate a set of weight vectors which corresponds to a uniform distribution across the Pareto front at the desired resolution. We then compare the results to both a uniform and even distribution of weight vectors via a commonly used measure of evenly distributed points on the unit hypersphere – the Coulomb potential [43], or Riesz kernel [37], defined as:

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

and for $s = 2$, Eq. (9) is equivalent, up to a multiplicative constant, to the Coulomb potential energy. In this work we refer to Eq. (9) as the s-energy metric or simply s-energy. The set $\mathbf{Z}$ in the present work is the set of objective vectors $\mathbf{z}$. Intuitively, when Eq. (9) is minimized then the mean nearest neighbor distance of the set of points $\mathbf{z}$ is maximized whilst the variance of this distance is minimized. This of course is subject to the constraints imposed by the geometry of the PF. For some examples on the distribution of solutions using Eq. (9) the reader is referred to [43]. It has been shown that for a k-dimensional manifold the s-energy with $s \geqslant k$ is minimized when points on that manifold are evenly distributed [44]. Therefore, since the Pareto front of a k-objective problem is at most a $(k-1)$-dimensional manifold [10], the s parameter in the s-energy metric used for the following experiments is set to $k-1$.

We have assumed a Pareto front geometry that can be described by:

$$\begin{aligned} &f_1^p + f_1^p + , \ldots , f_k^p = 1^p \\ &f_i \geqslant 0, \quad \text{for all } i \in \{1, \ldots, k\} \ \text{ and } p > 0. \end{aligned} \tag{10}$$

Note that Eq. (10) is the positive orthant of a generalized hypersphere which is a good approximation[4] for a wide range of PF geometries in benchmark [27,26,45] and real world problems [46–48].

For 2 to 11 dimensions and for a set of Pareto front geometries $p = \{100, 6, 2, 1, \frac{1}{2}, \frac{1}{6}\}$, see Fig. 1, N number of objective vectors are selected according to generalized decomposition and the methods described in [15,10]. The number of selected objective vectors used in every instance can be seen in Table 1. 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

[4] We assume the problems are normalized.

Table 1
The number of objective vectors, N, for constant H used in the experiment seen in Fig. 2.

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

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 PF geometry,[5] a set of weighting vectors is generated according to uniform and even distribution schemes. For generalized decomposition, the weighting vectors are generated using a reference Pareto front with the desired distribution. However in real-world optimization problems the PF geometry will, in general, be unknown, so we have assumed an affine geometry for the reference PF and this is used for all problem instances. For example, in 2 dimensions, if the PF is the first quadrant of a unit circle (see Fig. 1 for $p = 2$), we use the 1-simplex evenly sampled and then the *optimal*[6] weighting vectors are estimated by solving Eq. (7). In general, we use an evenly distributed sample on the $(k-1)$-simplex.

Subsequently, using the inverse relationship to Eq. (7), namely:

$$\begin{aligned} &\min_{\mathbf{F}(\mathbf{x})} \|\mathbf{F}(\mathbf{x}) \circ \tilde{\mathbf{w}}\|_\infty, \\ &\text{subject to } \sum_{i=1}^{k} f_i = 1, \\ &\text{and } f_i \geqslant 0, \quad \forall\, i \in \{1,\dots,k\}. \end{aligned} \tag{11}$$

the Pareto optimal solutions $\mathbf{F}(\mathbf{x})$ that minimize every subproblem $\tilde{\mathbf{w}}$ are calculated. However, as can be seen in Eq. (11), the inverse problem to Eq. (7) can be solved analytically only for an affine Pareto front. Although, in the case of a concave or convex PF, the affine PF obtained by Eq. (11) can be projected onto the generalized unit hypersphere (Eq. (10)) and the obtained solutions will still be optimal for their corresponding weighting vectors.

Lastly, the log_{10} of the s-energy of obtained solutions for every method is calculated for all numbers of objectives provided in Table 1. The results are presented in Fig. 1 and are normalized according to the method with the minimal s-energy on every dimension for a particular value p, such that the method with the least relative log_{10} s-energy value is at the bottom of the plots.

It is clear from the results in Fig. 2 that generalized decomposition performs extremely well in comparison with the other two methods for a wide range of Pareto front geometries. At the extremes, as $p \to 0$ and $p \to \infty$, its performance becomes comparable to the other two methods. However, consider what these two extremes (degenerate cases) mean for the problem in Eq. (1). When considering the limit $p \to 0$, Eq. (1) collapses to a single objective problem whose minimum, in our particular normalization, is the $\mathbf{0}$ vector, see Fig. 2. The second degeneracy is manifested when $p \to \infty$, in which case there are k optimal solutions. Again, for our particular normalization these solutions are the intersection of the "Pareto front" with the axes. In our experience, pathological cases like these are rare in practice and signify that the assumption that the different scalar objective functions, $f_i(\cdot)$, in Eq. (1) are *competing* is violated.

For intermediate values of p caution must be exercised when interpreting the results in Fig. 2. This is because we have not used an absolute basis for the minimal s-energy in the comparison. An exception to this is the case where $p = 1$, where the distribution of points that minimizes the s-energy is known for an affine PF geometry. For any other case, the distribution that minimizes the s-energy of points distributed on a generalized hypersphere, which is referred to as the best packing problem, is unknown [49,50]. Because of this difficulty we have normalized using the method that produces the least s-energy to produce Fig. 2. For example, for a PF geometry with $p = 6$ and 11 objectives, generalized decomposition produces a distribution of points that are better distributed when compared with the other two methods and results in approximately 25 orders of magnitude lower s-energy. However, since the s-energy of the ideal distribution is unknown, we cannot say how far the distribution of solutions produced by generalized decomposition is from the ideal distribution. We only know that the ideal s-energy is smaller or equal to the best produced distribution here.

Therefore, generalized decomposition captures the desired distribution of the target PF to a much higher degree compared with the methods employed by [10,15], even when the Pareto front geometry is unknown. Of course the ideal performance will be obtained with generalized decomposition only when the PF geometry is known and there exists a method for points to be distributed perfectly on that manifold.

From this discussion it follows that for an MOEA based on generalized decomposition, the three performance objectives that an EA, when applied to an MAP, has to achieve, namely – (i) convergence, (ii) well distributed solutions along the PF and

[5] Namely for each $p = \{100, 6, 2, 1, \frac{1}{2}, \frac{1}{6}\}$.

[6] The generated set of weighting vectors is actually sub-optimal in this case since we assume that we do not know the PF geometry. If this information is available then the generated weighting vectors will be optimal. Optimal in the sense that these weighting vectors will produce a set of subproblems whose Pareto optimal solutions minimize the selected "goodness of distribution" measure.

