RESEARCH PAPER

A modified rotation strategy for directed search domain algorithm in multiobjective engineering optimization

Kaiqiang Wang[1,2] · **Sergey V. Utyuzhnikov**[1,3]

Received: 2 October 2016 / Revised: 28 June 2017 / Accepted: 20 July 2017 / Published online: 23 August 2017

Abstract In the real-life multiobjective optimization, it is often required to generate a well-distributed Pareto set. Only a few methods are capable of tackling such a problem in a quite general formulation. The Directed Search Domain method (DSD) proved to be efficient and, therefore, attracted much attention. In this paper, two modifications to the rotation strategy of DSD algorithm are proposed. The first modification is meant to improve its computational efficiency. The second modification is to enhance the evenness of the Pareto set with a number of additional Pareto points. These points are obtained according to some specific rotation angles calculated in a particular way. The modified approach is verified on several test cases with three objectives, including an engineering case. The proposed algorithm is compared with both the original DSD and DSD-II algorithms. It is shown that the new approach can maintain the distribution of the Pareto set at a high level with a relatively low computational cost.

Keywords Rotation strategy · Directed search domain · Multiobjective optimization · Pareto set · Engineering optimization

1 Introduction

Engineering design and optimization is an active research filed. A real-life engineering design and optimization problem usually includes a variety of contradictory objectives such as high performance, low cost, long life and manufacturability. These objectives should be taken into consideration simultaneously in the optimization process. Generally speaking, the solution to such a problem is usually not unique. Instead, it represents a possible trade-off between the different objectives and cannot be improved without deterioration of at least one of the objectives and violation of the constraints. This leads to the notion of a Pareto solution (Miettinen 1999). Each Pareto point in the objective space represents a solution of the multiobjective optimization (MOO), and forms a set called the Pareto set. In general, a sufficient number of Pareto points are needed to represent the entire Pareto frontier in the objective space. However, in practice, the decision maker (DM) can only select a few possible Pareto points among the Pareto set according to additional requirements. In such a context, an even distribution of Pareto points can provide the DM with good visualization of the Pareto frontier, and substantially simplify the work of the DM. In particular, this can be exploited for local approximation of the Pareto frontier (Utyuzhnikov et al. 2008) and in ranking (Jaini and Utyuzhnikov 2017). Hence, it is of paramount importance to generate a well-distributed Pareto set to acquire maximum information on the Pareto surface at a minimal computational cost (Utyuzhnikov et al. 2009).

✉ Sergey V. Utyuzhnikov
s.utyuzhnikov@manchester.ac.uk

Kaiqiang Wang
kaiqiang.wang@outlook.com

[1] School of Mechanical, Aerospace and Civil Engineering, The University of Manchester, Sackville Street, Manchester M13 9PL, UK

[2] China Academy of Space Technology, Beijing, China

[3] Moscow Institute of Physics and Technology, Dolgoprudny 141700, Russia

A variety of methods have been developed to solve multi-objective optimization problems (Siddiqui et al. 2012; Yang and Deb 2013; Yang 2013; Fogue et al. 2013; Ganesan et al. 2013; Ghosh and Chakraborty 2014; Ansary and Panda 2014; Gobbi et al. 2015; Oke and Siddiqui 2015; Siddiqui and Christensen 2016; Fou-ladgar and Lotfi 2016). In the literature, there are mainly two categories of MOO methods to search for Pareto solutions and generate a Pareto set to approximate the Pareto frontier: evolutionary methods and classical methods. The evolutionary methods consider all objective functions simultaneously, and are usually based on stochastic algorithms and random operators. Some popular evolutionary MOO algorithms include the Niched Pareto Genetic Algorithm (NPGA) (Horn et al. 1994; Erickson et al. 2001), the Nondominated Sorting Genetic Algorithm (NSGA) (Srinivas and Deb 1994; Deb et al. 2002), the Strength Pareto Evolutionary Algorithm (SPEA) (Zitzler and Thiele 1999; Zitzler et al. 2001), and some algorithms using multiobjective particle swarm optimization (MOPSO) (Coello and Lechuga 2002; Hettenhausen et al. 2013; Dehuri et al. 2015). Evolutionary methods have several advantages. They are not sensitive to non-smoothness of objective functions, and can sometimes be efficient in finding a global extremum. However, in multiobjective optimization, there is no guarantee for evolutionary methods to capture an optimum solution. Meanwhile, a large number of solutions should be taken into account to generate an even set of optimal solutions. As a consequence, the evolutionary approaches are time-consuming in multiobjective optimization. Actually, in the real-life design, most of the solutions are redundant (Erfani and Utyuzhnikov 2011).

In contrast, the classical methods generally utilize deterministic algorithms. The weighted sum method and constraint-based algorithms are among the most popular approaches. However, these methods cannot guarantee the quasi-even distribution of the Pareto set. In some cases, they can only capture part of the Pareto frontier (Marler and Arora 2004). The well-known classical MOO methods, which are able to approximate the whole Pareto frontier, include the Normal-Boundary Intersection (NBI) method (Dad and Dennis 1997; 1998), the Normal constraint (NC) method (Messac et al. 2003; Messac and Mattson 2004), the Physical Programming method (Messac 1996; Messac and Mattson 2002), and the Directed Search Domain (DSD) method (Utyuzhnikov et al. 2009; Erfani and Utyuzhnikov 2011). All of these methods exploit the *anchor points* which are the minima of each objective in the objective space, as well as an aggregate objective function (AOF) (preference function).

In the NBI method, the Pareto frontier is construct-ed by the intersection of the normal to the utopia hyperplane and boundary of the feasible objective domain. The Pareto set is acquired after a single optimization problem has been solved along each line. The evenness of the Pareto set is determined by those of lines which are orthogonal to the hyperplane. Nevertheless, the NBI method seems to be not robust, because the feasible search domain is reduced to a line and a corresponding equality constraint is established . In order to solve this problem, NC method is proposed as a modification of NBI. The single optimization in the NC is based on inequality constraints, which makes the method more flexible and stable.

In the PP method, the DM experience is taken into consideration directly. The designer assigns each objective to one of the four class functions, and the optimization is based on minimization of an AOF determined by the class functions. However, the approach includes several free parameters and requires preliminary information on the location of the Pareto frontier.

The DSD method (Utyuzhnikov et al. 2009; Erfani and Utyuzhnikov 2011) can be considered as an alternative approach. It introduces the affine transformation of objectives to shrink a search domain to obtain a Pareto solution in a selected area of the objective space. By evenly controlling the search domains in the objective space, DSD is capable of capturing the entire Pareto frontier and providing a well-distributed Pareto set. Another advantage is that it does not generate redundant non-Pareto solutions, thus the optimization time can be saved. It has been shown that the DSD method outperforms the NBI and NC methods on a number of test cases with respect to the computational efficiency and the distribution of Pareto set. After the original DSD algorithm, a modified directed search domain algorithm called DSD-II is put forward to improve the optimization efficiency in Erfani et al. (2013). In DSD-II, the shrinking and rotating strategies are modified.

However, in some cases, the search domains are not distributed well enough in the objective space via the rotation strategy of DSD method. Then the evenness of the Pareto set is deteriorated. In addition, in DSD the rotation angle varies following a passive search algorithm reducing optimization efficiency. In DSD-II, the rotation strategy is not implemented. Instead, redundant points on the utopia hyperplane are generated in order to capture the whole Pareto frontier. This may lead to many redundant solutions and negatively affect the computational efficiency. In the context of these problems, the objective of this paper is to improve both the effect and efficiency of the original DSD algorithm without generating redundant solutions. The proposed approach can be considered as an alternative to DSD-II. Two modifications are proposed. One is to enhance the efficiency of searching for the Pareto solutions with the use of dichotomy in rotation strategy. The other modification is to improve the evenness of the Pareto set distribution based on some specifically calculated rotation angles.

The paper is structured as follows. In Section 2, basic concepts and definitions about multiobjective optimization problems are introduced. Section 3 provides the main principles and procedures of the DSD algorithm including DSD-II. The two new modifications to the original DSD algorithm are described in Section 4. The modified approach is evaluated in Section 5 on the basis of simulation results of three numerical cases and one engineering case. In Section 6, discussion as well as conclusions are presented.

2 Multiobjective optimization

A general multiobjective optimization (MOO) problem is described as follows:

$$\begin{aligned} &\text{Min} \quad \boldsymbol{f}(\boldsymbol{x}) = (f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), ..., f_n(\boldsymbol{x})), \\ &\text{subject to} \quad \boldsymbol{x} \in \boldsymbol{D}^*. \end{aligned} \tag{1}$$

where f_i $(i = 1, 2, ..., n)$ are objective functions, which form the objective space $\boldsymbol{Y} \subseteq \mathbb{R}^n$. The vector $\boldsymbol{x}$ represents a design variable in the decision space $\boldsymbol{D}$, and $\boldsymbol{D}^* \subseteq \boldsymbol{D}$ is a feasible space which is the set of elements satisfying all the constraints.

