Struct Multidisc Optim (2013) 48:1129–1141
DOI 10.1007/s00158-013-0946-1

RESEARCH PAPER

# A modified directed search domain algorithm for multiobjective engineering and design optimization

**Tohid Erfani · Sergey V. Utyuzhnikov · Brian Kolo**

Received: 25 July 2012 / Revised: 5 May 2013 / Accepted: 9 May 2013 / Published online: 12 June 2013


**Abstract** Multiobjective optimization is one of the key challenges in engineering design process. Since the answer to such problem is not unique, a set of evenly distributed solutions is particularly important for a designer. The Directed Search Domain (DSD) method is a numerical optimization approach that has proven to be efficient enough to tackle such optimization problems. In this paper, we propose two modifications to the DSD approach which make the solution algorithm simpler for program implementation. These modifications are related to the control of the search domain and reformulation of the appropriate single objective optimization problem. As a result, the computational efficiency of the method is increased due to the lower number of objective function evaluations. The capabilities of the new approach are demonstrated on a set of test cases.

**Keywords** Directed search domain · Multiobjective optimization · Engineering design · Pareto set

T. Erfani (✉)
Civil, Environmental and Geomatic Engineering Department,
University College London,
Gower Street, London, WC1E 6BT, UK
e-mail: t.erfani@ucl.ac.uk

S. V. Utyuzhnikov
School of Mechanical, Aerospace and Civil Engineering,
The University of Manchester, Manchester, UK

B. Kolo
Zifian Inc., 11260 Roger Bacon Drive, Reston, VA 20190, USA

## 1 Background and motivation

Engineering design is an active and wide-ranging research area in optimization. Design process usually includes simultaneous optimizations of different criteria (objectives) which are often in conflict with one another. As a result, the ideal (best) solution for all objectives usually lies outside of the feasible design space. Therefore, the goal is to obtain a set of solutions which would represent the best trade-off among the objectives. Such solutions are called *Pareto* solutions. Mathematically, a solution is called a Pareto solution if no improvement with respect to one objective can be made without compromising the quality of at least one of the other objectives. In multiobjective optimization, Pareto solutions in the objective space form a surface called the *Pareto frontier*.

In real-life design, a decision maker is usually able to analyze only a very limited number of solutions for trade-off analysis. Therefore, they should be provided with an evenly enough distributed set of solutions which are representative of the entire Pareto frontier. In addition, there may exist challenges that make the generation of a well spread set of solutions difficult such as discontinuities in the Pareto frontier, non-uniform density of feasible solutions, non-convexity and non-linearity of objective functions and the constraints.

In the literature, there are mainly two categories of methods to approximate the Pareto frontier. A category corresponds to the stochastic algorithms where a population of points is evolved in each problem generation using random operators (e.g. mutation and crossover) (Venter and Haftka 2010; Deb 2001). The second category is represented by classical approaches which are the focus of this paper.

In classical multiobjective optimization, there exist several methods that can efficiently obtain Pareto solutions.

The weighted sum method and constraint-based algorithms are among the most used approaches. Often, these methods either cannot provide an evenly distributed set of solutions or fail to capture part of the Pareto frontier as indicated in Marler and Arora (2004).

The well-known approaches, which are capable of generating the whole Pareto frontier and explained below, include the Normal Boundary Intersection (NBI) (Das and Dennis 1998), the Physical Programming (PP) (Messac and Mattson 2002), the Normalized Normal Constraint (NNC) (Messac et al. 2003), the Directed Search Domain (DSD) (Erfani and Utyuzhnikov 2010; Utyuzhnikov et al. 2009; Utyuzhnikov 2010) and the Successive Pareto Optimization (SPO) (Mueller-Gritschneder et al. 2009) methods. Each of these algorithms exploits the *anchor points* which are the minima of each objective in the objective space.

All aforementioned methods employ different scalarization approaches. In particular, NBI and NNC share a close strategy for Pareto generation by introducing a reduction constraint. In NBI, a set of rays from the reference points and orthogonal to the utopia hyperplane are introduced to determine the local feasible space. The distance from the feasible boundary is taken as the objective function, and optimization along these rays provides an even generation of solutions. However, the equality constraint as the reduction criterion may not be easily satisfied in a general context. To address this problem, the NNC improves computational stability by reducing the design space. The reduction is made by using an inequality constraint defined by a plane and its orthogonal vector. Messac et al. (2003) demonstrated that in practice the NNC method is less likely to generate non-Pareto and locally Pareto solutions than the NBI approach.

Siddiqui et al. (2012) provide a modification to the original NBI algorithm by sacrificing some accuracy in problems with many variables and/or constraints. Although test examples show the strength of the proposed method, when compared to the original NBI algorithm, it becomes computationally expensive for higher dimension problems.

From a different perspective, the DSD introduces a search domain based on the affine transformation of objectives and searches for the solution within each domain. To guarantee a well distributed Pareto set, DSD evenly spreads local search domains so that they can be easily controlled in the objective space. As reported by Utyuzhnikov et al. (2009), on some test cases the DSD method outperforms the NNC and NBI with regards to the computational burden and quality of solutions. In addition, it has been shown that on some challenging test cases DSD is able to generate evenly distributed solutions whereas NNC and NBI may fail (Erfani and Utyuzhnikov 2010). However, due to introduction of new coordinate systems and matrix manipulations, the method may be difficult to implement.

In higher dimensional cases, as recognized in Das and Dennis (1998), the above procedures may not capture the entire Pareto frontier if the orthogonal projection of the Pareto surface onto the utopia hyperplane exceeds the utopia polygon. The NNC, DSD and SPO techniques each provide different strategies to cover the entire Pareto frontier. SPO implements a sequential Pareto generation technique to approximate the boundary of the Pareto surface and evenly distribute reference points on the appropriate hyperplane not extending beyond the Pareto boundary. Further optimization for each reference point in SPO is similar to that of NBI. This strategy works well for proposed test cases, but requires assumptions that may be violated in real-world applications. Alternatively, for the NNC method, the utopia hyperplane is extended to cover the whole Pareto surface. This extension is implemented via two naive optimization problems. The nadir point is also exploited to determine the extent of the Pareto surface. The DSD method applies a different technique for identifying points on the Pareto frontier. The local search domain along the edges of the utopia polygon is rotated towards the outer part of the plane in order to capture a solution in the peripheral area.

Both the SPO and NNC methods may fail to generate the Pareto frontier when anchor points coincide. This leads to degeneration of the utopia plane to a lower dimension and, hence, loss of the normal vector to the utopia plane. Meanwhile, as is shown in Erfani and Utyuzhnikov (2010), DSD can utilize the rotation technique and continue the search until the entire Pareto surface is obtained. However, this may lead to some difficulties due to the extent of rotation in higher dimension problems.

The goal of this paper is to introduce two modifications to the DSD method to make its implementation easier. First, we change the formulation of the single objective optimization problem in a search domain and replace it by a simpler one. This allows us to avoid the transformation between coordinate systems. Next, we eliminate the rotation strategy and introduce a geometrical technique to define a new modified utopia hyperplane for the optimization process. The latter task is simple to be implemented and, contrary to the NNC, it does not rely on further optimization formulations.