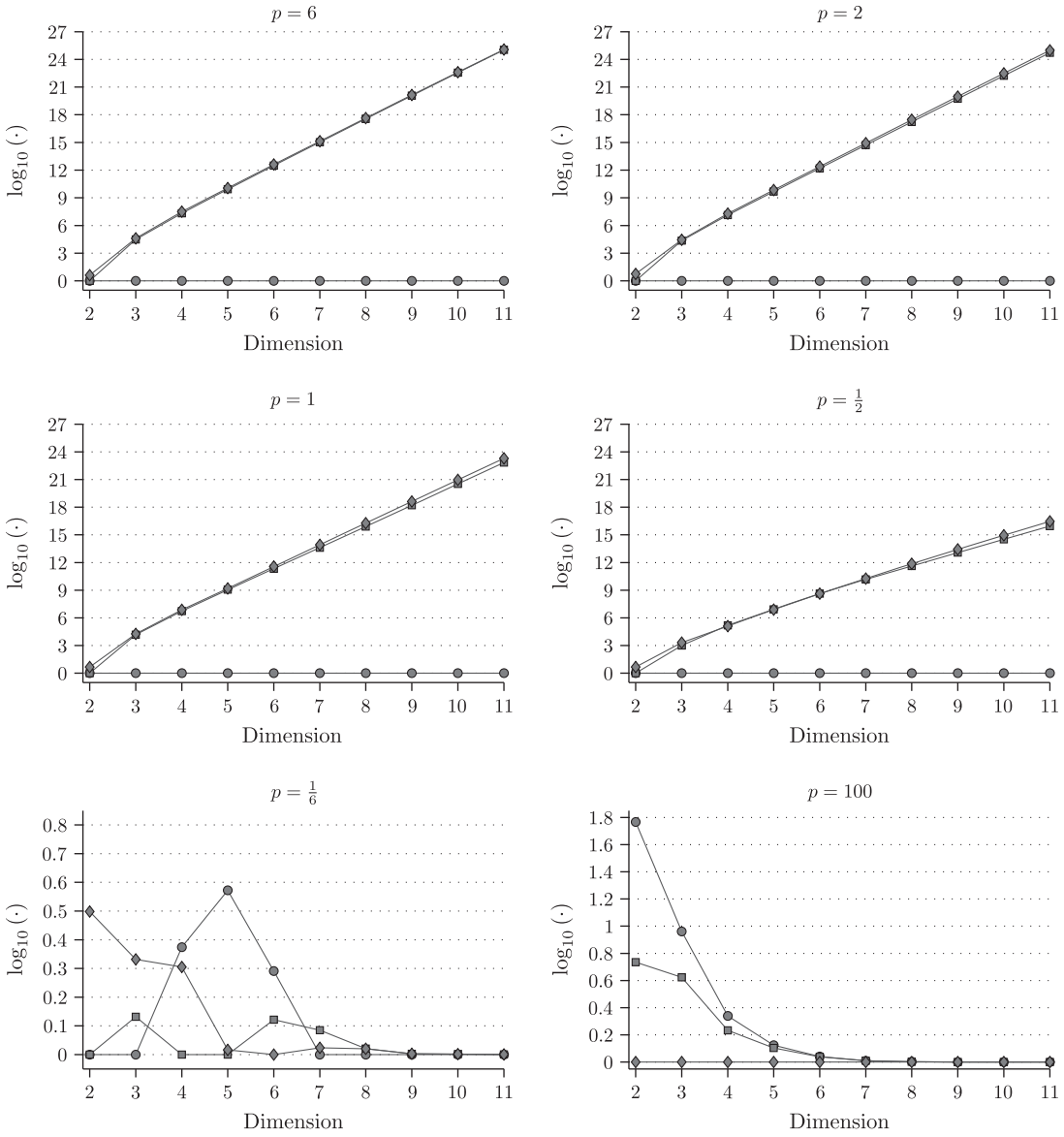

Fig. 2. Logarithm of the energy ratio of generalized decomposition (circles), relative to evenly distributed weighting vectors (squares) [10] and uniformly distributed weighting vectors (diamonds) [15], for different Pareto front geometries (see Fig. 1).

(iii) coverage of the entire PF – degenerate to only one, that of convergence. This is because with gD we can generate the appropriate weighting vectors that satisfy objectives (ii) and (iii), therefore the algorithm will have to focus only on minimizing the associated subproblems. If successful, then all three objectives will be satisfied, in the sense that the desired distribution of Pareto-optimal points will be achieved. An exception to this is when the Pareto front contains some degeneracy as in Fig. 1.

3. Cross entropy method

In this section we introduce the main ideas of the cross entropy method. Furthermore, in Section 3.2 we present the continuous version of CE, as it is employed in this work.

3.1. Introduction to the cross entropy method

The cross entropy method was introduced by Rubinstein [36] 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

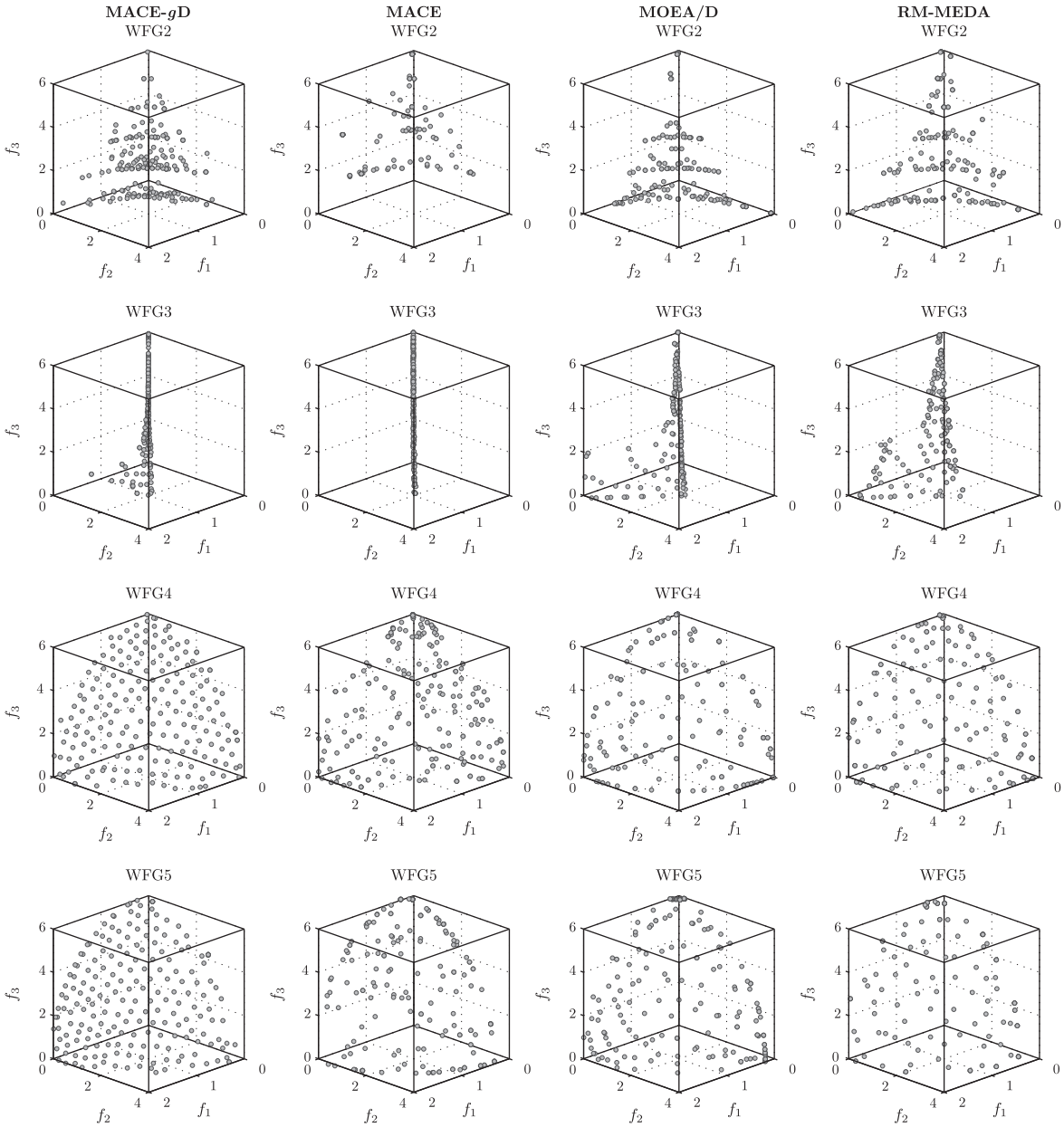

Fig. 3. MACE-**g**D, MACE, MOEA/D and RM-MEDA Pareto front for 3-objective instances of the WFG2–WFG5 test problems.

distribution for continuous problems, while Markov chains are employed in the discrete case [36]. It is interesting to note that in this form CE is similar, in principle, to probability collectives (PC), a method introduced by Wolpert [51] for distributed control and optimization.

In CE, the optimization problem is cast as a rare event estimation. 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.