In general, the solution of problem (1) is not unique. Therefore, a set of solutions called the Pareto optimal set is introduced on the basis of the following definition (Miettinen 1999):

Definition 1 Pareto solution: Vector $\boldsymbol{x}^* \in \boldsymbol{D}^*$ is called a Pareto (optimal) solution to problem (1) if and only if there does not exist $\boldsymbol{x} \in \boldsymbol{D}^*$ such that $f_i(\boldsymbol{x}) \leq f_i(\boldsymbol{x}^*)$ for all $i = 1, ..., n$ and $f_j(\boldsymbol{x}) < f_j(\boldsymbol{x}^*)$ for at least one j $(1 \leq j \leq n)$.

Then, in the objective space the vector $\boldsymbol{f}(\boldsymbol{x}^*)$ represents a Pareto solution not dominated by any other feasible solution. The set of all Pareto points represents the Pareto frontier, the best trade-off solutions to the multiobjective optimization problem (1). If the above condition only holds in a vicinity of $\boldsymbol{x}^*$, then $\boldsymbol{x}^*$ is called a local Pareto solution.

Definition 2 Anchor point: In the objective space $\boldsymbol{Y}$, an anchor point μ_i represents the minimum of an ith objective function subject to all the constraints.

According to the above definition, a non-unique anchor point is sometimes obtained. Then, a lexicographic-based prioritization is introduced as follows Utyuzhnikov et al. (2009).

Definition 3 Modified anchor point: In the feasible objective space $\boldsymbol{Y}^* = \{\boldsymbol{f}(\boldsymbol{x}) \mid \boldsymbol{x} \in \boldsymbol{D}^*\}$, the anchor point μ_i of an ith objective function is determined in the circular order: min f_i, min f_{i+1}, ..., min f_n, min f_1, min f_2, ..., min f_{i-1}. Any successive minimization problem: min f_{i+m} $(1 \leq i + m \leq n, m \neq 0)$ is only to be considered on the set $\{A_{m-1}\}$, which is the solution to the previous minimization problem.

Definition 4 A hyperplane that contains all the anchor points is called the utopia hyperplane.

The above definitions are illustrated in Fig. 1.

3 Review of directed search domain (DSD) algorithm

DSD is a classical algorithm for generating a quasi-even Pareto set in multiobjective optimization problems. In Erfani et al. (2013), a modified directed search domain algorithm called DSD-II is proposed to further improve the optimization efficiency.

3.1 The original DSD algorithm

The main idea of the original DSD algorithm is to utilize transformation technique to shrink the search domain and seek the Pareto solution in the resulting shrunk area. To address a multiobjective optimization problem containing n objective functions as explained in (1) by DSD, the following steps are needed.

Step 1. The objective functions should be scaled if necessary.

Step 2. Search for the anchor points or modified anchor points.

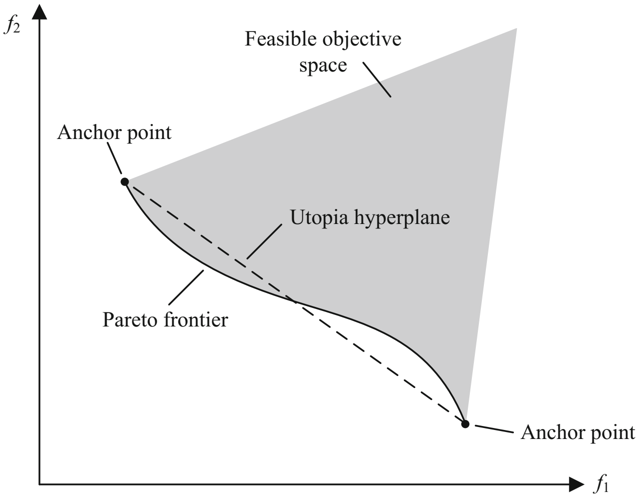

Fig. 1 Basic definitions of MOO in objective space

Step 3. Form the interior of the utopia hyperplane P by

$$P = \sum_{i=1}^{n} \alpha_i \mu_i,$$
$$\sum_{i=1}^{n} \alpha_i = 1,$$
$$0 \le \alpha_i \le 1 \ (i = 1, ..., n), \tag{2}$$

where α_i are coefficients for generating reference points denoted by $\mathcal{M}$ in the next step.

Step 4. Generate a set of evenly distributed reference points $\mathcal{M}$ on the utopia hyperplane by varying α_i in (2).

Step 5. Shrink the search domain. First, for each reference point $\boldsymbol{M} = (M_1, ..., M_n) \in \mathcal{M}$, the following single optimization problem is formulated similar to the physical programming (Messac 1996; Messac and Mattson 2002):

$$\begin{aligned} \text{Min} \quad & \sum_{i=1}^{n} f_i(\boldsymbol{x}), \\ \text{s.t.} \quad & f_i(\boldsymbol{x}) \le M_i, \\ & \boldsymbol{x} \in \boldsymbol{D}^*. \end{aligned} \tag{3}$$

In problem (3), the constraints $f_i(\boldsymbol{x}) \le M_i$ are the search domain formed on the basis of reference point $\boldsymbol{M}$. Therefore, different reference points result in different search domains, and such different single objective optimization problems are expected to have different solutions. However, two nearby reference points may share a part of the other search domains. Then, redundant solutions can be generated and deteriorate the evenness of the Pareto set. To overcome this problem in DSD algorithm, each search domain is shrunk to be distinct. To achieve this, a new coordinate system with the origin at a reference point $\boldsymbol{M}$ is introduced to reduce the size of the search domain, and the axes of the coordinate system form a given angle with respect to a unit vector. Finally, the constraint to the single objective optimization problem (3) is modified as

$$\hat{f}_i(x) \le \sum_{j=1}^{n} M_j B_{ji} \ (i = 1, ..., n), \tag{4}$$

where $A = B^{-1}$ is the transformation matrix from the Cartesian coordinate system to the new local coordinate system, and $\hat{f}_i(x) = \sum_{j=1}^{n} f_j(x) B_{ji}$ is the ith objective function based on the new local coordinate system. To calculate the matrix B, the reader is referred to the original paper (Utyuzhnikov et al. 2009).

Step 6. Solve the new single objective optimization based on the shrunk search domain for each reference point $\boldsymbol{M} = (M_1, ..., M_n) \in \mathcal{M}$:

$$\begin{aligned} \text{Min} \quad & \sum_{i=1}^{n} f_i(\boldsymbol{x}), \\ \text{s.t.} \quad & \hat{f}_i(x) \le \sum_{j=1}^{n} M_j B_{ji}, \\ & \boldsymbol{x} \in \boldsymbol{D}^*. \end{aligned} \tag{5}$$

Sometimes there may not be any feasible solution found in the search domain, when the boundary of the feasible objective space is non-convex. In such a context, the search domain should be flipped to the opposite side of the utopia hyperplane in order to search the points on the Pareto frontier by reversing the inequalities in optimization problem (5) as

$$\hat{f}_i(x) \ge \sum_{j=1}^{n} M_j B_{ji} \ (i = 1, ..., n). \tag{6}$$

Step 7. Rotate the search domain if necessary. In some high-dimensional cases, it may happen that the orthogonal projection of the utopia hyperplane does not fully cover the entire Pareto surface. In order to capture uncovered region, the search domain should be rotated if the reference point $\boldsymbol{M}$ is located on the edge of the hyperplane. Here, definitions of an edge reference point and edge Pareto point are introduced.

Definition 5 Edge reference point: In the objective space $\boldsymbol{Y}$, the reference point located on the edge of the utopia hyperplane is defined as an edge reference point.

Definition 6 Edge Pareto point: In the objective space $\boldsymbol{Y}$, the Pareto point located on the edge of the Pareto surface is called an edge Pareto point.

Generally speaking, for each edge reference point, an edge Pareto point can be found with the rotation strategy. To realize this, a vector $\boldsymbol{s}$ is introduced as the outer normal to the edge of the polygon on the utopia hyperplane (Fig. 2) first.

$$\boldsymbol{s}_i = \frac{\boldsymbol{b}_{i-1} + \beta_i \boldsymbol{b}_i}{|\boldsymbol{b}_{i-1} + \beta_i \boldsymbol{b}_i|}, \ \beta_i = -\frac{(\boldsymbol{b}_{i-1}, \ \boldsymbol{b}_i)}{(\boldsymbol{b}_i, \ \boldsymbol{b}_i)}, \tag{7}$$

where $\boldsymbol{b}_i = \mu_i - \mu_{i-1}$ are the boundaries of the polygon. Thus, a new vector $\boldsymbol{l}$ is defined by

$$\boldsymbol{l} = -\cos\theta \boldsymbol{n}_h + \sin\theta \boldsymbol{s}_i, \ 0 < \theta < \pi/2, \tag{8}$$

where $\boldsymbol{n}_h$ is the normal vector to the utopia hyperplane. By changing θ from 0 to $\pi/2$, the rotation of the search domain

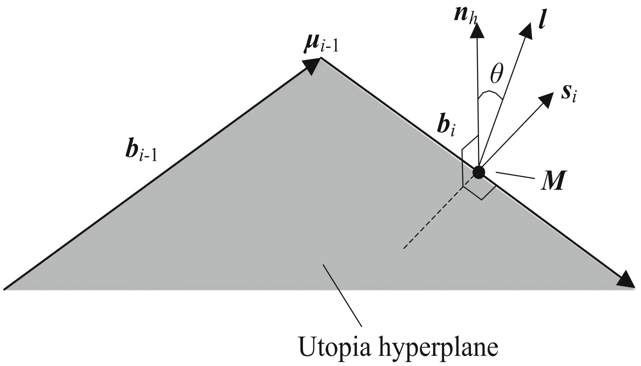

Fig. 2 Rotation of vector $\boldsymbol{l}$

is from the normal direction of the hyperplane to the direction lying on the outer part of the polygon (Utyuzhnikov et al. 2009). The rotation continues until no new solution is captured.