The rest of the paper is organized as follows. Section 2 provides a general overview of the multiobjective optimization problem. In addition, the structure of the DSD method is reformulated and its main steps are explained. The new modifications to the DSD algorithm are described in Section 3 where we introduce the modified DSD method called DSD-II. Then, in Section 4, the method is tested on some challenging and well-studied numerical problems as well as an engineering design problem. Section 5 deals with some further remarks including the sensitivity of the method to the algorithm parameters. Finally, the algorithm is summarized in Section 6.

## 2 Technical preliminaries

In the following, bold font is used to distinguish vectors from scalars.

### 2.1 Multiobjective optimization

A generic form of a multiobjective optimization problem is given by

$$\begin{aligned} &Min \;\; \boldsymbol{F}(\boldsymbol{x}) = [f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots, f_{ns}(\boldsymbol{x})], \\ &subject\,to \;\; \boldsymbol{x} \in \mathcal{D}^*, \end{aligned} \tag{1}$$

where $f_i$ $(i = 1, \ldots, ns)$ are the objective functions and $\mathcal{D}^*$ is the *feasible space* defined by the problem constraints. Usually, the corresponding minimum with respect to all objective function is located outside $\mathcal{D}^*$. Therefore, we look for a set of solutions that are called *Pareto* optimal solutions based on the following definition (Marler and Arora 2004; Erfani and Utyuzhnikov 2010):

A point $\boldsymbol{x}^*$ is called Pareto optimal solution for (1) if and only if there does not exist $\boldsymbol{x} \in \mathcal{D}^*$ satisfying $f_i(\boldsymbol{x}) \leq f_i(\boldsymbol{x}^*)$ for all $i = 1, \ldots, ns$ and $f_j(\boldsymbol{x}) < f_j(\boldsymbol{x}^*)$ for at least one $j$. The vector $\boldsymbol{F}(\boldsymbol{x^*})$ is then called a non-dominated or Pareto point. The set of all Pareto points forms the *Pareto frontier*.

This boundary provides the best possible trade-off solutions to the multiobjective optimization problem (1).

The *anchor* point for *each* objective function is obtained by optimizing the problem (1) for each objective separately. An $i$-th anchor point is written as

$$\boldsymbol{\mu}_i = \left[ f_1\left(\boldsymbol{x}^i\right), f_2\left(\boldsymbol{x}^i\right), \ldots, f_{ns}\left(\boldsymbol{x}^i\right)\right],$$

where $\boldsymbol{x}^i = arg \min\limits_{\boldsymbol{x}} \; f_i(\boldsymbol{x})$.

The convex region defined by the *anchor* points is called the *utopia* polygon. Further, the *utopia* point $\boldsymbol{U}^*$ is the best performance vector given by

$$\boldsymbol{U}^* = \left[ f_1\left(\boldsymbol{x}^1\right), f_2\left(\boldsymbol{x}^2\right), \ldots, f_{ns}\left(\boldsymbol{x}^{ns}\right)\right].$$

The *pseudo nadir* point $\boldsymbol{N}^*$, on the other hand, is the worst objective value of the anchor points formulated as

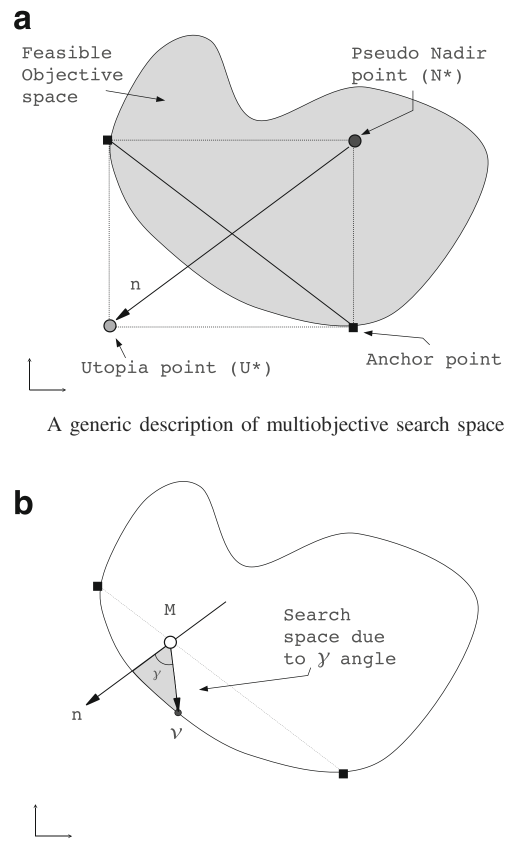
$$\boldsymbol{N}^* = [n_1, n_2, \ldots, n_{ns}],$$

where

$$n_i = max \left\{ f_i\left(\boldsymbol{x}^1\right), f_i\left(\boldsymbol{x}^2\right), \ldots, f_i\left(\boldsymbol{x}^{ns}\right)\right\}$$

These definitions are shown graphically in Fig. 1a.

In this paper we modify the DSD method to make its implementation easier. We next provide a brief description




A generic description of multiobjective search space

New shrinking strategy using the inner product angles

**Fig. 1** A generic two dimensional multiobjective problem shrinkage strategy

of the DSD method in order to facilitate comparison with DSD-II in the remainder of the paper.

### 2.2 Overview of the DSD

DSD is a method for the even generation of the Pareto frontier in a general multiobjective optimization problem with an arbitrary number of objective functions and constraints. It is based on shrinking the search domain and finding the Pareto solution in a selected area on the Pareto frontier. The algorithm contains the following steps. First, the anchor points ($\boldsymbol{\mu_i}$) are generated for each objective function. Thereafter, $\boldsymbol{\mu_i}$ are used to form the interior of the utopia hyperplane $P$ by

$$\begin{aligned} &P = \sum_{i=1}^{ns} \alpha_i \boldsymbol{\mu_i}, \\ &\sum_{i=1}^{ns} \alpha_i = 1, \\ &0 \leq \alpha_i \leq 1. \end{aligned} \tag{2}$$

By varying $\alpha_i$ uniformly, it is possible to generate evenly distributed reference points on $P$ denoted by $\boldsymbol{M}$. The points are evenly distributed if they are equally spaced from each other using appropriate metrics (Utyuzhnikov et al. 2009). For each $\boldsymbol{M}$ on $P$, the following single objective optimization problem is solved

$$Min \sum_{i=1}^{ns} f_i(\boldsymbol{x}),$$

*subject to*

$$\begin{aligned} & f_i(\boldsymbol{x}) \leq M_i, \quad (i = 1, ..., ns), \\ & \boldsymbol{x} \in \mathcal{D}^*, \end{aligned} \tag{3}$$

where $\boldsymbol{M} = [M_1, M_2, \ldots, M_{ns}]$. The above set of constraints introduces a local search domain. Solution of problem (3) for each reference point $\boldsymbol{M}$ leads to a set of Pareto solutions which are not necessarily evenly distributed. This is due to the extent of the search domain in which two neighbour reference points may share part of the other's search domain. This can lead to the same solution for neighbouring reference points (Utyuzhnikov et al. 2009). To guarantee evenly distributed solutions, a distinct search domain for each reference point is introduced. To achieve this, the search domain is reduced by introducing a new coordinate system with the origin at point $\boldsymbol{M}$. The axes of the coordinate system form a given angle with respect to a unit vector $\boldsymbol{l}$. This leads to the following modified constraint to the single objective optimization for $\boldsymbol{M}$ in (3):

$$\hat{f}_i \leq \sum_{j=1}^{ns} M_j B_{ji} \quad (i = 1, \ldots, ns), \tag{4}$$

where $A = B^{-1}$ is the transformation matrix from the Cartesian coordinate system with basis vectors $\boldsymbol{e}_j$ ($j$=1,...,$ns$) to the new local coordinate system with basis vectors $\boldsymbol{a}_j$ ($j$=1,...,$ns$), and $\hat{f}_i$ is the $i-$th objective function of the new variables. To determine the matrix $B$ in multidimensional space, the reader is referred to the original papers (Utyuzhnikov et al. 2009; Erfani and Utyuzhnikov 2010).