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

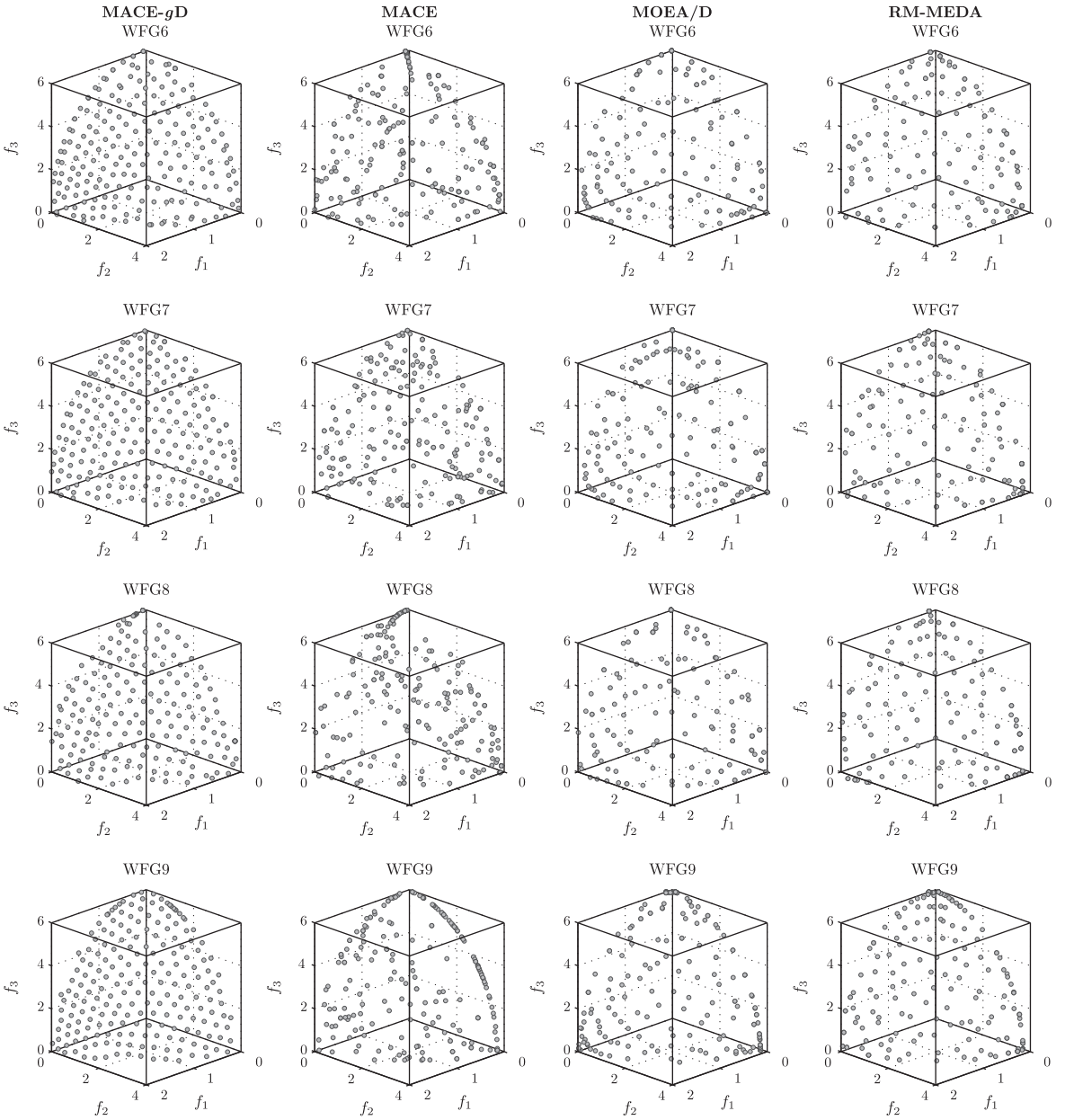

Fig. 4. MACE-**g**D, MACE, MOEA/D and RM-MEDA Pareto front for 3 objective instances of the WFG6–WFG9 test problems.

Since its introduction, several studies expanding on the initial methodology have been presented. Most notably, the minimum cross-entropy (MCE) method [52], where a non-parametric instrumental distribution is used. Albeit, MCE is computationally more demanding compared with CE. Another interesting approach to extend CE, presented by Botev et al. [53], 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:

$$\min_{\mathbf{x}} f(\mathbf{x}), \tag{12}$$

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 $\mathbf{S}$, Eq. (12) can be interpreted in a different way, i.e. as a rare event estimation. Therefore Eq. (12) can be restated as follows:

[8] By rare in this context we mean that for, $C = \{\mathbf{x} : \|\mathbf{x}^\star - x\|_2 \leqslant \varepsilon, \varepsilon > 0\}$ 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.

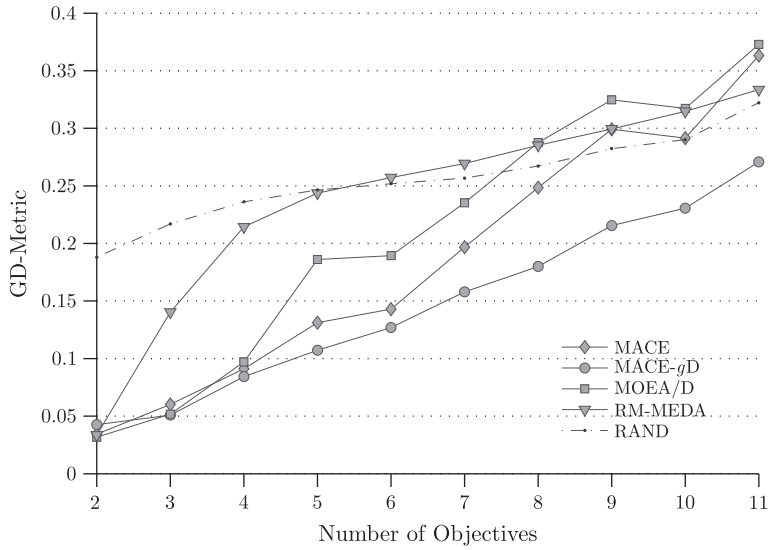

Fig. 5. Mean GD-metric performance of studied algorithms over WFG2–9 for 2–11 objectives.

$$\mathbb{E}_{\mathbf{u}} I_{f(\mathcal{X})\leqslant\gamma} = \mathbf{P}_{\mathbf{u}}(f(\mathcal{X}) \leqslant \gamma) = \ell, \tag{13}$$

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 $H(\mathcal{X};\gamma) \equiv I_{f(\mathcal{X})\leqslant\gamma}$,

$$H(\mathcal{X};\gamma) = \begin{cases} 1 & f(\mathcal{X}) \leqslant \gamma \\ 0 & f(\mathcal{X}) > \gamma. \end{cases} \tag{14}$$

Now to estimate ℓ for some $\tilde{\gamma}$ that $\|\tilde{\gamma} - \gamma^{\star}\| \leqslant \epsilon$, 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 that the aforementioned assumption is not true, ℓ can be estimated using the *crude Monte Carlo* (CMC) estimator:

$$\hat{\ell} = \frac{1}{N}\sum_{i=1}^{N} H(\mathcal{X};\gamma). \tag{15}$$

If, however, our prior assumption holds that the indicator function $I_{f(\mathcal{X})\leqslant\rho}$ in Eq. (15) 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:

$$\hat{\ell} = \frac{1}{N}\sum_{i=1}^{N} W(\mathcal{X}_i;\mathbf{u},\mathbf{v}) H(\mathcal{X};\gamma), \tag{16}$$

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:

$$g^{\star}(\mathbf{x}) = \frac{f(\mathbf{x};\mathbf{u})H(\mathcal{X};\gamma)}{\ell}. \tag{17}$$