It can be shown (Utyuzhnikov et al. 2009) that

$$A_0 = \frac{\sin\gamma_c}{\sin\gamma_0} I + \frac{\sin(\gamma_0 - \gamma_c)}{\sin\gamma_0} E \tag{9}$$

where γ_c is the angle used to shrink the search domain, $\cos\gamma_0 = 1/\sqrt{n}$, I is the unit matrix, and all elements of the matrix E are unities.

In the rotation strategy, a rotation matrix R is needed to map the unit vector $\boldsymbol{l}_0 = (l_0, l_0, ..., l_0)^{\mathrm{T}}$ onto $\boldsymbol{l}$:

$$R\boldsymbol{l}_0 = \boldsymbol{l} \tag{10}$$

The matrix R is calculated by

$$R = DTD^{\mathrm{T}}, \tag{11}$$

where D is an orthogonal matrix obtained by the Gram-Schmidt orthogonalization procedure

$$D = Gram - Schmidt(\boldsymbol{l}_0, \boldsymbol{l}, \boldsymbol{e}_3, ..., \boldsymbol{e}_n). \tag{12}$$

In (12), $\boldsymbol{e}_k = (\delta_{1k}, \delta_{2k}, ..., \delta_{nk})$ where $\delta_{ij} = 1$ if $i = j$ and $\delta_{ij} = 0$ if $i \neq j$ $(i, j = 1, ..., n)$. In (11), matrix T describes an elementary rotation in the hyperplane created by the vectors $\boldsymbol{l}_0$ and $\boldsymbol{l}$:

$$T = \begin{pmatrix} (\boldsymbol{l}_0, \boldsymbol{l}) & -\sqrt{1-(\boldsymbol{l}_0, \boldsymbol{l})^2} & 0 \cdots 0 \\ \sqrt{1-(\boldsymbol{l}_0, \boldsymbol{l})^2} & (\boldsymbol{l}_0, \boldsymbol{l}) & 0 \cdots 0 \\ 0 & 0 & 1 \cdots 0 \\ \vdots & \vdots & \vdots \ddots 0 \\ 0 & 0 & 0 \ldots 1 \end{pmatrix} \tag{13}$$

Thus, the matrix A on Step 5 above should be updated to matrix A_r:

$$A_r = A_0 R^{\mathrm{T}}. \tag{14}$$

Meanwhile, the matrix B in the single-objective optimization problem on Step 4 and 5 should be updated to matrix B_r:

$$B_r = A_r^{-1}. \tag{15}$$

Step 8. Eliminate local Pareto solutions by a filtering procedure.

For details of steps and strategies in DSD algorithm, the reader is referred to the original paper (Utyuzhnikov et al. 2009; Erfani and Utyuzhnikov 2011).

3.2 DSD-II

As a modified DSD algorithm, DSD-II mainly improves the shrinking procedure by a vector-based shrinking strategy. A new vector $\boldsymbol{v}$ is defined as

$$\boldsymbol{v} = \boldsymbol{M}_c - \boldsymbol{M}, \tag{16}$$

where $\boldsymbol{M}_c = \boldsymbol{f}(\boldsymbol{x})$ with $\boldsymbol{x}$ to be searched in the shrunk search domain. Thereafter, the new shrinking inequality is given as

$$\gamma = \arccos\left|\frac{\boldsymbol{v}\cdot\boldsymbol{n}_h}{\|\boldsymbol{v}\|\|\boldsymbol{n}_h\|}\right| \leq \delta. \tag{17}$$

By setting the value of δ, the search domain based on each reference point is shrunk. Thus, two advantages are brought for improvement of computational efficiency. On the one hand, the shrinking calculation process is simplified so that the complex coordinate system transformation is avoided. On the other hand, the flipping procedure is eliminated since the new shrunk search domain already covers both sides of the utopia hyperplane.

As to the rotation strategy, DSD-II proposes modified utopia hyperplane to remove the rotating procedure for further enhancing the efficiency. However, this is achieved by some increasing the computational time in comparison with the original DSD algorithm. The reader is referred to the paper (Erfani et al. 2013) for details of modified utopia hyperplane generation.

4 A modified rotation strategy: DSD-III

On the basis of the original DSD algorithm, two modifications are proposed to strengthen the efficiency as well as to generate a better distributed Pareto set:

1. With the use of dichotomy, the efficiency of searching for the edge Pareto points is enhanced.
2. Based on some specific rotation angles, the evenness of the Pareto set is improved.

The proposed approach, which is based on the original DSD algorithm with the modified rotation strategy, is called

DSD-III. This method is efficient in case the rotation strategy is needed. It is more effective since the dichotomous method is used rather than a passive search. The evenness might be better because more Pareto points are added if there are some gaps, which the original DSD method is not able to fill out well enough. It is worth noting that DSD-III can straightforwardly implement the non-flipping approach from DSD-II.

4.1 Searching for the edge Pareto points

The major steps of searching for the edge Pareto points with the use of dichotomy are illustrated as Fig. 3.

Step 1. Set the original search interval of the rotation angle θ to $[0\ \pi/2]$.

Step 2. Calculate the midpoint of the current search interval of θ:

$$\theta_{mid} = \frac{\theta_l + \theta_u}{2}, \tag{18}$$

where θ_l and θ_u are the lower and upper bounds of the search interval, respectively.

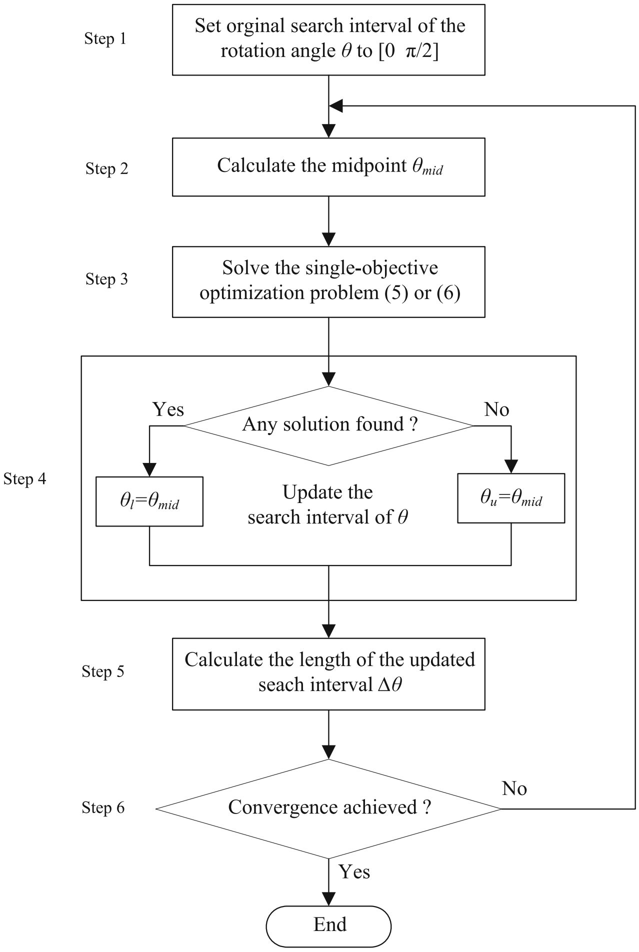
Step 1
Set orginal search interval of the rotation angle θ to [0 π/2]
Step 2
Calculate the midpoint θ_{mid}
Step 3
Solve the single-objective optimization problem (5) or (6)
Step 4
Yes
Any solution found ?
No
θ_l=θ_{mid}
Update the search interval of θ
θ_u=θ_{mid}
Step 5
Calculate the length of the updated seach interval $\Delta\theta$
Step 6
Convergence achieved ?
No
Yes
End

Fig. 3 Major steps of searching for the edg e Pareto points

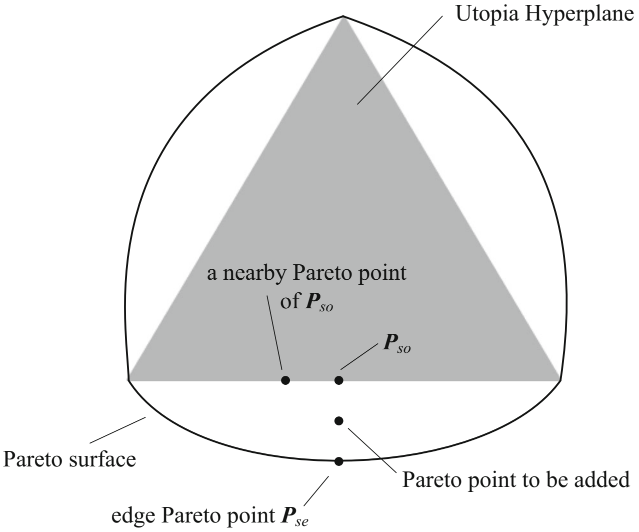

Fig. 4 Orthogonal projection of the hyperplane and Pareto surface

Step 3. Calculate the matrix B_r based on θ_{mid} according to (7)–(15). Then, solve the single-objective optimization problem (5) or (6) if the search domain needs to be flipped.