In the case of a non-convex boundary, it may happen that there will be no feasible solution in the search domain. In this case, the search domain is flipped to the opposite side of the utopia hyperplane to capture the points on the Pareto frontier. This is done by reversing the inequalities in problem (3). It should be noted that the search domain is flipped if no solution is attained on one side of the utopia plane. Therefore, there is no need to know the shape of the Pareto frontier *a priori*.

To overcome the deficiency discussed in Section 1, the search domain is rotated to the outer part of the utopia polygon if the point $M$ is located on its edge.

### 2.3 DSD modifications

As illustrated in its original formulation, the DSD method has been used and tested on different challenging test cases with promising results (Erfani and Utyuzhnikov 2010). However, the DSD can be improved for some applications with respect to:

- Shrinking the search domain based on transformation from Cartesian to local coordinate system may be computationally costly and hard to implement due to matrix manipulation.
- Flipping the search domain may not be efficient when the search space is highly oscillating; that is the Pareto frontier varies repeatedly from a convex to non-convex shape.
- Rotation strategy may become hard to implement specially if the anchor points coincide (Utyuzhnikov et al. 2009).

In the current paper, we address all of above issues and propose a modified version of the DSD, which we call DSD-II.

## 3 DSD-II: algorithm sketch

The major steps of DSD-II follow.

*Shrinking and flipping* In the proposed DSD-II, the shrinking constraint still exists. However, the shrinking procedure based on a coordinate system transformation is replaced as follows. A new vector $\boldsymbol{v}$

$$\boldsymbol{v} = \boldsymbol{M}_c - \boldsymbol{M}, \tag{5}$$

is introduced where $\boldsymbol{F}(\boldsymbol{x}_c) = \boldsymbol{M}_c$ with $\boldsymbol{x}_c$ being the search point in the current iteration. An additional vector $\boldsymbol{n}$, orthogonal to the utopia hyperplane $P$, is also exploited. Then, the shrinking process in problem (3) is described as

$$\arccos\left(\frac{\boldsymbol{v} \cdot \boldsymbol{n}}{\|\boldsymbol{v}\| \, \|\boldsymbol{n}\|}\right) \leq \theta, \tag{6}$$

where the left hand side is the angle between $\boldsymbol{v}$ and $\boldsymbol{n}$, and (.) is the inner product between two vectors see Fig. 1. By restricting the value for $\theta$, it is easier to shrink the search domain to generate a solution in the desired area. It is noted that the above equation is a vector-based calculation, which provides a generic expression of the search angle. Thus, it is invariant with respect to the dimension of the problem.

To overcome the computational cost inherent to the flipping strategy, at each iteration, we alter the above constraint as follows:

$$\gamma = \arccos \left| \frac{\boldsymbol{v} \cdot \boldsymbol{n}}{\|\boldsymbol{v}\| \, \|\boldsymbol{n}\|} \right| \leq \theta, \tag{7}$$

which is considered as the replacement for the existing shrinking constraint in problem (3).

Inequality (7) obviously eliminates the first two aforementioned difficulties. On one hand, the new development does not rely on the coordinate system, which eliminates the matrix manipulations, and, on the other hand, the flipping is not required. This reduces the computational cost. It should be noted that, while the DSD introduces a polyhedral as the search region, DSD-II implements a cone directed toward the Pareto frontier.

*Rotating* The rotation strategy in DSD is implemented by introducing a unit vector that is the outer normal to the edge of the utopia hyperplane. The vector can rotate from the normal direction of the utopia polygon to the direction lying outside the polygon. The rotation continues until no new solution is obtained (Utyuzhnikov et al. 2009; Erfani and Utyuzhnikov 2010). The DSD rotation is applied in two different situations. One case is when the utopia hyperplane cannot cover the whole Pareto surface. The other occasion is when degeneration of the utopia hyperplane results in the loss of the normal direction $\boldsymbol{n}$. In the latter case, there is no strategy to handle such a problem in the literature.

Although there exist some guidelines to optimize the rotation strategy for the first situation (Utyuzhnikov et al. 2009), in the second case the rotation might be cumbersome and inefficient. To improve the efficiency of the DSD method, we proceed as follows.

We redefine the direction $\boldsymbol{n}$ from the pseudo nadir point $\boldsymbol{N}^*$ to the utopia point $\boldsymbol{U}^*$. In the case where approximate bounds of the objective space are known, one may take the direction from the worst (possibly pseudo nadir) point to the best (possibly utopia) point.

$$\boldsymbol{n} = \frac{\boldsymbol{U}^* - \boldsymbol{N}^*}{\left\| \boldsymbol{U}^* - \boldsymbol{N}^* \right\|}. \tag{8}$$

A hyper-box bounding the search space is constructed as

$$C = \left[f_1^{min}, f_1^{max}\right] \times \left[f_2^{min}, f_2^{max}\right] \times \ldots \times \left[f_{ns}^{min}, f_{ns}^{max}\right], \tag{9}$$

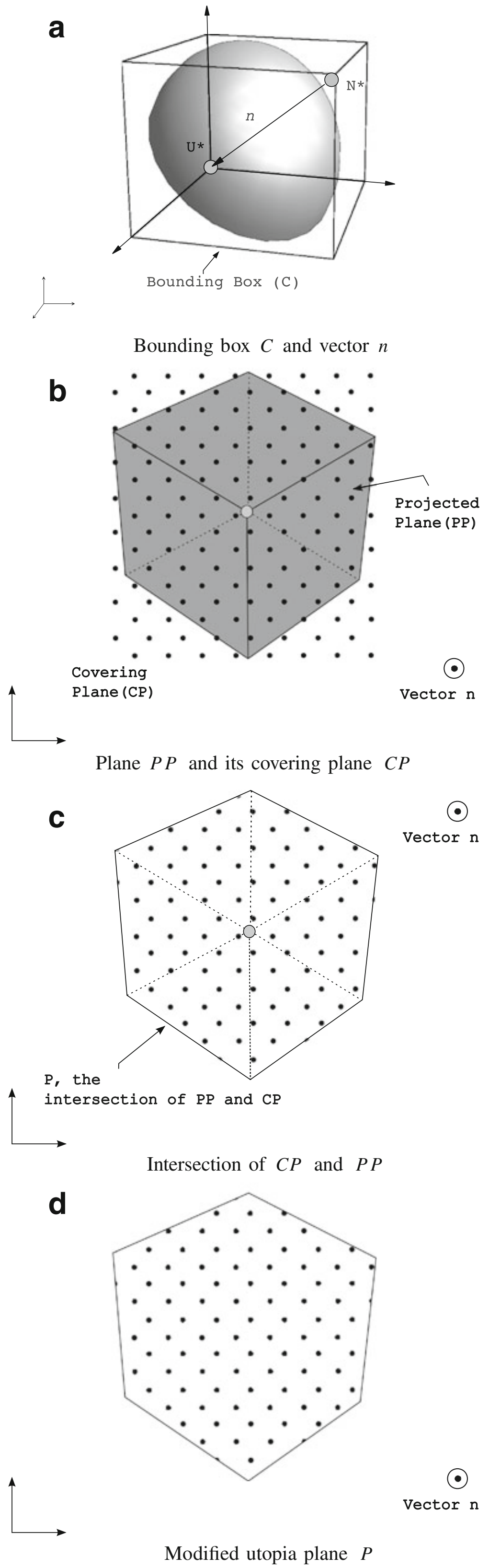



**Fig. 2** A generic description on defining the modified utopia plane $P$

Here $C$ is a bounding hyper-box with a uniform distribution of nodes within it. Next, the nodes $\boldsymbol{c} \in C$ are projected onto a hyperplane $PP$ with its normal vector defined by $\boldsymbol{n}$:

$$PP = \boldsymbol{c}^t - \boldsymbol{c}^t \boldsymbol{n} \ \boldsymbol{n}^t. \tag{10}$$