However Eq. (17) 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:

$$\mathcal{D}(g,h) = \int g(\mathbf{x}) \ln\left(\frac{g(\mathbf{x})}{h(\mathbf{x})}\right) d\mathbf{x}, \tag{18}$$

and upon expansion:

$$\mathcal{D}(g,h) = \int g(\mathbf{x}) \ln g(\mathbf{x})\ d\mathbf{x} - \int g(\mathbf{x}) \ln h(\mathbf{x})\ d\mathbf{x}. \tag{19}$$

Since the first term in Eq. (19) 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:

$$\mathbf{v}^{\star} = \max_{\mathbf{v}} \mathbb{E}_{\tilde{\mathbf{v}}} H(\mathcal{X};\gamma) W(\mathcal{X};\mathbf{u},\tilde{\mathbf{v}}) \ln g(\mathcal{X};\mathbf{v}), \tag{20}$$

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 $\bar{\gamma} \geqslant \gamma$ equal to the ρ-quantile of $f(\mathcal{X})$ under $\mathbf{v}$. The program in Eq. (20) is solved for decreasing levels of $\bar{\gamma}$ until $\bar{\gamma} \leqslant \gamma$. So Eq. (20), in the CE method, becomes:

$$\mathbf{v}_t = \max_{\mathbf{v}} \mathbb{E}_{\mathbf{v}_{t-1}} H(\mathcal{X};\gamma_{t-1}) W(\mathcal{X};\mathbf{u},\mathbf{v}_{t-1}) \ln\, g(\mathcal{X};\mathbf{v}), \tag{21}$$

whose stochastic counterpart is:

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

where $\mathcal{X}_1,\ldots,\mathcal{X}_N$ is drawn from $g(\cdot;\mathbf{v}_{t-1})$. Typically Eq. (22) is convex and if the instrumental densities $g(\cdot;\cdot)$ are chosen from the *natural exponential family* (NEF) [54], then, Eq. (22) can be solved analytically [52] by solving the following system of equations:

$$\max_{\mathbf{v}} \frac{1}{N} \sum_{i=1}^{N} H(\mathcal{X}_i) W(\mathcal{X}_i;\mathbf{u},\mathbf{v}_{t-1}) \nabla_{\mathbf{v}} \ln g(\mathcal{X}_i;\mathbf{v}) = 0. \tag{23}$$

The updating rules for the instrumental densities can be obtained analytically which translates to a much lower computational overhead. This is a major strength in CE. Briefly, some distributions in the NEF family are the Gaussian, Poisson and the gamma distributions [55].

The procedure described by Eqs. (21)–(23) 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 $g(\cdot;\mathbf{v})$.

3.2. 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 $|\gamma_t - \gamma_{t-1}| \leqslant \delta$ for some small δ.

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

$$g(x;\mu,\sigma) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right), \tag{24}$$

and its truncated equivalent for problems with boundary constraints. We should mention that the updating rules derived using Eq. (23) are identical for the regular and truncated Gaussian [53].

It is suggested in [52] that for the optimization case, 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 available information, 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 Eq. (23) we obtain the updating rules for the normal distribution $\mathbf{v}_t = \{\mu_t, \sigma_t\}$:

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

$$\hat{\sigma}_t = \left(\frac{\sum_{i=1}^{\rho N} W(\mathcal{X}_i;\mathbf{u},\mathbf{v}_{t-1})(\mathcal{X}_i - \hat{\mu})^2}{\sum_{i=1}^{\rho N} W(\mathcal{X}_i;\mathbf{u},\mathbf{v}_{t-1})}\right)^{\frac{1}{2}}, \tag{26}$$

where ρ is some small value, e.g. 0.1. The updating rules in Eqs. (25) and (26) could lead to premature convergence [52], so a *smoothed* version is usually employed:

$$\begin{aligned} \mu_t &= \alpha\hat{\mu}_t + (1-\alpha)\mu_{t-1} \\ \sigma_t &= \beta_t\hat{\sigma}_t + (1-\beta_t)\sigma_{t-1}, \end{aligned} \tag{27}$$

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

$$\begin{aligned} &\beta_t = \beta - \beta\left(1 - \frac{1}{t}\right)^q, \\ &\beta \in (0.7, 1), \\ &q \in (5, 9). \end{aligned} \tag{28}$$

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.

4. Generalized decomposition-based many objective cross-entropy

The proposed algorithm is based on the CE method, see Section 3, and the newly introduced concept of generalized decomposition, as described in Section 2, and is known as *many-objective cross entropy based on generalized decomposition* (MACE-gD) 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 a search for Pareto optimal solutions away from such regions. The *main* algorithm in MACE-gD is the CE method for continuous optimization problems, as described in Section 3.2.

Algorithm 1. MACE-gD.

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: $\mathcal{M}_i^{(t)} \leftarrow \alpha\hat{\boldsymbol{\mu}}_t + (1-\alpha)\hat{\boldsymbol{\mu}}_{t-1}$
14: $\mathcal{S}_i^{(t)} \leftarrow \beta_t\hat{\boldsymbol{\sigma}}_t + (1-\beta_t)\hat{\boldsymbol{\sigma}}_{t-1}$
15: $\hat{\mathbf{x}}_i^{(t)} \leftarrow \mathcal{N}\left(\mathcal{M}_i^{(t)}, \mathcal{S}_i^{(t)}\right)$
16: $\hat{\mathbf{V}}_i^{(t)} \leftarrow g_\infty\left(\hat{\mathbf{x}}_i^{(t)}, \mathbf{w}_i, \mathbf{z}^\star\right)$
17: **if** $\hat{\mathbf{V}}_i^{(t)} \leqslant \mathbf{V}_i^{(t)}$ **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** $t \leqslant MaxGenerations$
25: $\mathbf{x} \leftarrow \mathcal{M}^{(t)}$

An overview of MACE-gD 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. 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 [10]. 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 Eq. (7) for every point in the reference PF generated in **Step 1** to obtain the optimal weighting vectors **w**.

The reference Pareto front used in this work for the WFG4–9 test problems in Section 6.3 is an evenly distributed set on a concave front. 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:

$$\begin{pmatrix} \mathcal{N}(\mu_{1,1},\sigma_{1,1}) & \cdots & \mathcal{N}(\mu_{1,n},\sigma_{1,n}) \\ \vdots & \ddots & \vdots \\ \mathcal{N}(\mu_{N,1},\sigma_{N,1}) & \cdots & \mathcal{N}(\mu_{N,n},\sigma_{N,n}) \end{pmatrix}, \tag{29}$$

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-gD 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 ith 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 $\mathcal{M}^{(t)}$.

5. Benchmark algorithms

The aims of the empirical testing of MACE-gD that follows are twofold: (1) to compare the algorithm to the existing best-in-class methods for (a) decomposition-based optimization and (b) multi-objective EDAs; (2) to compare the impact of generalized decomposition to the popular even distribution scheme for weight vectors. To satisfy aim (1), we compare MACE-gD against MOEA/D and also the *regularity model-based estimation of distribution algorithm* (RM-MEDA) [56]. To satisfy aim (2) we introduce a version of MACE-gD that employs a set of evenly spaced weighting vectors instead of using generalized decomposition; this version is simply referred to as MACE.

5.1. Multi-objective evolutionary algorithm based on decomposition

As mentioned in Section 1, 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. [10], 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 [10].