Step 4. Update the search interval of θ. If no feasible solution is found, then the upper bound of the search interval should be updated as $\theta_u = \theta_{mid}$. Otherwise, the lower bound should be updated as $\theta_l = \theta_{mid}$.

Step 5. Calculate the length of the updated search interval $\Delta\theta = \theta_u - \theta_l$.

Step 6. Analyze the convergence. If the interval length $\Delta\theta$ is smaller than expected, then the process of the edge Pareto points searching is completed, and the edge rotation angle is $\theta_e = \theta_l$. Next, the edge Pareto point based on the current reference point can be found according to the rotation angle θ_e. Otherwise, another iteration is required.

4.2 Improve the evenness of the Pareto set

Based on an edge reference point, a Pareto point can be acquired without rotation first, denoted as $\boldsymbol{P}_{so}$ (Fig. 4). Then, with the use of rotation strategy, an edge Pareto point $\boldsymbol{P}_{se}$ can be obtained. Thus, another definition is introduced.

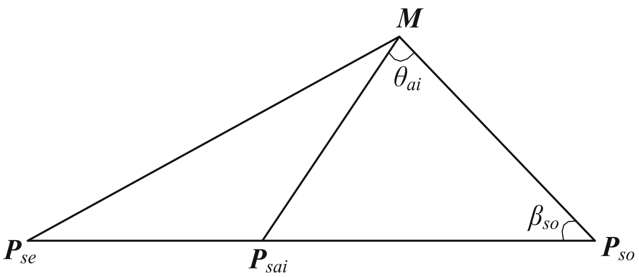

Fig. 5 Geometrical relationship among the reference point and some Pareto points

Fig. 6 Pareto points of Case 1 by DSD ($\Delta\alpha_i = 0.05$)

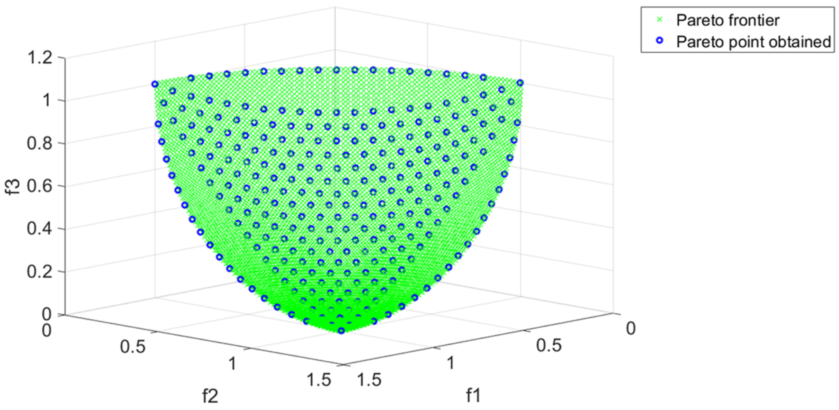

Fig. 7 Pareto points of Case 1 by DSD-II ($\Delta\alpha_i = 0.05$)

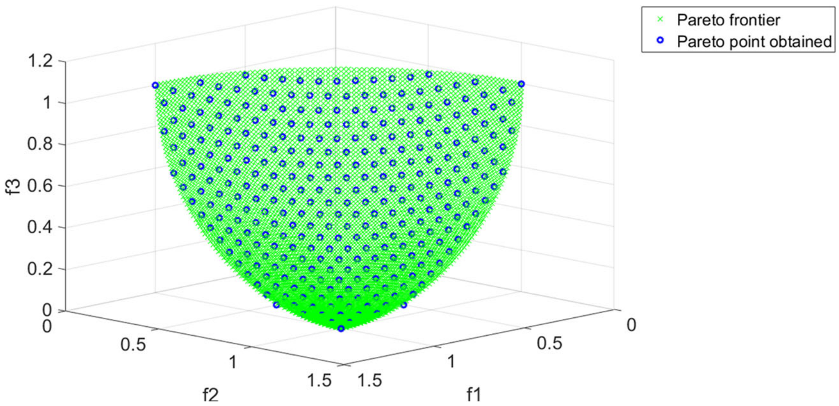

Fig. 8 Pareto points of Case 1 by DSD-III ($\Delta\alpha_i = 0.05$)

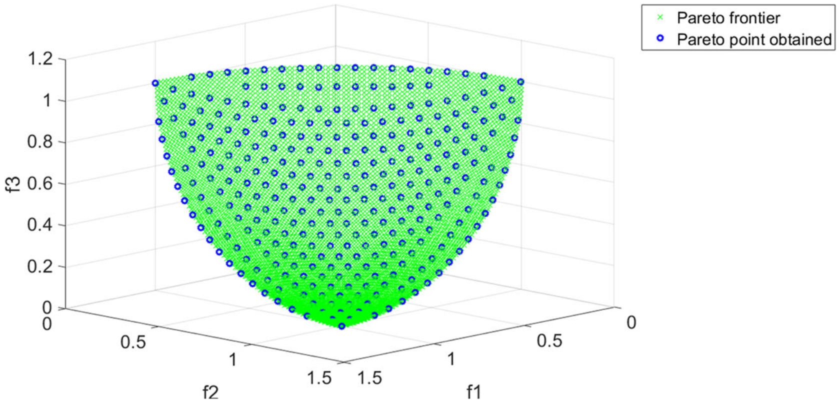

Table 1 Comparison on optimization performance in Case 1

$\Delta\alpha_i$	DSD			DSD-II			DSD-III			η_2	η_3	$\eta_{3,2}$
	E	$t(s)$	n_p	E	$t(s)$	n_p	E	$t(s)$	n_p			
0.1	1.47	34.2	87	2.44	7.31	84	1.47	4.62	87	21.4%	13.5%	63.2%
0.05	3.05	76.4	282	2.70	17.1	314	1.75	11.8	315	22.4%	15.4%	69.0%
0.025	6.77	160.2	972	2.19	57.1	1280	2.21	33.4	1206	35.6%	20.8%	58.5%
0.02	8.96	204.8	1467	2.92	84.3	1970	2.16	48.9	1863	41.2%	23.9%	58.0%
0.01	25.3	458.3	5443	2.59	345.3	7902	3.28	157.9	7335	75.3%	34.5%	45.7%

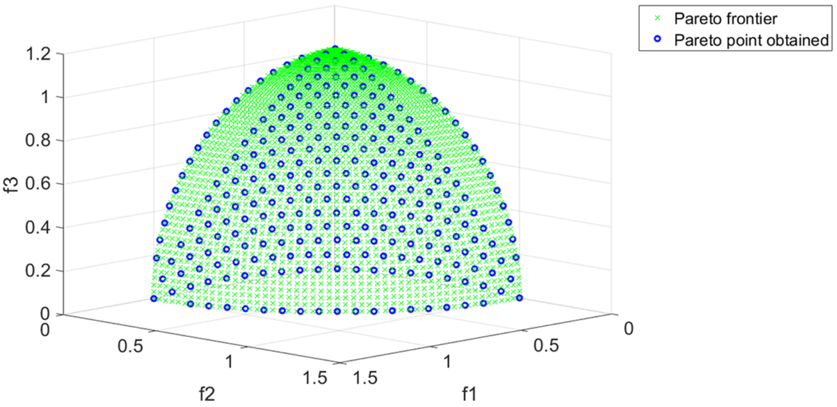

Fig. 9 Pareto points of Case 2 by DSD ($\Delta\alpha_i = 0.05$)