Obviously, projected points on $PP$ are not evenly distributed. To assure such a set, a covering hyperplane $CP$ with the same dimension as $PP$ is constructed by the grid containing uniformly distributed points. Then,

$$P = CP \cap PP \tag{11}$$

is a hyperplane with evenly distributed points $\boldsymbol{M}$. The schematic procedure for this step can be found in Fig. 2.

Given the above amendments to the DSD method, we present the DSD-II in Algorithm 1, in which the flipping and rotation strategies are no longer required.

**Algorithm 1** DSD-II Algorithm

**1.** Find the anchor points.
**2.** Find the modified utopia hyperplane $P = CP \cap PP$
**3.** Generate evenly distributed reference points $\boldsymbol{M}$ on $P$
**for all** $\boldsymbol{M} \in P$ **do**
**4.** Solve the following problem:

$$\begin{aligned} &Min \quad \sum_{i=1}^{ns} f_i(\boldsymbol{x}), \\ &s.t. \quad \gamma \leq \theta, \\ &\qquad \boldsymbol{x} \in \mathcal{D}^*. \end{aligned}$$

**end for**
**5.** Apply a filtering algorithm to eliminate dominated points from the given set of Pareto solutions.

It is noted that the filtering approach presented in the step 5 of the algorithm aims to remove the local Pareto solutions (Erfani and Utyuzhnikov 2010). The filtering is based on the domination concept illustrated in Pareto definition in Section 2 that returns a subset of Pareto points for which none will be dominated by any other. Furthermore, in the implementation of both DSD and DSD-II, we use the summation over all the objective functions as the aggregate objective function (AOF) for single objective optimization subproblems.

## 4 Simulation results

In this section, we demonstrate the performance of the DSD-II on test cases and compare the results with those of DSD. Test problems are chosen from a number of past studies in this area and present various challenges in multiobjective optimization. For further illustration we investigate the performance of the approach as applied to an engineering design problem.

### 4.1 Performance measures

To evaluate the diversity of the solutions along the Pareto frontier, we use the coefficient of evenness $E$ as suggested in Utyuzhnikov et al. (2009).

First, introduce the Riemann metric given by

$$d\boldsymbol{r}^2 = \sum_{i=1}^{ns-1} \sum_{j=1}^{ns-1} g_{ij} d\boldsymbol{x}^i d\boldsymbol{x}^j, \tag{12}$$

where $\{\boldsymbol{x}^i\}$ $(i = 1, \ldots, ns - 1)$ is a coordinate system on the Pareto surface and $g$ is its metric tensor. Then, the coefficient $E$ defined by

$$E = \frac{max_i \ min_j \ r_{ij}}{min_i \ min_j \ r_{ij}}, \quad i \neq j, \forall i = 1, \ldots, number\ of\ Pareto\ points, \tag{13}$$

represents the ratio between the maximal possible distance from a Pareto point and another nearest point and the minimal one. The symbol $r_{ij}$ denotes the distance between Pareto solutions $i$ and $j$ $(i \neq j)$ in metric (12). In the case

**Table 1** Comparison between DSD and DSD-II on $E$ (evenness), $Ob$ (the number of objective function evaluations) and $Ti$ (computing time in seconds)

| | DSD | | | DSD-II | | |
|---|---|---|---|---|---|---|
| | $E$ | $Ob$ | $Ti$(s) | $E$ | $Ob$ | $Ti$(s) |
| Spiral | 1.19 | 810 | 2.2 | 1.21 | **728** | **1.0** |
| | (0.11) | (20.36) | (0.37) | (0.14) | (19.31) | (0.44) |
| ZDT3 | 1.22 | 930 | 3.0 | 1.31 | **800** | **2.0** |
| | (0.31) | (18.61) | (0.17) | (0.22) | (17.10) | (0.21) |
| DTLZ2 | 1.45 | 2347 | 5.6 | 1.38 | **1982** | **3.3** |
| | (0.10) | (25.04) | (0.11) | (0.13) | (19.38) | (0.15) |
| DTLZ7 | 1.43 | 5770 | 7.2 | 1.39 | **5290** | **5.2** |
| | (0.09) | (21.44) | (0.08) | (0.12) | (20.81) | (0.17) |
| DTLZ5 | 1.42 | 2270 | 4.2 | 1.53 | **1804** | **3.2** |
| | (0.13) | (20.32) | (0.29) | (0.10) | (21.73) | (0.20) |

Values in the brackets are the variance of the performance over 30 runs. The best metric value between the two methods is highlighted by the **bold** font (the difference is statistically significant if ANOVA test's $p$-value is less than 0.05) of a completely even set of solutions, we have $E = 1$. In addition to evenness, we report the number of objective functions evaluations $Ob$ for the sake of comparison with the DSD method as well as the computational time (in seconds) required to solve the problem ($Ti$). In all test cases the number of decision variables, $m$, equals 10 and the subproblems are solved using a gradient-based algorithm. Further, the problems are solved 30 times to account for different random starting points for each subproblem. The results in Table 1 are the mean value of the total runs and the figures show the best performance among 30 runs. An ANOVA test is run to investigate whether the difference between the DSD and DSD-II algorithms on the performance metrics is significant. We distinguish between the performance metrics values if the corresponding ANOVA test p-value is less than 0.05. Therefore, the results that are statistically significant show a difference between the performance of the algorithms on the test samples.