MOEA/D depends on one of several available decomposition techniques, – weighted sum, Chebyshev [4] and a penalty-based variant of the normal boundary intersection [57,10] decompositions – with each having its own strengths and weaknesses. The minimization problem from Section 1, when using the Chebyshev decomposition is restated according to Eq. (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:

$$w_1^i = \frac{i}{H}, \quad w_2^i = 1 - w_1^i, \quad i \in \{0,\ldots,H\}, \tag{30}$$

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

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 [10]. 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) [57]. In NBI the following program is to be solved:

$$\begin{aligned} &\min_{\mathbf{x}} \; g_{nbi}(\mathbf{x}; \mathbf{w}^i, \mathbf{z}^\star) = d \\ &\text{subject to } \mathbf{z}^\star - \mathbf{F}(\mathbf{x}) = d \cdot \mathbf{w}^i, \end{aligned} \tag{31}$$

where Zhang et al. [10] suggest a penalty function approach to handle the equality constraint. Thus Eq. (31) is transformed to:

$$\begin{aligned} &\min_{\mathbf{x}} \; g_{nbi}(\mathbf{x}; \mathbf{w}^i, \mathbf{z}^\star) = d_1 + p d_2 \\ &d_1 = \frac{\|(\mathbf{z}^\star - \mathbf{F}(\mathbf{x}))^T \mathbf{w}^i\|_2}{\|\mathbf{w}^i\|_2}, \\ &d_2 = \|\mathbf{F}(\mathbf{x}) - (\mathbf{z}^\star - d_1 \mathbf{w}^i)\|_2, \end{aligned} \tag{32}$$

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 Eq. (32) has the potential to produce truly evenly distributed Pareto optimal solutions [10]. Unfortunately Eq. (32) has three significant drawbacks. First, the normal-boundary intersection method does not guarantee that the solutions to the subproblems will be Pareto optimal [57]. 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 Eq. (30). Create a matrix B containing the nearest neighbors 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 neighborhood of each solution. The choice of individuals is random among neighboring solutions.
Step 4.2 Evaluate the newly generated solution using Eq. (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 ith subproblem with the new solution. Otherwise, retain the old solution.
Step 4.5 Check if the new solution is better for any of the neighboring 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 [10].

5.2. Regularity model-based estimation of distribution algorithm

The second algorithm that we employ in our comparative studies, see Section 6, is the regularity model-based multi-objective estimation of distribution algorithm (RM-MEDA) proposed by Zhang et al. [56]. 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. [56] used inductively the Karush–Kuhn–Tucker condition [4] 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. [56] 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. An issue 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 [56] RM-MEDA was evaluated against PCX-NSGA-II [58], GDE3 [59] and MIDEA [60] and, on average, outperformed the aforementioned algorithms on variants of the ZDT[10] test problems [28]. 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

[10] Zitzler, Deb, Thiele (ZDT).

Table 2
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
$\lvert\mathcal{P}^\star\rvert$	500	1000	1500	2000	2500	3000	3500	4000	4500	5000

iteration. The implementation of RM-MEDA that is employed in this work is the publicly available version in MATLAB code provided by the authors [56].

5.3. Random search

Due to the difficulties encountered by MOEAs in solving MAPs with more than 3 objectives, random search can still be regarded as an appropriate baseline comparator for empirical testing. 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 [61]. 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 6.

6. Comparative studies

6.1. Performance indicator

In Section 2, it was argued 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. The most important metric of interest, therefore, becomes some measure of convergence to the PF. Therefore the generational distance (GD) indicator has been chosen as the main performance metric for our comparative study.

- Generational Distance (GD), introduced in [62], is defined thus:

$$D(A,\mathcal{P}^\star) = \frac{\sum_{s\in A}\min\{\|\mathcal{P}_1^\star - s\|_2,\ldots,\|\mathcal{P}_N^\star - s\|_2\}}{|A|} \tag{33}$$

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.

6.2. Experiment description

The problem set chosen to perform the experiments is the WFG toolkit [26], specifically problems WFG2–WFG9, since they contain several challenging problems, are scalable and the PFs are known. 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. We assume that the DM is interested in a even distribution of solution images across the entire Pareto front and generate an appropriate reference set, $|\mathcal{P}^\star|$.

The neighborhood size T in MOEA/D was selected to be 10% of the population size N, since, according to [13], this appears to be a setting producing good results for MAPs. The population size was the same for all the algorithms, see Table 2. The parameters of the CE method are the same in MACE and MACE-gD and have been selected according to the suggestions in [52], see Table 3. Lastly, the reference Pareto fronts used in MACE-gD 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, an even distribution of points on a concave Pareto front geometry.

6.3. Experiment results

A summary of the GD-metric performance of the algorithms is presented in Tables 4–11. 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, based on 50 independent trials, to verify whether the mean performance of the studied algorithms is different.

Table 3
Settings for MACE and MACE-gD.

ρ	α	β	q
0.1	0.9	0.9	7

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 count. Since we have 5 algorithms, the maximum count for an algorithm is 4. A count 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 Tables 4–11. An algorithm with a count of 4 is denoted with a (1), one with a count 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 4–11.

Table 4 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 weighting vector set 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, see Fig. 2. 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, could lead to the hypothesis that Pareto-based algorithms are potentially not very well suited for problems with convex PF geometries in high dimensions. While this hypothesis appears valid for this experiment it should be emphasized that this does not necessarily speak to the performance of other Pareto-based algorithms. This hypothesis is further supported by the fact that RM-MEDA uses a variant of non-dominated sorting [56]. 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 5. 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,[11] 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 6 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 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 7 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

[11] Intuitively the feasible objective space resembles a *wedge* whose edge is the PF.

Table 4
GD-metric performance of the studied algorithms on the WFG2 problem for 2–11 objectives.

WFG2					
Obj. #	MACE	MACE-gD	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 5
GD-metric performance of the studied algorithms on the WFG3 problem for 2–11 objectives.

WFG3					
Obj. #	MACE	MACE-gD	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)

Table 6
GD-metric performance of the studied algorithms on the WFG4 problem for 2–11 objectives.

WFG4					
Obj. #	MACE	MACE-gD	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 7
GD-metric performance of the studied algorithms on the WFG5 problem for 2–11 objectives.

WFG5					
Obj. #	MACE	MACE-gD	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 8
GD-metric performance of the studied algorithms on the WFG6 problem for 2–11 objectives.

WFG6					
Obj. #	MACE	MACE-gD	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 9
GD-metric performance of the studied algorithms on the WFG7 problem for 2–11 objectives.

WFG7					
Obj. #	MACE	MACE-gD	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)

Table 10
GD-metric performance of the studied algorithms on the WFG8 problem for 2–11 objectives.

WFG8					
Obj. #	MACE	MACE-gD	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)

Table 11
GD-metric performance of the studied algorithms on the WFG9 problem for 2–11 objectives.

WFG9					
Obj. #	MACE	MACE-gD	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)

being the 2-objective test instance. However for more than 9 objectives, random search outperforms 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 [34] 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 [63,64].

Table 8 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 Tables 7–10. 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 is non-separable, as is the case for WFG2–3 and WFG8–9, see Tables 4 and 5 and Tables 10 and 11. 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.

Tables 9 and 10 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 4–8.

Lastly, Table 11 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 6 for WFG4, 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.

6.4. 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 sharing between different subproblems. In that context it is similar to the T parameter in MOEA/D. However the *neighborhood* 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, the neighborhood 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 each of $\rho = \{0.1, 0.2, \ldots, 0.9\}$ on the WFG9 problem. All other parameters are identical to those employed in Section 6.3. The results can be seen in Figs. 6 and 7. In Fig. 6 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. 6, is better when compared with MACE is expected, given the results in Table 11. 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. 6 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. 7, 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 behavior 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

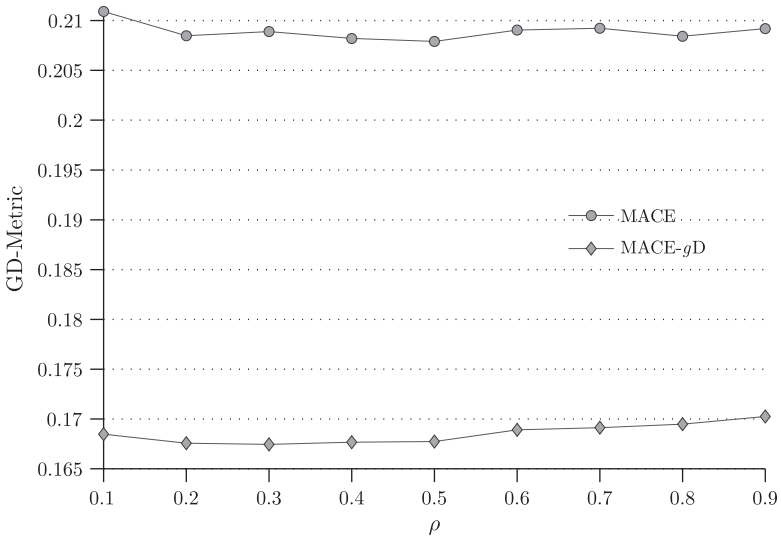

Fig. 6. Mean GD-metric performance of MACE-gD (circles) and MACE (diamonds), over all objectives for the WFG9 test problem.