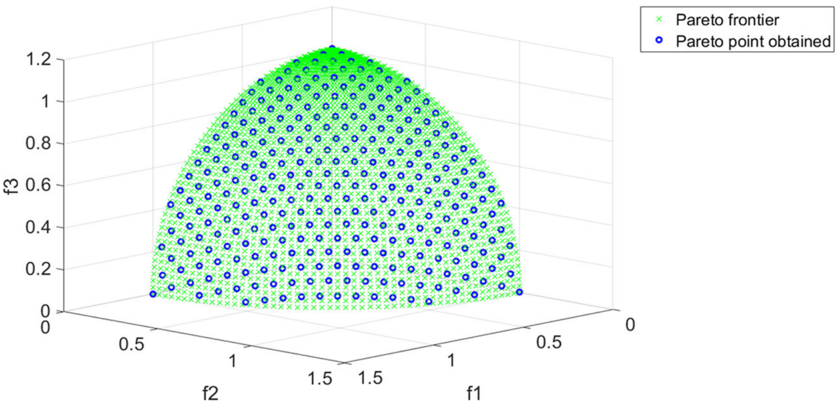

Fig. 10 Pareto points of Case 2 by DSD-II ($\Delta\alpha_i = 0.05$)

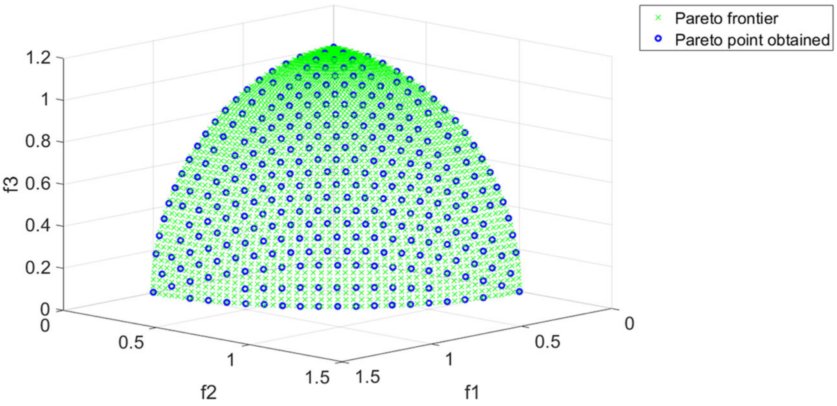

Fig. 11 Pareto points of Case 2 by DSD-III ($\Delta\alpha_i = 0.05$)

Table 2 Comparison on optimization performance in Case 2

$\Delta\alpha_i$	DSD			DSD-II			DSD-III			η_2	η_3	$\eta_{3,2}$
	E	$t(s)$	n_p	E	$t(s)$	n_p	E	$t(s)$	n_p			
0.1	1.49	45.2	87	2.44	13.7	84	1.49	18.6	87	30.3%	41.2%	136%
0.05	2.73	104.8	282	2.77	42.6	312	1.55	47.6	315	40.6%	45.4%	112%
0.025	5.46	253.0	972	2.12	151.4	1272	1.77	137.9	1191	59.8%	54.5%	91.1%
0.02	6.83	328.1	1465	2.22	219.5	1986	1.83	198.7	1851	66.9%	60.6%	90.5%
0.01	19.18	923.4	5440	2.34	711.9	7929	2.42	662.3	7267	77.1%	71.7%	93.0%

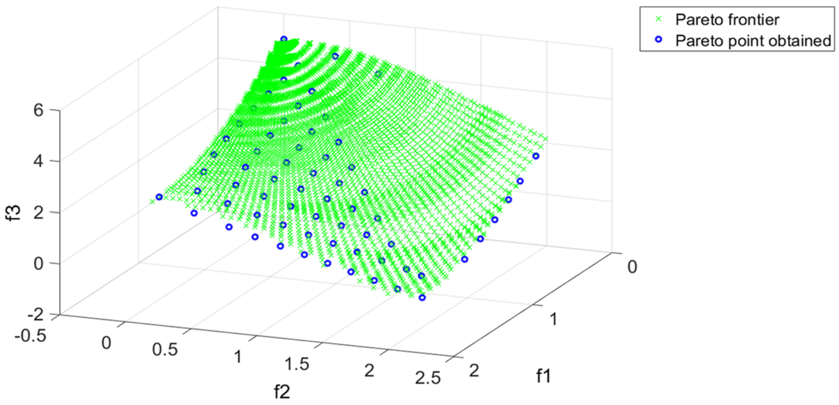

Fig. 12 Pareto points of Case 3 by DSD ($\Delta\alpha_i = 0.1$)

Definition 7 Edge point distance: For an edge Pareto point $\boldsymbol{P}_{se}$, the distance between $\boldsymbol{P}_{se}$ and the Pareto point $\boldsymbol{P}_{so}$ obtained based on the same edge reference point without rotation is called the edge point distance d_e, mathematically defined as

$$d_e = |\boldsymbol{P}_{se}\boldsymbol{P}_{so}| = \|\boldsymbol{P}_{se} - \boldsymbol{P}_{so}\|. \qquad (19)$$

In some cases, the edge point distance of an edge Pareto point is so large that the Pareto set is not quasi-distributed. In this case, more Pareto points based on the same reference point are required according to some specific rotation angles.

First of all, the number of the Pareto points to be added, denoted as n_a, should be determined as follows.

$$n_a = \left[\frac{d_e}{d_{np}\eta_d}\right], \qquad (20)$$

where d_{np} is the distance between two nearby Pareto points already obtained before rotating the search domain, assume $\boldsymbol{P}_{s1}$ and $\boldsymbol{P}_{s2}$ are a pair of nearby Pareto points, then d_{np} is calculated by

$$d_{np} = \|\boldsymbol{P}_{s1} - \boldsymbol{P}_{s2}\|, \qquad (21)$$

and η_d is a parameter. By varying the value of η_d, the number of the Pareto points to be added is adjusted to guarantee the even distribution of the Pareto set. The suggested value of η_d is from 0.75 to 0.9. It is also worth noting that n_a should be rounded to an integer.

Consider the geometrical relationship among the reference point $\boldsymbol{M}$ and some Pareto points, depicted in Fig. 5. Then, the specific rotation angles $\boldsymbol{\theta}_a$ can be determined as

$$\boldsymbol{\theta}_a = (\theta_{a1}, \theta_{a2}, ...\theta_{a(n_a-1)}),$$
$$\theta_{ai} = \arcsin(\frac{|\boldsymbol{P}_{sai}\boldsymbol{P}_{so}|}{|\boldsymbol{P}_{sai}\boldsymbol{M}|} sin\beta_{so}) \ \ (i = 1, ..., n_a - 1), \qquad (22)$$

where $|\boldsymbol{P}_{sai}\boldsymbol{P}_{so}|$ is defined as

$$|\boldsymbol{P}_{sai}\boldsymbol{P}_{so}| = \frac{i}{n_a}|\boldsymbol{P}_{se}\boldsymbol{P}_{so}| = \frac{i}{n_a}d_e, \qquad (23)$$

β_{so} is given as

$$\beta_{so} = \arccos(\frac{|\boldsymbol{P}_{so}\boldsymbol{M}|^2 + |\boldsymbol{P}_{se}\boldsymbol{P}_{so}|^2 - |\boldsymbol{P}_{se}\boldsymbol{M}|^2}{2\,|\boldsymbol{P}_{so}\boldsymbol{M}|\,|\boldsymbol{P}_{se}\boldsymbol{P}_{so}|}), \qquad (24)$$

and $|\boldsymbol{P}_{sai}\boldsymbol{M}|$ is calculated as

$$|\boldsymbol{P}_{sai}\boldsymbol{M}| = (|\boldsymbol{P}_{so}\boldsymbol{M}|^2 + |\boldsymbol{P}_{sai}\boldsymbol{P}_{so}|^2 - 2cos\beta_{so}\,|\boldsymbol{P}_{so}\boldsymbol{M}|\,|\boldsymbol{P}_{sai}\boldsymbol{P}_{so}|)^{0.5}. \qquad (25)$$

Finally, by rotating the search domain according to $\boldsymbol{\theta}_a$, the corresponding Pareto points can be acquired. With their addition it becomes possible to strengthen the evenness of the Pareto set.