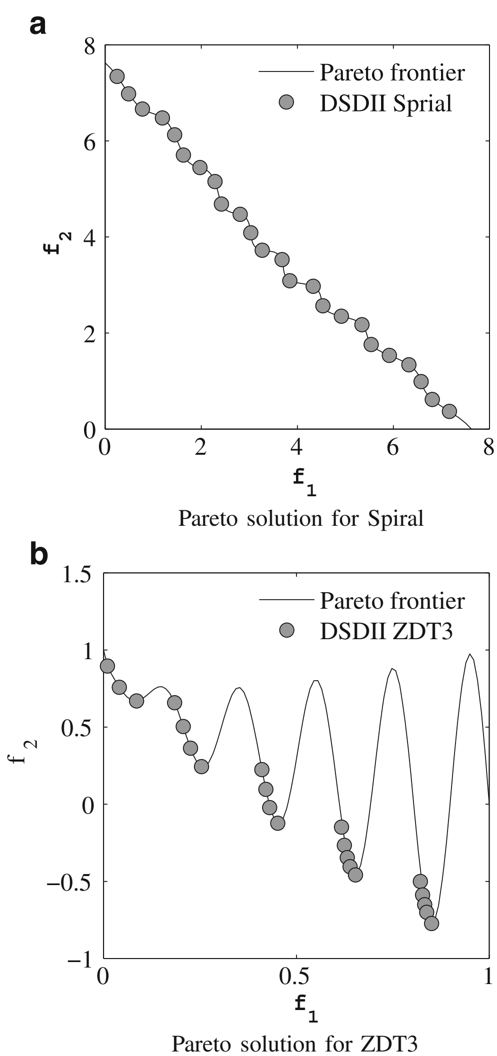



**Fig. 3** Results of DSD-II on two dimensional test problems

## 4.2 Numerical test problems

*Spiral* This test case (Branke et al. 2004) examines the efficiency of the method in obtaining the solution with a highly oscillating Pareto surface as follows

$$\begin{aligned}&Min\ (f_1(x),\ f_2(x)),\\&s.t.\\&\quad 0 \le x_i \le 1\ \ (i = 1, \ldots, m),\end{aligned}$$

*where*

$$\begin{aligned}f_1(\boldsymbol{x}) &= g(\boldsymbol{x})r(x_1)sin(\pi x(1)/2),\\f_2(\boldsymbol{x}) &= g(\boldsymbol{x})r(x_1)cos(\pi x(1)/2),\\G(\boldsymbol{x}) &= 1 + \frac{9}{m-1}\sum_{i=2}^{m} x_i,\\r(x_1) &= 5 + 10(x_1 - 0.5)^2 + \frac{1}{8}cos(2\pi k x_1)),\end{aligned}$$

The results in Table 1 show that due to fewer objective function evaluations, DSD-II solves the problem faster than the original version for the set of 25 solutions.

*ZDT3* This test case is proposed in Deb (2001) and is formulated as follows:

$$\begin{aligned}&Min\ (f_1(\boldsymbol{x}),\ f_2(\boldsymbol{x})),\\&s.t.\\&\quad 0 \le x_i \le 1\ \ (i = 1, \ldots, m),\end{aligned}$$

*where*

$$\begin{aligned}f_1(\boldsymbol{x}) &= x_1,\\f_2(\boldsymbol{x}) &= G(\boldsymbol{x})\left(1 - \sqrt{\frac{x_1}{G(\boldsymbol{x})}} - \frac{x_1}{G(\boldsymbol{x})}sin(10\pi x_1)\right),\\G(\boldsymbol{x}) &= 1 + \frac{9}{m-1}\sum_{i=2}^{m} x_i^2,\end{aligned}$$

As can be seen in Fig. 3, the problem has a discontinuous Pareto frontier. Due to the number of disconnected frontiers, the DSD flipping strategy is handled in some part of the search region if the solution is not attained on one side of the utopia line. The high computational cost of this flipping is obvious when the performance of DSD is compared with that of DSD-II. In Table 1, the two algorithms find a uniform spread of non-dominated solutions indicated by the $E$s. However, the set of solutions obtained using DSD-II requires fewer objective function evaluations (for 25 reference points). It should be noted that some local solutions are removed using the filtering strategy in Section 3.

*DTLZ2* As reported in Utyuzhnikov et al. (2009), NBI and NNC methods generate a number of redundant solutions for this three-dimensional test case

$$Min\ (f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), f_3(\boldsymbol{x})),$$
$$s.t.$$
$$0 \le x_i \le 1 \ \ (i = 1, ..., m),$$

*where*

$$f_1(\boldsymbol{x}) = (1 + G(\boldsymbol{x})) \cos(x_1\pi/2) \cos(x_2\pi/2),$$
$$f_2(\boldsymbol{x}) = (1 + G(\boldsymbol{x})) \cos(x_1\pi/2) \sin(x_2\pi/2),$$
$$f_3(\boldsymbol{x}) = (1 + G(\boldsymbol{x})) \sin(x_1\pi/2),$$
$$G(\boldsymbol{x}) = \sum_{i=3}^{m}(x_i - 0.5)^2.$$

As the Pareto surface is not convex, the number of objective function evaluations are smaller in DSD-II as compared to DSD. Furthermore, due to the extension of the Pareto surface projected onto the utopia plane, the rotation strategy is required by DSD to capture all 47 Pareto solutions (See Fig. 4).

*DTLZ7* In the following test case proposed by Deb et al. (2005), half the Pareto surface cannot be covered by the utopia hyperplane.

$$Min\ (f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), f_3(\boldsymbol{x})),$$
$$s.t.$$
$$0 \le x_i \le 1 \ \ (i = 1, ..., m),$$

*where*

$$f_1(\boldsymbol{x}) = x(1),$$
$$f_2(\boldsymbol{x}) = x(2),$$
$$f_3(\boldsymbol{x}) = (1 + G(\boldsymbol{x}))H(\boldsymbol{F}, G),$$
$$G(\boldsymbol{x}) = 1 + \frac{9}{8}\sum_{i=3}^{m} x_i,$$
$$H(\boldsymbol{F}, G) = 3 - \sum_{i=1}^{2}\left(\frac{f_i}{1 + G(x)}(1 + sin(3\pi f_i))\right).$$

Moreover, the problem has four disconnected Pareto-optimal regions in the search space. Although the results do not indicate any significant change from that of the DSD method for the set of 50 reference points, the DSD-II avoids the use of the rotation strategy. Moreover, the discontinuity of the Pareto frontier requires flipping the search domain in DSD while DSD-II does not need this. The local solutions in both approaches are filtered out from the set of Pareto solutions at the end of the optimization procedure (See Fig. 4).

*DTLZ5* This test problem is reported in Deb et al. (2005) as follows

$$Min\ (f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), f_3(\boldsymbol{x})),$$
$$s.t.$$
$$0 \le x_i \le 1 \ \ (i = 1, ..., m),$$

*where*

$$f_1(\boldsymbol{x}) = (1 + g(x_3)) \cos(\theta_1) \cos(\theta_2),$$
$$f_2(\boldsymbol{x}) = (1 + g(x_3)) \cos(\theta_1) \sin(\theta_2),$$
$$f_3(\boldsymbol{x}) = (1 + g(x_3)) \sin(\theta_1),$$
$$G(\boldsymbol{x}) = \sum_{i=3}^{m}(x_i - 0.5)^2,$$
$$\theta_1 = \frac{\pi}{2}(x_1),$$
$$\theta_2 = \frac{\pi}{4(1 + G(\boldsymbol{x}))}(1 + 2G(x)x_2).$$

This test case is challenging for all deterministic methods. It introduces a situation where the number of anchor points is smaller than the number of objective functions. Therefore, NNC and NBI fail on this test case. As reported in Erfani and Utyuzhnikov (2010), DSD successfully tackles this task due to its rotation and flipping strategy which comes with a cost of the greater number of objective function evaluations in comparison to the DSD-II (see Table 1). In both DSD and DSD-II, 50 reference points are considered and some local solutions are generated and removed by the filtering approach (See Fig. 4).