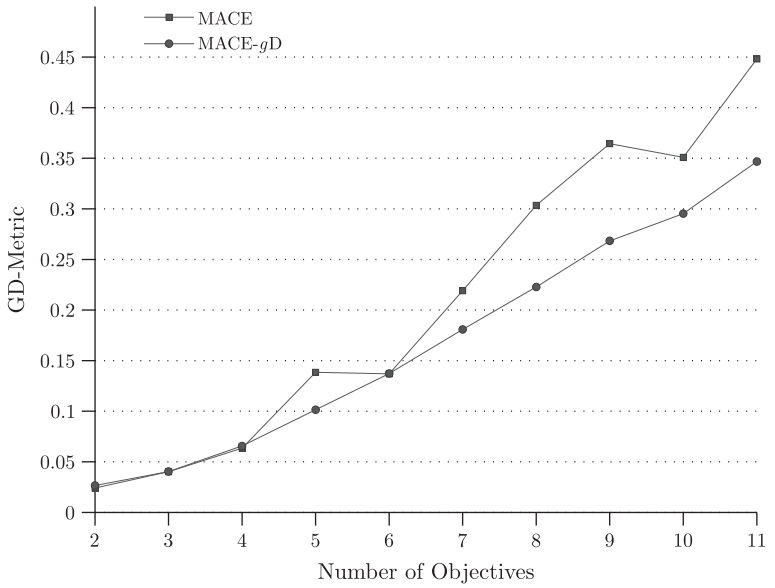

Fig. 7. Mean GD-metric performance of MACE and MACE-gD, over all ρ values for the WFG9 test problem.

same fashion as in MOEA/D, subproblems are aggregated in a very small region of the PF, therefore sharing information with neighboring solutions is less disruptive. Conversely, the weighting vectors in MACE-gD 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-gD appears to be a quasi-convex function of ρ, see Fig. 6. We believe this is due to the presence of two competing trends in MACE-gD. First, as we increase ρ, more samples are used in the updating rules in Eqs. (25) and (26), 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. 7 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 6.3. Additionally, it seems that the linear scaling of performance of the MACE-gD algorithm as seen in Fig. 5, is persistent for a range of ρ values.

7. 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 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 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-gD, see Figs. 3 and 4. As can be seen, the solutions produced by MACE-gD 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 to the ones used in Section 6.3. 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. 8. 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. 8 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,

$$\begin{aligned}
&C_1 = \left\{\mathbf{z} : (z_1 - c_1)^2 + (z_2 - c_2)^2 + (z_3 - c_3)^2 \geqslant r^2\right\},\\
&r^2 = 0.65,\ \mathbf{c} = (0.33, 0.33, 0.33),\\
&C_2 = \left\{\mathbf{z} : (z_1 - c_1)^2 + (z_2 - c_2)^2 + (z_3 - c_3)^2 \leqslant r^2\right\},\\
&r^2 = 0.15,\ \mathbf{c} = (0.53, 0.23, 0.8),\\
&C_3 = \left\{\mathbf{z} : (z_1 - c_1)^2 + (z_2 - c_2)^2 + (z_3 - c_3)^2 \leqslant r^2\right\},\\
&r^2 = 0.1,\ \mathbf{c} = (0.23, 0.53, 0.8),
\end{aligned}$$

and,

$$\begin{aligned}
&C_4 = C_a \cap C_b,\\
&C_a = \{\mathbf{z} : (z_1 - c_1)^2 + (z_2 - c_2)^2 + (z_3 - c_3)^2 \geqslant r_a^2\},\\
&C_b = \{\mathbf{z} : (z_1 - c_1)^2 + (z_2 - c_2)^2 + (z_3 - c_3)^2 \leqslant r_b^2\},\\
&r_a^2 = 0.2, \quad r_b^2 = 0.27,\ \mathbf{c} = (0.63, 0.63, 0.38).
\end{aligned}$$

Subsequently Eq. (7) was solved to obtain the weighting vectors corresponding to these regions and using these weighting vectors MACE-gD was able to generate a PF that closely resembles the initially chosen regions, see Fig. 8. This concept extends directly to MAPs, however the results are much more difficult to visualize.

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.

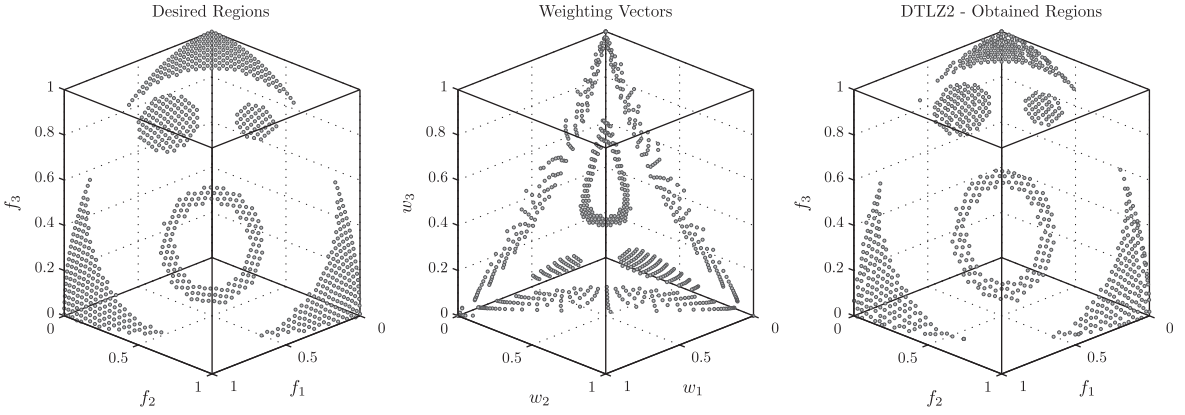

Fig. 8. Left: Preferred regions of the Pareto front. Middle: Weighting vectors corresponding to the preferred PF regions. Right: Obtained Pareto optimal solutions on a 3-objective instance of the DTLZ2.

8. 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, for these benefits to materialize to their fullest, certain prior information is needed. Specifically, the geometry of the Pareto front needs to be known *a priori* and a method needs to be available to generate the distribution across that geometry that the DM requires. 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. Nevertheless, even if such information is not available we have shown that assuming an affine PF geometry and distributing solutions on that manifold according to some measure of *goodness of distribution* can still produce results which are dramatically superior to the alternative methods available (see Fig. 2).

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. A benefit of this is that CE is based on sound theoretical principles which can facilitate further analysis of this method. Also, the hypothesis presented in [34], 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 6.3. However, as no clustering method is 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. 5, with respect to the GD-metric. Also, MACE-*g*D seems to be robust with respect to its main control parameter, ρ, see Section 6.4. 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.

Acknowledgments

The authors would like thank Jacob Mattingley for providing access to his tool CVXGEN [66]. In this work CVXGEN is employed to solve Eq. (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.

Appendix A. Convex sets and functions

Some fundamental definitions about convex sets and functions are given below. For further details the reader is referred to [41] for an applications oriented presentation and [65] for a more theoretical approach.