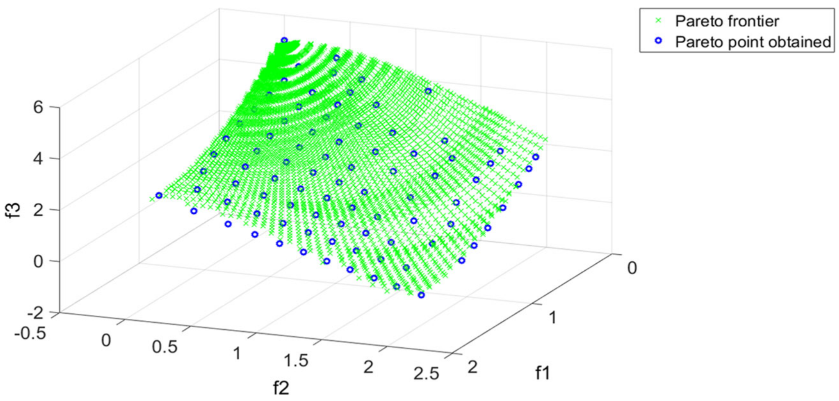

Fig. 13 Pareto points of Case 3 by DSD-III approach ($\Delta\alpha_i = 0.1$)

Table 3 Comparison on optimization performance in Case 3

$\Delta\alpha_i$	DSD			DSD-III approach			η_t
	E	$t(s)$	n_p	E	$t(s)$	n_p	
0.1	8.78	390.1	66	3.40	52.6	87	13.5%
0.05	21.86	786.9	225	6.87	101.0	317	12.8%
0.025	61.08	1634.3	781	8.47	236.3	1187	14.5%
0.02	104.81	2069.0	1181	8.22	292.8	1805	14.2%

5 Test cases

The DSD-III method is validated on four test cases, including three numerical cases and one engineering case. The results obtained by our approach are compared against the original DSD algorithm with regards to the time required for the whole optimization process and the distribution of the Pareto set.

In order to mathematically describe the evenness of the Pareto set, a coefficient of evenness needs to be defined (Utyuzhnikov et al. 2009). For an ith Pareto point $\boldsymbol{P}_s^i$ (except the anchor points) in the Pareto set, the distance vector between it and other Pareto points is given by

$$\boldsymbol{d}^i = (\|\boldsymbol{P}_s^i - \boldsymbol{P}_s^1\|, \ldots, \|\boldsymbol{P}_s^i - \boldsymbol{P}_s^{i-1}\|, \\ \|\boldsymbol{P}_s^i - \boldsymbol{P}_s^{i+1}\|, \ldots, \|\boldsymbol{P}_s^i - \boldsymbol{P}_s^{n_p}\|), \tag{26}$$

where n_p is the number of Pareto points (including n anchor points) contained in the Pareto set. Then, the effective distance vector $\boldsymbol{d}_{eff}^i$ for $\boldsymbol{P}_s^i$ is defined as

$$\boldsymbol{d}_{eff}^i = (\min(\boldsymbol{d}^i), 2\text{nd}\min(\boldsymbol{d}^i), \ldots, n\text{th}\min(\boldsymbol{d}^i)), \tag{27}$$

where nth min$(\boldsymbol{d}^i)$ is the nth smallest value of $\boldsymbol{d}^i$.

Then, the coefficient of evenness E is determined as

$$E = \frac{\max(\boldsymbol{d}_{eff}^1, \boldsymbol{d}_{eff}^2, \ldots, \boldsymbol{d}_{eff}^{n_p-n})}{\min(\boldsymbol{d}_{eff}^1, \boldsymbol{d}_{eff}^2, \ldots, \boldsymbol{d}_{eff}^{n_p-n})}. \tag{28}$$

If $E = 1$, then the Pareto set is completely even. If E increases, then the evenness of the Pareto set deteriorates.

In the whole multiobjective optimization process, MATLAB R2015a is utilized for simulation. The fmincon optimization function is used as a basis optimizer, in which the active-set algorithm is chosen as the optimization algorithm.

5.1 Numerical test cases

First, we consider three numerical multiobjective optimization cases.

5.1.1 Case 1

Case 1 is a relatively simple three-dimensional test case with a concave Pareto frontier.

$$\begin{aligned} \text{Min} \quad & \boldsymbol{f}(\boldsymbol{x}) = (x_1, x_2, x_3), \\ \text{s.t.} \quad & g_1(\boldsymbol{x}) = (x_1-1)^2 + (x_2-1)^2 + (x_3-1)^2 \le 1, \\ & 0 \le x_i \le 1 \quad (i = 1, 2, 3). \end{aligned} \tag{29}$$

The Pareto sets acquired with DSD, DSD-II, and DSD-III are depicted in Figs. 6, 7, and 8, respectively, where $\Delta\alpha_i$ is the step for changing α_i in (2). Comparison among the three algorithms on the evenness coefficient E, as well as the required computational time t, is presented in Table 1. Here, n_p is the number of Pareto points obtained. η_2 and η_3 are the ratios of the DSD-II and DSD-III computational times to that of DSD, respectively. To compare DSD-III with DSD-II, the parameter $\eta_{3,2}$ is set, which is the ratio of the DSD-III computational time to that of DSD-II.

Compared with the DSD algorithm, the DSD-III approach drastically reduces the computing time by about $65\% - 85\%$. Meanwhile, it also improves the coefficient of evenness E from $6 - 25$ to only $1.5 - 3$. Thus, a more even Pareto set is obtained by DSD-III. In DSD-III, as the step $\Delta\alpha_i$ decreases, more reference points are generated. Hence, more Pareto points are needed to provide a quasi-even distribution of the Pareto set.

Compared with DSD-II, the DSD-III approach cuts down the computational time by nearly $30\%-55\%$. The efficiency

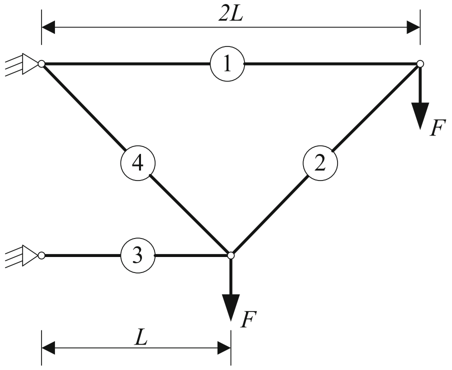

Fig. 14 4-bar truss structure

of DSD-III increases as the step $\Delta\alpha_i$ decreases. This is because more redundant solutions are generated by DSD-II. It is worth noting that the coefficient of evenness E retains on approximately the same level of about $1.5 - 3$ for both DSD-II and DSD-III.

5.1.2 Case 2

Case 2 is the three-dimensional test case $DTLZ2$ (Abraham and Jain 2005) on a concave Pareto frontier, which is a little more complex than Case 1.

$$\begin{aligned}
&\text{Min} \quad \boldsymbol{f}(\boldsymbol{x}) = (f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), f_3(\boldsymbol{x})),\\
&\text{s.t.} \quad 0 \le x_i \le 1 \quad (i = 1, 2, 3),\\
&\qquad \text{where}\\
&\qquad f_1(\boldsymbol{x}) = (1 + g(x_3)) \cos(x_1\pi/2) \cos(x_2\pi/2),\\
&\qquad f_2(\boldsymbol{x}) = (1 + g(x_3)) \cos(x_1\pi/2) \sin(x_2\pi/2),\\
&\qquad f_3(\boldsymbol{x}) = (1 + g(x_3)) \sin(x_1\pi/2),\\
&\qquad g_3(\boldsymbol{x}) = (x_3 - 0.5)^2. \qquad (30)
\end{aligned}$$

The Pareto sets generated with DSD, DSD-II, and DSD-III are shown in Figs. 9, 10 and 11, respectively. Comparison among the three methods on the optimization performance is presented in Table 2.

In Case 2, DSD-III decreases the optimization time by about $30\% - 60\%$ compared with DSD, whilst reducing the coefficient of evenness E from $5 - 19$ to less than 2.5. Similar to Case 1, the smaller $\Delta\alpha_i$ is, the better evenness of the generated Pareto set is achieved.

In this test case, when $\Delta\alpha_i$ is equal or larger than 0.05, DSD-II performs better with respect to the computational time. This is mainly because not so many redundant solutions are generated under this circumstance. However, as $\Delta\alpha_i$ drops below 0.05, DSD-III works faster. As to the coefficient of evenness, DSD-III slightly outperforms DSD-II. It improves E by about $0.4 - 1$ while both methods maintain it at a high level, from about 1.5 to 2.5.

5.1.3 Case 3

Case 3 is a three-dimensional test case with a number of constraints. The objective functions are more complex.

$$\begin{aligned}
&\text{Min} \quad \boldsymbol{f}(\boldsymbol{x}) = (f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), f_3(\boldsymbol{x})),\\
&\text{s.t.} \quad 0 \le x_i \le 2 \quad (i = 1, 2, 3),\\
&\qquad 0 \le f_1 \le 2, \quad 0 \le f_2 \le 2,\\
&\qquad \text{where}\\
&\qquad f_1(\boldsymbol{x}) = [1 + (x_3 - 0.5)^2] \cos(x_1\pi/2) \cos(x_2\pi/2),\\
&\qquad f_2(\boldsymbol{x}) = [1 + (x_3 - 0.5)^2] \cos(x_1\pi/2) \sin(x_2\pi/2),\\
&\qquad f_3(\boldsymbol{x}) = (5 - x_3) - 3\sin(f_1) + 2\cos(f_2). \qquad (31)
\end{aligned}$$

In Case 3, DSD-II turned out to be too sensitive to the initial values of the design variables. Therefore, only the Pareto sets obtained with DSD and DSD-III are shown in Figs. 12 and 13, respectively. Comparison between these two approaches on the optimization performance is presented in Table 3.