### 4.3 Engineering design problem

*I-Beam design* This problem is taken from Yang et al. (2002). The goal is to find the dimensions of the concrete I-Beam which satisfy geometric and strength constraints. The objectives are simultaneous minimization of $f_1$ cross-sectional area of the beam and $f_2$ the static deflection of the I-Beam considering the orthogonal and cross sectional forces of $P = 600$ kN and $Q = 50$ kN, respectively (Fig. 5). The problem is formulated as follows:

$$Min\ f_1 = 2x_2x_4 + x_3(x_1 - 2x_4),$$
$$Min\ f_2 = \frac{PL^3}{48EI},$$
$$s.t.$$
$$\frac{M_y}{Z_y} + \frac{M_z}{Z_z} \le \sigma_a,$$
$$10 \le x_1 \le 80, \ \ 10 \le x_2 \le 50,$$
$$0.9 \le x_3 \le 5, \ \ 0.9 \le x_4 \le 5,$$

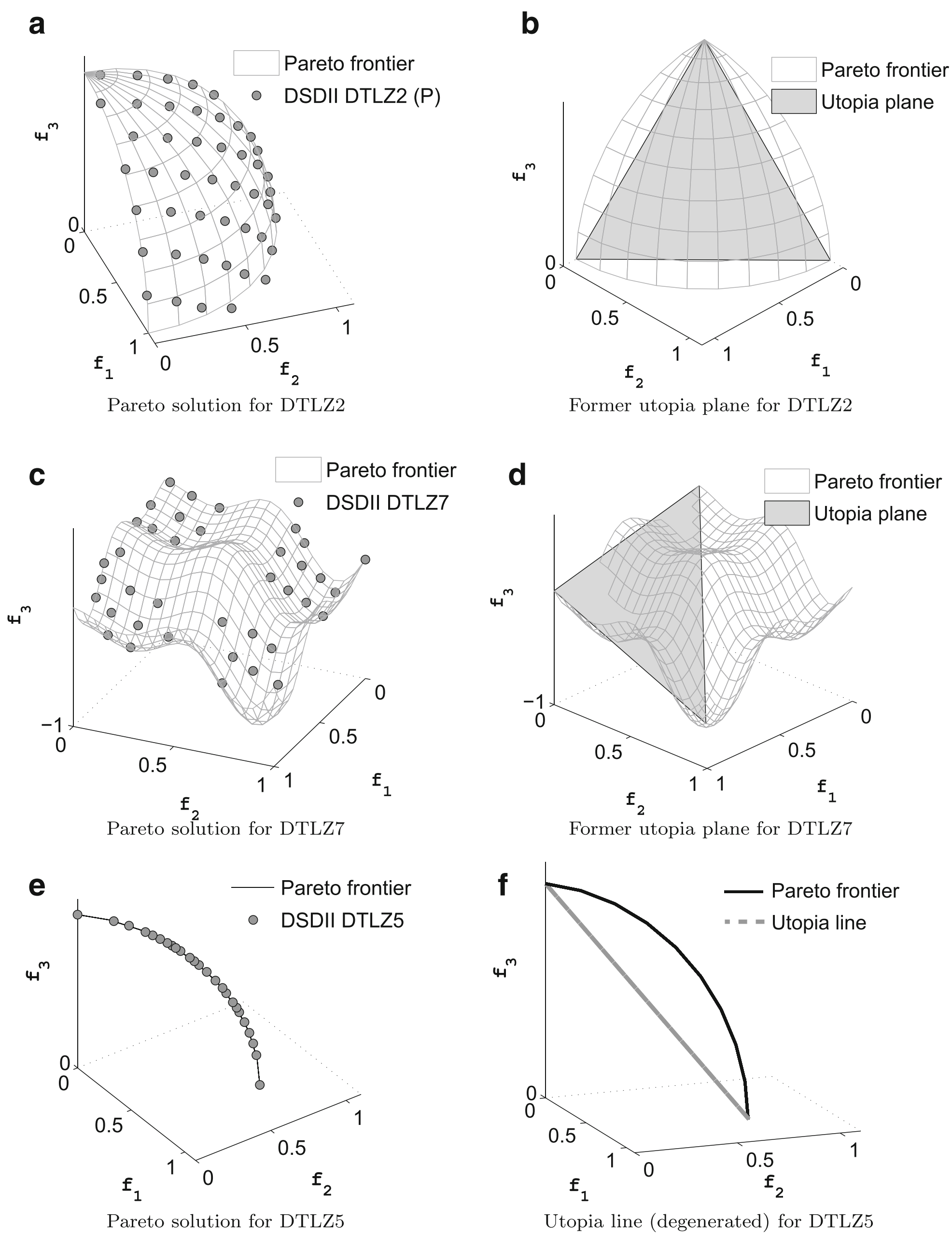



**Fig. 4** Results of DSD-II on three dimensional test problems

*where*

$$I=\frac{1}{12}\Big[x_3(x_1-2x_4)^3+2x_2x_4\Big(4x_4^2+3x_1(x_1-2x_4)\Big)\Big],$$

$$M_y=0.25PL,\quad M_z=0.25QL,$$

$$Z_y=\frac{1}{6x_1}\Big[x_3(x_1-x_4)^3+2x_2x_4\Big(4x_4^2+3x_1(x_1-2x_4)\Big)\Big],$$

$$Z_z=\frac{1}{6x_2}\left[(x_1-x_4)x_3^3+2x_4x_2^3\right],$$

$$E=2\times 10^4 kN/cm^2,\ \sigma_a=16kN/cm^2.$$

Figure 6 shows the performance of both DSD and DSD-II on this problem. DSD neglects the left hand-side of the Pareto frontier on this problem due to the search direction $\boldsymbol{l}$. In DSD, $\boldsymbol{l}$ is usually taken as the normal to the utopia line, as explained in Section 2.2. Since the search space is highly skewed, the solution procedure becomes highly sensitive. Even a small change in $\boldsymbol{l}$ leads to a different solution. For DSD, the adversely scaled objective space makes the left part of the Pareto frontier almost parallel to $\boldsymbol{l}$ and, hence, the solutions are poorly captured in this area. Meanwhile, since DSD-II performs the shrinkage based on the direction from $\boldsymbol{U}^*$ to $\boldsymbol{N}^*$, (which inherent to the skewness), it does not face these difficulties during Pareto generation.

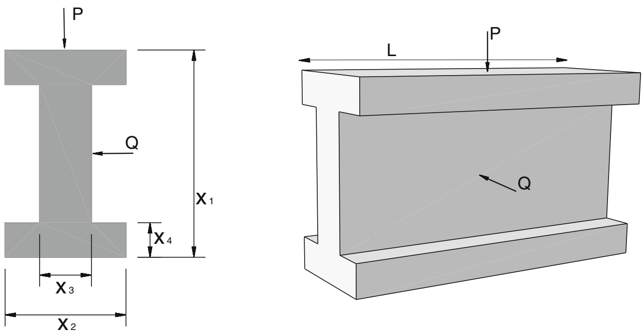



**Fig. 5** Concrete I-Beam design problem

## 5 Further consideration

### 5.1 Extra points on the modified utopia hyperplane

One can introduce a test problem in which some of the reference points generated on the modified utopia plane $P$ lead to exploring a part of the space where no feasible solution exists. This exploration occurs if the plane $P$ exceeds the projection of the Pareto surface. Therefore, it is not necessary to solve the sub-optimization problems for those points $\boldsymbol{M}$ outside the Pareto-surface projection onto $P$. To identify those unnecessary points $\boldsymbol{M}$, we need to investigate whether there exists any feasible point inside the search cone corresponding to point $\boldsymbol{M}$. This can be done by solving a naive optimization problem in which the objective function is a constant value subject to the original and conic constraints. If there is no solution, we discard the point $\boldsymbol{M}$ from further consideration. An alternative approach is to use a larger angle $\theta$ associated with the shrinkage for points near the edges of $P$. In doing so, a larger region is included in the search domain. However, this creates another difficulty due to the extent of the search space which may lead to the same solution for different search cones.