A.1. Convex sets

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

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

The combination of the points $\mathbf{x}$, $\mathbf{y}$ in Eq. (A.1), is called a *convex combination* and can be extended to multiple points in a similar manner to the extension of affine combinations:

$$\sum_{i=1}^{d} \theta_i \mathbf{x}_i, \quad \text{with } \sum_{i=1}^{d} \theta_i = 1, \text{ and } \theta_i \geqslant 0, \text{ for all } i = 1, \ldots, d. \tag{A.2}$$

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

$$\mathbf{conv}\ C = \left\{ \sum_{i=1}^{d} \theta_i \mathbf{x}_i : \mathbf{x}_i \in C,\ \sum_{i=1}^{d} \theta_i = 1, \theta_i \geqslant 0 \right\}, \tag{A.3}$$

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:

$$f(\theta \mathbf{x} + (1 - \theta)\mathbf{y}) \leqslant \theta f(\mathbf{x}) + (1 - \theta) f(\mathbf{y}). \tag{A.4}$$

A function is strictly convex if the inequality in Eq. (A.4) 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 A.2.

A.2. 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:

$$\mathbf{epi}\ f = \{(\mathbf{x}, t) : \mathbf{x} \in \mathbf{dom}\, f, t \in \mathbb{R}, f(\mathbf{x}) \leqslant t\}, \tag{A.5}$$

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:

$$\mathbf{hypo}\ f = \{(\mathbf{x}, t) : \mathbf{x} \in \mathbf{dom}\, f, t \in \mathbb{R}, f(\mathbf{x}) \geqslant t\}. \tag{A.6}$$

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 (see Fig. A.9):

- 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.

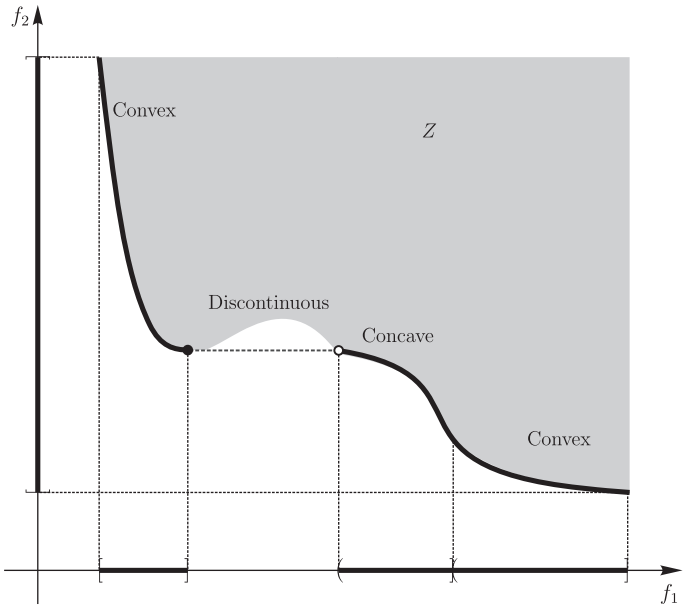

Fig. A.9. A Pareto front which is partially convex, partially concave and discontinuous. Notice that the frame of reference, which in this case is f_1, used to determine the convex and concave parts is arbitrary, namely the same parts of the Pareto front would be partially convex and concave, even if 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.

A.3. Pareto front geometry

Assuming that the Pareto front can be represented by a piecewise continuous function, $g : \mathbb{R}^{k-1} \rightarrow \mathbb{R}$ 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 **epi** g is a convex set.
- Concave, if **hypo** g is a convex set.
- Affine, if both **epi** g and **hypo** g are convex.
- Discontinuous, if **dom** g is nonconvex or g is discontinuous.
- Partially convex, if g is convex over a convex subset of **dom** g.
- Partially concave, if g is concave over a convex subset of **dom** g.
- Partially affine, if g is convex and concave over a convex subset of **dom** g.
- Piecewise convex, if g partially convex over all convex subsets of **dom** g.
- Piecewise concave, if g partially concave over all convex subsets of **dom** g.
- Piecewise affine, if g partially affine over all convex subsets of **dom** g.

References

[1] P. Fleming, R. Purshouse, Evolutionary algorithms in control systems engineering: a survey, Control Eng. Pract. 10 (11) (2002) 1223–1241.
[2] M. Tapia, C. Coello, Applications of multi-objective evolutionary algorithms in economics and finance: a survey, in: IEEE Congress on Evolutionary Computation, 2007, pp. 532–539.
[3] N. Krasnogor, W. Hart, J. Smith, D. Pelta, Protein structure prediction with evolutionary algorithms, in: Proceedings of the Genetic and Evolutionary Computation Conference, vol. 2, 1999, pp. 1596–1601.
[4] K. Miettinen, Nonlinear Multiobjective Optimization, vol. 12, Springer, 1999.
[5] E. Zitzler, M. Laumanns, L. Thiele, et al., SPEA2: improving the strength Pareto evolutionary algorithm, in: EUROGEN, no. 103, 2001, pp. 1–21.
[6] R. Purshouse, P. Fleming, Evolutionary many-objective optimisation: an exploratory analysis, IEEE Congress on Evolutionary Computation, vol. 3, IEEE, 2003, pp. 2066–2073.
[7] D. Goldberg, J. Holland, Genetic algorithms and machine learning, Mach. Learn. 3 (2) (1988) 95–99.
[8] I. Giagkiozis, R.C. Purshouse, P.J. Fleming, An overview of population-based algorithms for multi-objective optimisation, Int. J. Syst. Sci. 0 (0) (2013) 1–28.
[9] C. Fonseca, P. Fleming, Genetic algorithms for multiobjective optimization: formulation, discussion and generalization, in: Conference on Genetic Algorithms, vol. 423, 1993, pp. 416–423.
[10] Q. Zhang, H. Li, MOEA/D: a multiobjective evolutionary algorithm based on decomposition, IEEE Trans. Evol. Comput. 11 (6) (2007) 712–731.
[11] F. Edgeworth, Mathematical Psychics: An Essay on the Application of Mathematics to the Moral Sciences, no. 10, CK Paul, 1881.
[12] V. Pareto, Cours d'économie politique, Librairie Droz, 1896.
[13] H. Ishibuchi, N. Tsukamoto, Y. Nojima, Evolutionary many-objective optimization: a short review, in: IEEE Congress on Evolutionary Computation, 2008, pp. 2419–2426.
[14] R. Takahashi, R. Saldanha, W. Dias-Filho, J. Ramírez, A new constrained ellipsoidal algorithm for nonlinear optimization with equality constraints, IEEE Trans. Magn. 39 (3) (2003) 1289–1292.
[15] A. Jaszkiewicz, On the performance of multiple-objective genetic local search on the 0/1 knapsack problem – a comparative experiment, IEEE Trans. Evol. Comput. 6 (4) (2002) 402–412.
[16] E. Hughes, Multiple single objective Pareto sampling, Congress on Evolutionary Computation, vol. 4, IEEE, 2003, pp. 2678–2684.
[17] E. Hughes, MSOPS-II: a general-purpose many-objective optimiser, in: IEEE Congress on Evolutionary Computation, 2007, pp. 3944–3951.
[18] I. Giagkiozis, R.C. Purshouse, P.J. Fleming, Generalized decomposition, in: Evolutionary Multi-Criterion Optimization, Lecture Notes in Computer Science, vol. 7811, Springer, Berlin, Heidelberg, 2013, pp. 428–442.
[19] S. Jiang, Z. Cai, J. Zhang, Y.-S. Ong, Multiobjective optimization by decomposition with Pareto-adaptive weight vectors, in: International Conference on Natural Computation, vol. 3, 2011, pp. 1260–1264.
[20] S. Jiang, J. Zhang, Y. Ong, Asymmetric Pareto-adaptive scheme for multiobjective optimization, in: Advances in Artificial Intelligence, Lecture Notes in Computer Science, vol. 7106, Springer, Berlin, Heidelberg, 2011, pp. 351–360.
[21] E. Zitzler, L. Thiele, Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach, IEEE Trans. Evol. Comput. 3 (4) (1999) 257–271.
[22] L. While, P. Hingston, L. Barone, S. Huband, A faster algorithm for calculating hypervolume, IEEE Trans. Evol. Comput. 10 (1) (2006) 29–38.
[23] C. Fonseca, L. Paquete, M. Lopez-Ibanez, An improved dimension-sweep algorithm for the hypervolume indicator, in: IEEE Congress on Evolutionary Computation, 2006, pp. 1157–1163.
[24] F. Gu, H. Liu, K. Tan, A multiobjective evolutionary algorithm using dynamic weight method, Int. J. Innov. Comput. Inform. Control 8 (5B) (2012) 3677–3688.
[25] I. Giagkiozis, R. Purshouse, P. Fleming, Towards understanding the cost of adaptation in decomposition-based optimization algorithms, in: IEEE International Conference on Systems, Man and Cybernetics, 2013, pp. 615–620.
[26] S. Huband, P. Hingston, L. Barone, L. While, A review of multiobjective test problems and a scalable test problem toolkit, IEEE Trans. Evol. Comput. 10 (5) (2006) 477–506.
[27] K. Deb, L. Thiele, M. Laumanns, E. Zitzler, Scalable multi-objective optimization test problems, in: Congress on Evolutionary Computation, vol. 1, 2002, pp. 825–830.
[28] E. Zitzler, K. Deb, L. Thiele, Comparison of multiobjective evolutionary algorithms: empirical results, Evol. Comput. 8 (2) (2000) 173–195.
[29] H. Mühlenbein, G. Paass, From recombination of genes to the estimation of distributions I. Binary parameters, Parall. Probl. Solving Nat. (1996) 178–187.
[30] J. He, X. Yao, Drift analysis and average time complexity of evolutionary algorithms, Artif. Intell. 127 (1) (2001) 57–85.
[31] T. Chen, K. Tang, G. Chen, X. Yao, Analysis of computational time of simple estimation of distribution algorithms, IEEE Trans. Evol. Comput. 14 (1) (2010) 1–22.
[32] M. Hauschild, M. Pelikan, A Survey of Estimation of Distribution Algorithms, Tech. Rep., University of Missouri – St. Louis, 2011.
[33] M. Pelikan, Bayesian optimization algorithm, Hierarchical Bayesian Optim. Algor. (2005) 31–48.
[34] L. Emmendorfer, A. Pozo, Effective linkage learning using low-order statistics and clustering, IEEE Trans. Evol. Comput. 13 (6) (2009) 1233–1246.

[35] C. Echegoyen, Q. Zhang, A. Mendiburu, R. Santana, J. Lozano, On the limits of effectiveness in estimation of distribution algorithms, in: IEEE Congress on Evolutionary Computation, IEEE, 2011, pp. 1573–1580.
[36] R. Rubinstein, The cross-entropy method for combinatorial and continuous optimization, Methodol. Comput. Appl. Probabi. 1 (2) (1999) 127–190.
[37] S.B. Damelin, P.J. Grabner, Energy functionals, numerical integration and asymptotic equidistribution on the sphere, J. Complex. 19 (3) (2003) 231–246.
[38] K. Miettinen, M. Mäkelä, On scalarizing functions in multiobjective optimization, OR Spectrum 24 (2) (2002) 193–213.
[39] I. Giagkiozis, P. Fleming, Methods for Many-Objective Optimization: An Analysis, Research Report No. 1030, November 2012.
[40] M. Grant, S. Boyd, Y. Ye, in: Disciplined Convex Programming, Nonconvex Optimization and its Applications, vol. 84, Springer, 2006, pp. 155–210.
[41] S. Boyd, L. Vandenberghe, Convex Optimization, Cambridge University Press, 2004.
[42] M. Grant, S. Boyd, CVX: Matlab Software for Disciplined Convex Programming. <http://cvxr.com/cvx/>.
[43] E. Saff, A. Kuijlaars, Distributing many points on a sphere, Math. Intell. 19 (1) (1997) 5–11.
[44] D. Hardin, E. Saff, Discretizing manifolds via minimum energy points, Notices AMS 51 (10) (2004) 1186–1194.
[45] H. Li, Q. Zhang, Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II, IEEE Trans. Evol. Comput. 13 (2) (2009) 284–302.
[46] H. Thompson, A. Chipperfield, P. Fleming, C. Legge, Distributed aero-engine control systems architecture selection using multi-objective optimisation, Control Eng. Pract. 7 (5) (1999) 655–664.
[47] R. Tavakkoli-Moghaddam, A. Rahimi-Vahed, A. Mirzaei, A hybrid multi-objective immune algorithm for a flow shop scheduling problem with bi-objectives: weighted mean completion time and weighted mean tardiness, Inform. Sci. 177 (22) (2007) 5072–5090.
[48] G. Zhang, X. Shao, P. Li, L. Gao, An effective hybrid particle swarm optimization algorithm for multi-objective flexible job-shop scheduling problem, Comput. Ind. Eng. 56 (4) (2009) 1309–1318.
[49] A. Katanforoush, M. Shahshahani, Distributing points on the sphere, I, Exp. Math. 12 (2) (2003) 199–210.
[50] P. Leopardi, Distributing Points on the Sphere: Partitions, Separation, Quadrature and Energy, Ph.D. Thesis, University of New South Wales, 2007.
[51] D. Wolpert, Information theory – the bridge connecting bounded rational game theory and statistical physics, Complex Eng. Syst. (2006) 262–290.
[52] R. Rubinstein, A stochastic minimum cross-entropy method for combinatorial optimization and rare-event estimation, Methodol. Comput. Appl. Probabi. 7 (1) (2005) 5–50.
[53] Z. Botev, D. Kroese, T. Taimre, Generalized cross-entropy methods with applications to rare-event simulation and optimization, Simulation 83 (11) (2007) 785.
[54] P. De Boer, D. Kroese, S. Mannor, R. Rubinstein, A tutorial on the cross-entropy method, Ann. Oper. Res. 134 (1) (2005) 19–67.
[55] C.N. Morris, Natural exponential families with quadratic variance functions, Ann. Stat. 10 (1982) 65–80.
[56] Q. Zhang, A. Zhou, Y. Jin, RM-MEDA: a regularity model-based multiobjective estimation of distribution algorithm, IEEE Trans. Evol. Comput. 12 (1) (2008) 41–63.
[57] I. Das, J. Dennis, Normal-Boundary Intersection: An Alternate Method for Generating Pareto Optimal Points in Multicriteria Optimization Problems, Tech. Rep., DTIC Document, 1996.
[58] K. Deb, A. Sinha, S. Kukkonen, Multi-objective test problems, linkages, and evolutionary methodologies, in: Conference on Genetic and Evolutionary Computation, ACM, 2006, pp. 1141–1148.
[59] S. Kukkonen, J. Lampinen, GDE3: the third evolution step of generalized differential evolution, IEEE Congress on Evolutionary Computation, vol. 1, IEEE, 2005, pp. 443–450.
[60] P. Bosman, D. Thierens, The naive MIDEA: a baseline multi-objective EA, in: Evolutionary Multi-Criterion Optimization, Springer, 2005, pp. 428–442.
[61] D. Wolpert, W. Macready, No free lunch theorems for optimization, IEEE Trans. Evol. Comput. 1 (1) (1997) 67–82.
[62] D. Van Veldhuizen, Multiobjective evolutionary algorithms: classifications, analyses, and new innovations, in: Evolutionary Computation, 1999.
[63] R. Purshouse, P. Fleming, On the evolutionary optimization of many conflicting objectives, IEEE Trans. Evol. Comput. 11 (6) (2007) 770–784.
[64] D. Hadka, P. Reed, Diagnostic assessment of search controls and failure modes in many – objective evolutionary optimization, Evol. Comput. 20 (3) (2012) 423–452.
[65] R. Rockafellar, Convex Analysis, vol. 28, Princeton University Press, 1970.
[66] J. Mattingley, S. Boyd, CVXGEN: a code generator for embedded convex optimization, Optimiz. Eng. (2012) 1–27.