In Case 3, the DSD-III approach drastically cuts down the optimization time by about 85%, and generates a Pareto

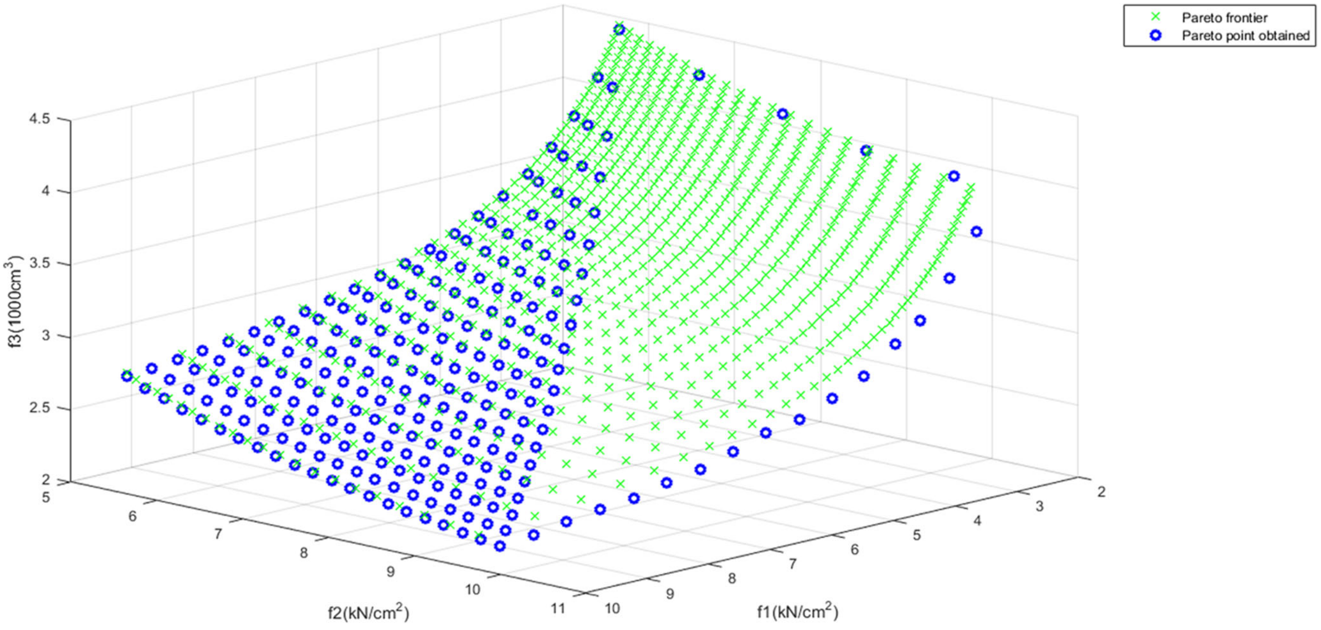

Fig. 15 Pareto points of the 4-bar problem by DSD ($\Delta\alpha_i = 0.05$)

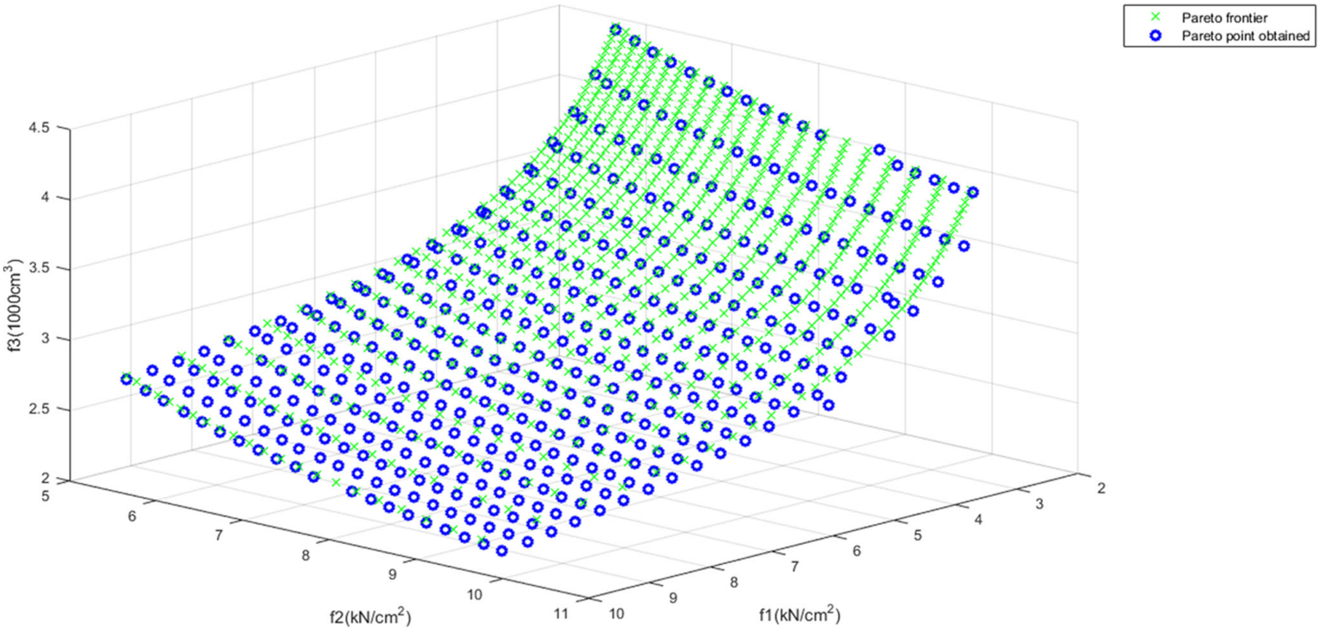

Fig. 16 Pareto points of the 4-bar problem by DSD-II ($\Delta\alpha_i = 0.05$)

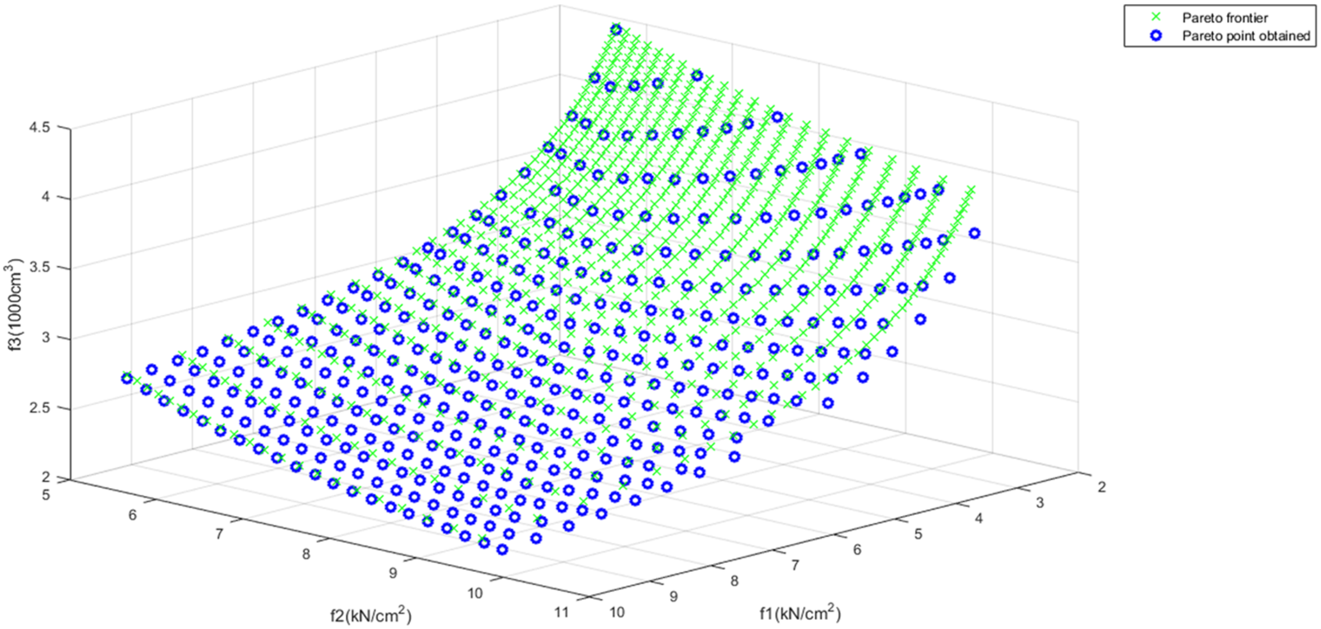

Fig. 17 Pareto points of the 4-bar problem by DSD-III ($\Delta\alpha_i = 0.05$)

Table 4 Comparison on optimization performance in the 4-bar problem

$\Delta\alpha_i$	DSD			DSD-II			DSD-III			η_2	η_3	$\eta_{3,2}$
	E	$t(s)$	n_p	E	$t(s)$	n_p	E	$t(s)$	n_p			
0.1	5.96	19.8	75	3.58	10.2	119	3.13	4.31	105	51.5%	21.8%	42.3%
0.05	12.57	43.7	249	3.67	31.2	436	4.13	13.9	391	71.4%	31.8%	44.6%
0.025	14.57	101.2	873	4.29	130.3	1667	4.17	47.2	1500	129%	46.6%	36.2%
0.02	14.86	141.0	1336	4.46	226.5	2567	4.68	69.1	2329	161%	49.0%	30.5%

set with better evenness than the original DSD algorithm at the same time.