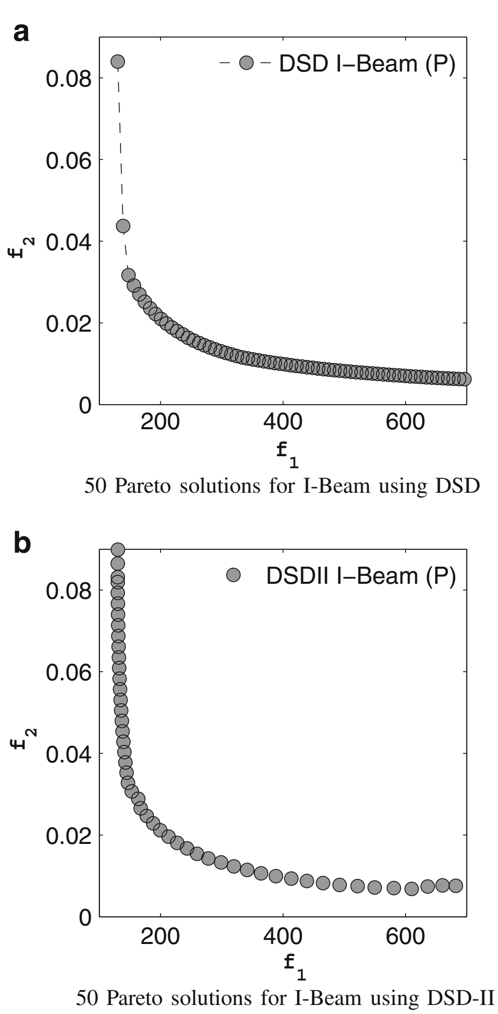



**Fig. 6** Pareto solutions for I-Beam

### 5.2 Scalability in the case of large number of objective functions

In multiobjective optimization problems the scalability to any number of decision variables and objectives is an important issue. However, as each subproblem in DSD-II is a scalar optimization problem, there is no difficulty with the method framework. To demonstrate this, the DTLZ2 test case is used to investigate DSD-II's ability to scale up its performance with four and eight objective functions. From Deb et al. (2005), it is known that Pareto optimal solutions must satisfy

$$\sum_{i=1}^{ns} f_i^{*2} = 1,$$

in objective space. For both the DTLZ2-4D and DTLZ-8D test problems, one can see in Fig. 7 that for each solution the square value of each objective function sums to one. Table 2 shows the evenness and the time required to solve the problem in each test case. As before, the solution of the problems are computed for 30 times with a randomly chosen starting point for each subproblem.

It should be noted that while scalability is a challenge in population-based algorithms, this is not the case for multiobjective scalarization methods. This is because a single-objective optimization problem is solved in order to find each Pareto solution. Therefore, in the case of a good representative set of reference points on the utopia hyperplane, the number of function evaluations only depends on the scalarization method in a multiobjective algorithm. The AOF may require as many objective function evaluations as the dimension of the problem.

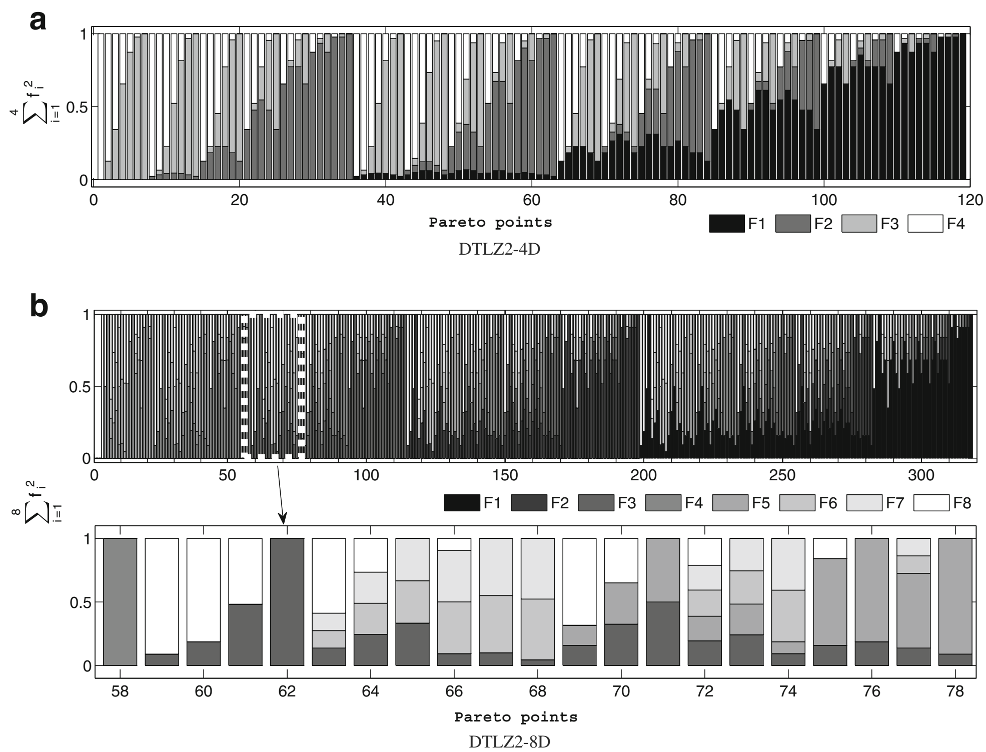



**Fig. 7** **a**, **b**. For each solution point the summation over square values of objective functions equals one. **b** Twenty Pareto optimal solution values are magnified showing the contribution of the square value of each objective function for each Pareto point in the summation

### 5.3 $\theta$ setting in the DSD-II

The variable $\theta$ is the only control parameter in DSD-II and defines the extent of shrinkage of each individual search domain. It plays a critical role in problems containing a highly skewed search space. The curvature of the Pareto surface may demand more shrinkage to guarantee one distinct solution in each search cone. While the Pareto frontier may bulk in the center of the utopia plane, it can be flat farther away and near the anchor points. For a given $\theta$, one finds that the method may work well for $\boldsymbol{M}$ situated in the center of the utopia plane. However, the shrunk space may be too large in a relatively flat area near the anchor points. The flat area appears if the Pareto frontier intersects the utopia hyperplane acutely. To alleviate this problem, either a smaller angle is used at the initial search stage for all points $\boldsymbol{M}$ or an adaptive shrinkage may be considered. Compared to the former, the latter is recommended because it is more robust due to a small search space for those points $\boldsymbol{M}$ which can work reasonably well with larger angles of shrinkage. The adaptive shrinking can be introduced by

$$\hat{\theta} = \frac{\theta}{\left\| \boldsymbol{U}^* - \boldsymbol{M} \right\|_2},$$

where $\|\,\|$ calculates the distance of $\boldsymbol{M}$ from the utopia point $\boldsymbol{U}^*$. In this case, the farther $\boldsymbol{M}$ is situated, the smaller the shrinkage is. The adaptive angle $\hat{\theta}$ works in a similar fashion as the scaling strategy where the maximal and minimal values of each objective are used to scale up the objective prior to numerical calculation. $\hat{\theta}$ assures that the proportional extent of shrinkage with respect to the distant of the reference point is used.

To test the adaptive shrinkage, we used the following test problem

*ZDT1-a* The original form of this test case is given in Deb (2001). Here, we have replaced $f_1$ by $100 f_1$ to introduce an undesirable severe skewing of the search space. The problem is given by

$$Min\ (f_1(\boldsymbol{x}),\ f_2(\boldsymbol{x})),$$
$$s.t.\ \ 0 \le x_i \le 1\ \ (i = 1, ..., m),$$

*where*

$$G(\boldsymbol{x}) = 1 + \frac{9}{m-1} \sum_{i=3}^{m} x_i^2,$$
$$f_1(\boldsymbol{x}) = 100 x_1,$$
$$f_2(\boldsymbol{x}) = G(\boldsymbol{x}) \left( 1 - \sqrt{\frac{x_1}{G(\boldsymbol{x})}} \right),$$

**Table 2** DSD-II performance on higher dimension problem: $E$ (evenness) and $Ti$ (real time spent in seconds)

| | $E$ | $Ti$(s) |
|---|---|---|
| DTLZ2-4D | 1.41 | 9.1 |
| DTLZ2-8D | 1.44 | 21.3 |

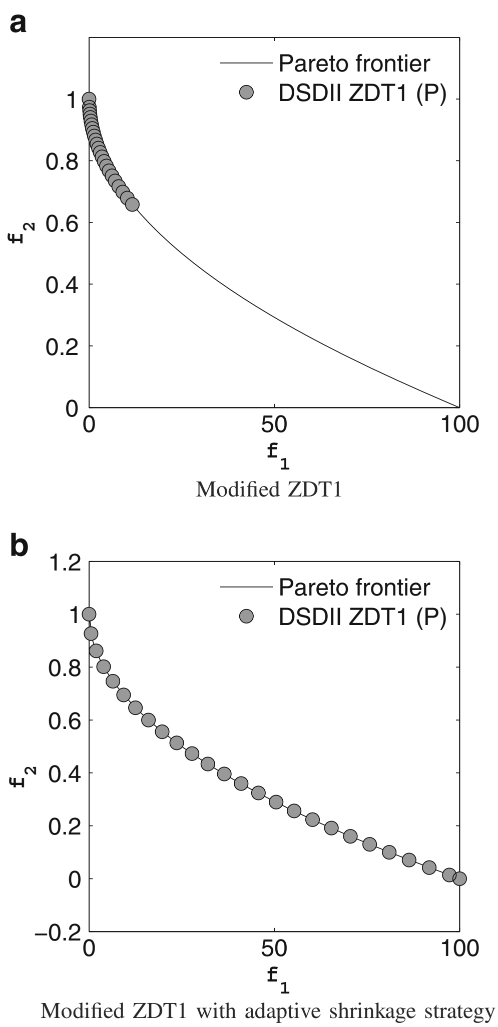



**Fig. 8** Pareto solutions for skewed ZDT1 using DSD-II

Adversely dispersed $ZDT1 - a$'s objective function space is shown in Fig. 8. Using the proposed shrinkage strategy, it is visually evident in Fig. 8 that the Pareto frontier is evenly generated ((0, 0) is the utopia point). It should be noted that the proposed adaptive shrinkage is an alternative remedy to the pre-scaling of the objective space before the optimization procedure.

## 6 Conclusion

A new DSD-II algorithm is proposed. By introducing an additional constraint in the shrinking strategy, we have modified the shrinking procedure used in the original DSD method. The modification leads to the elimination of the flipping strategy in the algorithm. In addition, instead of the rotation strategy, we have considered a modified utopia hyperplane which can cover the entire Pareto frontier. The approach proves to be computationally efficient. The method has been tested on different well known test cases and results have been compared with those of the DSD method. The comparison demonstrates that DSD-II may reduce the computational cost for finding a set of Pareto solutions. Meanwhile, the evenness of the generated solutions remains on the same level for both methods. Considering the fact that the DSD-II is easier to implement, one can conclude that both approaches (DSD and DSD-II) can be used interchangeably. For further consideration, a naive optimization is introduced to eliminate those reference points on the utopia hyperplane that lead to unfeasible solutions. As a result, the efficiency of the method is further increased. In addition, the scalability of the method to problems with many objectives has been evaluated. We have also studied the sensitivity of the DSD-II to shrinking the angle $\theta$. Empirically, it is demonstrated that for adversely scaled objective functions, the choice of $\theta$ affects the performance of the method. Therefore, an adaptive shrinkage strategy based on the distance metric is introduced. On a modified disparately scaled test case, the proposed strategy shows a clear advantage over the original one. As a future work, the existing classical methods may be compared to study the strengths and weaknesses of each of the individual methods.

**Acknowledgments** The authors are grateful to the unknown referees for their very useful remarks which allowed us to improve the quality of the paper. First author would like to thank Lucia Nigri who helped proofread part of the paper.

## References

Branke J, Deb K, Dierolf H, Osswald M (2004) Finding knees in multi-objective optimization. In: Parallel problem solving from Nature-PPSN VIII. Springer, pp 722–731

Das I, Dennis J (1998) Normal-boundary intersection: a new method for generating the Pareto surface in nonlinear multicriteria optimization problems. SIAM J Optim 8:631

Deb K (2001) Multi-objective optimization using evolutionary algorithms. Wiley, New York

Deb K, Thiele L, Laumanns M, Zitzler E (2005) Scalable test problems for evolutionary multiobjective optimization. In: Evolutionary multiobjective optimization. pp 105–145

Erfani T, Utyuzhnikov S (2010) Directed search domain: a method for even generation of the Pareto frontier in multiobjective optimization. Eng Optim 43(5):467–484

Marler R, Arora J (2004) Survey of multi-objective optimization methods for engineering. Struct Multidiscip Optim 26(6):369–395

Messac A, Mattson C (2002) Generating well-distributed sets of Pareto points for engineering design using physical programming. Optim Eng 3(4):431–450

Messac A, Ismail-Yahaya A, Mattson C (2003) The normalized normal constraint method for generating the Pareto frontier. Struct Multidiscip Optim 25(2):86–98

Mueller-Gritschneder D, Graeb H, Schlichtmann U (2009) A successive approach to compute the bounded Pareto front of practical multiobjective optimization problems. SIAM J Optim 20:915

Siddiqui S, Azarm S, Gabriel S (2012) On improving normal boundary intersection method for generation of Pareto frontier. Structural and Multidisciplinary Optimization, pp 1–14

Utyuzhnikov S (2010) Multi-objective optimization: quasi-even generation of pareto frontier and its local approximation. In: Varela J, Acuna S (eds) Handbook of optimization theory: decision analysis and application. Nova Science Publisher

Utyuzhnikov S, Fantini P, Guenov M (2009) A method for generating a well-distributed Pareto set in nonlinear multiobjective optimization. J Comput Appl Math 223(2):820–841

Venter G, Haftka R (2010) Constrained particle swarm optimization using a bi-objective formulation. Struct Multidiscip Optim 40(1):65–76

Yang B, Yeun Y, Ruy W (2002) Managing approximation models in multiobjective optimization. Struct Multidiscip Optim 24(2):141–156