5.2 Engineering design problem

In this case, the four-bar truss multiobjective optimization problem is analyzed (Koski 1988). The truss scheme is depicted in Fig. 14. The magnitude of the suspended load F is 10kN.

$$\begin{aligned}
\text{Min}\quad & \boldsymbol{f}(\boldsymbol{x}) = (f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), f_3(\boldsymbol{x})),\\
\text{s.t.}\quad & 0 \le A_i \le 5\text{cm}^2 \quad (i = 1, 2, 3, 4),\\
& |\sigma_i| \le 10\text{kN/cm}^2 \quad (i = 1, 2, 3, 4),\\
& \text{where}\\
& \boldsymbol{x} = (A_1, A_2, A_3, A_4),\\
& f_1(\boldsymbol{x}) = |\sigma_1|,\\
& f_2(\boldsymbol{x}) = |\sigma_4|,\\
& f_3(\boldsymbol{x}) = V.
\end{aligned} \tag{32}$$

In the above problem, $A_i (i = 1, 2, 3, 4)$ is the size of Bar i, $\sigma_i (i = 1, 2, 3, 4)$ is the stress in Bar i, and V is the volume of the structure. The objective is to design a four-bar truss with optimal stress in Bar 1 and 4, and optimal volume of the structure.

The Pareto sets obtained with DSD, DSD-II, and DSD-III are shown in Figs. 15, 16, and 17, respectively. The three approaches are compared with each other in Table 4.

In this engineering design problem, DSD-III reduces the computational time by nearly $50\% - 80\%$ compared with DSD. Furthermore, the evenness coefficient E is also significantly improved from about $6 - 15$ to $3 - 4.5$ in the meantime.

Compared with DSD-II, DSD-III largely cuts down the computational time by nearly $55\% - 70\%$. Both algorithms maintain the evenness coefficient E at the same level, from about 3 to 4.5.

It is worth noting that when $\Delta\alpha_i$ is equal to or smaller than 0.025, the computational efficiency of DSD-II is worse than that of DSD. This is due to that a fairly large number of redundant solutions are generated in this case.

6 Conclusion

A modified rotation strategy for DSD algorithm, DSD-III method, has been proposed. One of the two modifications is to search for the Pareto points located on the edge of Pareto frontier with the use of dichotomy method. Instead of changing the rotation angle step by step, the DSD-III approach can directly find out the edge Pareto point according to the rotation angle acquired by dichotomy. Thus, the whole optimization time is saved. The other modification is to insert a number of Pareto points to improve the evenness of the Pareto set. These Pareto points are obtained according to some specific rotation angles calculated in a particular way.

The new technique has been tested on different cases, including three numerical cases and one engineering case. Its performance has been compared with that of the original DSD and DSD-II algorithms. Compared with DSD, DSD-III always performs better with respect to both the computational time and the evenness of the Pareto set. As the density of the Pareto set increases, the effect of DSD-III in enhancing the evenness is heightened. In comparison with DSD, both DSD-II and DSD-III approaches can equally improve the evenness of the Pareto. As to the computational efficiency, DSD-III performs better than DSD-II in general. In particular, the DSD-III algorithm can be recommended for the problems in which the orthogonal projection of the Pareto surface onto the utopia hyperplane far exceeds the utopia polygon.

Acknowledgments The first author acknowledges support from Chinese Scholarship Council (CSC). The authors are grateful to the unknown referees for useful remarks that improved the quality of the paper.

References

Abraham A, Jain L (2005) Evolutionary multiobjective optimization. In: Evolutionary multiobjective optimization. Springer, pp 126–128

Ansary MAT, Panda G (2014) A modified quasi-newton method for vector optimization problem. Optimization 64(11):1–18

Coello CC, Lechuga M (2002) Mopso: A proposal for multiple objective particle swarm optimization. In: Proceedings of the 2002 congress on evolutionary computation, 2002. CEC'02, vol 2. IEEE, pp 1051–1056

Dad I, Dennis J (1997) A closer look at drawbacks of minimizing weighted sums of objectives for pareto set generation in multicriteria optimization problems. Struct Multidiscip Optim 14(1): 63–69

Dad I, Dennis J (1998) Normal-boundary intersection: a new method for generating the pareto surface in nonlinear multicriteria optimization problems. SIAM J Optim 8(3):631–657

Deb K, Pratap A, Agarwal S, Meyarivan T (2002) A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans Evol Comput 6(2):182–197

Dehuri S, Jagadev A, Panda M (2015) Multi-objective swarm intelligence: theoretical advances and applications, vol 592. Springer

Erfani T, Utyuzhnikov S (2011) Directed search domain: a method for even generation of the pareto frontier in multiobjective optimization. Eng Optim 43(5):467–484

Erfani T, Utyuzhnikov S, Kolo B (2013) A modified directed search domain algorithm for multiobjective engineering and design optimization. Struct Multidiscip Optim 48(6):1129–1141

Erickson M, Mayer A, Horn J (2001) The niched pareto genetic algorithm 2 applied to the design of groundwater remediation systems. In: International conference on evolutionary multi-criterion optimization. Springer, pp 681–695

Fogue M, Garrido P, Martinez FJ, Cano JC, Calafate CT, Manzoni P (2013) A novel approach for traffic accidents sanitary resource allocation based on multi-objective genetic algorithms. Expert Syst Appl Int J 40(1):323–336

Fou-ladgar N, Lotfi S (2016) A novel approach for optimization in dynamic environments based on modified cuckoo search algorithm. Soft Comput 20(7):2889–2903

Ganesan T, Vasant P, Elamvazuthi I (2013) Normal-boundary intersection based parametric multi-objective optimization of green sand mould system. J Manuf Syst 32(1):197–205

Ghosh D, Chakraborty D (2014) A new method to obtain fuzzy pareto set of fuzzy multi-criteria optimization problems. J Intell Fuzzy Syst 26(3):1223–1234

Gobbi M, Levi F, Mastinu G, Previati G (2015) On the analytical derivation of the pareto-optimal set with applications to structural design. Struct Multidiscip Optim 51(3):645–657

Hettenhausen J, Lewis A, Randall M, Kipouros T (2013) Interactive multi-objective particle swarm optimisation using decision space interaction. In: 2013 IEEE congress on evolutionary computation. IEEE, pp 3411–3418

Horn J, Nafpliotis N, Goldberg D (1994) A niched pareto genetic algorithm for multiobjective optimization. In: Proceedings of the First IEEE conference on evolutionary computation, 1994. IEEE World Congress on Computational Intelligence. IEEE, pp 82-87

Jaini N, Utyuzhnikov S (2017) Trade-off ranking method for multicriteria decision analysis. Journal of Multi-criteria Decision Analysis

Koski J (1988) Multicriteria truss optimization. In: Multicriteria optimization in engineering and in the sciences. Springer, pp 263–307

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

Messac A (1996) Physical programming effective optimization for computational design. AIAA J 34(1):149–158

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, Mattson C (2004) Normal constraint method with guarantee of even representation of complete pareto frontier. AIAA J 42(10):2101–2111

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

Miettinen K (1999) Nonlinear multiobjective optimization. Kluwer Academic, Boston

Oke O, Siddiqui S (2015) Efficient automated schematic map drawing using multiobjective mixed integer programming. Comput Oper Res 61:1–17

Siddiqui S, Christensen A (2016) Determining energy and climate market policy using multiobjective programs with equilibrium constraints. Energy 94:316–325

Siddiqui S, Azarm S, Gabriel SA (2012) On improving normal boundary intersection method for generation of pareto frontier. Struct Multidiscip Optim 46(6):839–852

Srinivas N, Deb K (1994) Multiobjective optimization using nondominated sorting in genetic algorithms. Evol Comput 2(3):221–248

Utyuzhnikov S, Maginot J, Guenov M (2008) Local pareto approximation for multi-objective optimization. Eng Optim 40(9):821–847

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

Yang XS (2013) Multiobjective firefly algorithm for continuous optimization. Eng Comput 29(2):175–184

Yang XS, Deb S (2013) Multiobjective cuckoo search for design optimization. Comput Oper Res 40(6):1616–1624

Zitzler E, Thiele L (1999) Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach. IEEE Trans Evol Comput 3(4):257–271

Zitzler E, Laumanns M, Thiele L (2001) SPEA2: Improving the strength Pareto evolutionary algorithm. Technical Report 103, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH)