Engineering Optimization

ISSN: 0305-215X (Print) 1029-0273 (Online) Journal homepage: http://www.tandfonline.com/loi/geno20

An extension of the directed search domain algorithm to bilevel optimization

Kaiqiang Wang & Sergey V. Utyuzhnikov

To cite this article: Kaiqiang Wang & Sergey V. Utyuzhnikov (2017) An extension of the directed search domain algorithm to bilevel optimization, Engineering Optimization, 49:8, 1420-1440, DOI: 10.1080/0305215X.2016.1248960

To link to this article: http://dx.doi.org/10.1080/0305215X.2016.1248960

Published online: 15 Nov 2016.

ENGINEERING OPTIMIZATION, 2017
VOL. 49, NO. 8, 1420–1440
http://dx.doi.org/10.1080/0305215X.2016.1248960

Taylor & Francis
Taylor & Francis Group

OPEN ACCESS Check for updates

An extension of the directed search domain algorithm to bilevel optimization

Kaiqiang Wang[a,b] and Sergey V. Utyuzhnikov[a,c]

[a]School of Mechanical, Aerospace and Civil Engineering, The University of Manchester, Manchester, UK; [b]China Academy of Space Technology, Beijing, China; [c]Moscow Institute of Physics and Technology, Dolgoprudny, Russia

ABSTRACT
A method is developed for generating a well-distributed Pareto set for the upper level in bilevel multiobjective optimization. The approach is based on the Directed Search Domain (DSD) algorithm, which is a classical approach for generation of a quasi-evenly distributed Pareto set in multiobjective optimization. The approach contains a double-layer optimizer designed in a specific way under the framework of the DSD method. The double-layer optimizer is based on bilevel single-objective optimization and aims to find a unique optimal Pareto solution rather than generate the whole Pareto frontier on the lower level in order to improve the optimization efficiency. The proposed bilevel DSD approach is verified on several test cases, and a relevant comparison against another classical approach is made. It is shown that the approach can generate a quasi-evenly distributed Pareto set for the upper level with relatively low time consumption.

ARTICLE HISTORY
Received 21 July 2016
Accepted 9 October 2016

KEYWORDS
Bilevel optimization; multiobjective optimization; DSD; Pareto set; double-layer optimizer

1. Introduction

In real-life engineering design and optimization, a variety of contradictory objectives such as low cost, high performance, long life and manufacturability should be taken into consideration simultaneously. In general, the solution to such a problem is usually not unique. Instead, it represents a possible trade-off between different objectives and cannot be improved without deterioration of at least one of them. This leads to the notion of a Pareto solution (see, *e.g.*, 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. Generally speaking, it is expected to have a sufficient number of Pareto points 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 this case, an even distribution of Pareto points can provide the DM a good visualization of the Pareto frontier, and substantially simplify the work of the DM. Hence, it is of paramount importance to generate a well-distributed Pareto set to acquire maximum information on the Pareto surface at minimal computational cost (Utyuzhnikov, Fantini and Guenov 2009).

Until now, there have been various categories of MOO algorithms to search for Pareto solutions and generate a well-distributed Pareto set. The Normal-Boundary Intersection (NBI) method (Dad and Dennis 1997, 1998), the Physical Programming method (Messac 1996; Messac and Mattson 2002), the Normal Constraint (NC) method (Messac, Ismail-Yahaya and Mattson 2003; Messac and Mattson 2004), and the Directed Search Domain (DSD) method (Utyuzhnikov, Fantini and Guenov 2009; Erfani and Utyuzhnikov 2011) are some examples of classical MOO algorithms. Evolutionary MOO algorithms include the Niched Pareto Genetic Algorithm (NPGA) (Horn, Nafpliotis

CONTACT Kaiqiang Wang kaiqiang.wang@postgrad.manchester.ac.uk

and Goldberg 1994; Erickson, Mayer and Horn 2001), the Non-dominated Sorting Genetic Algorithm (NSGA) (Srinivas and Deb 1994; Deb *et al.* 2002), the Strength Pareto Evolutionary Algorithm (SPEA) (Zitzler and Thiele 1999; Zitzler, Laumanns and Thiele 2001), and some algorithms using Multi-Objective Particle Swarm Optimization (MOPSO) (Coello Coello and Lechuga 2002; Hettenhausen *et al.* 2013; Dehuri, Jagadev and Panda 2015). Among the above, the DSD approach is a classical algorithm for providing a well-distributed Pareto set. The main idea of the DSD algorithm is to shrink a search domain to obtain a Pareto solution in a selected area of the objective space. A well-spread distribution of the selected search domains provides a quasi-even Pareto set (Erfani and Utyuzhnikov 2011). After the original DSD algorithm, a modified directed search domain algorithm called DSD-II is put forward for enhancement of optimization efficiency (Erfani, Utyuzhnikov and Kolo 2013).

However, industrial optimization problems seem more complex when they include system level and subsystem level design and optimization. For instance, a standard spacecraft system consists of several subsystems such as a propulsion subsystem, a power subsystem and a control subsystem. The system level aims at low weight, long life and sometimes large available inside volume, whilst each subsystem has its own objectives. For example, the propulsion subsystem expects low propellant consumption, and the power subsystem presumes high power generating capacity. In spacecraft design and engineering, the system level design is always based on the trade-off design among all the subsystems. Here, a bilevel multiobjective optimization problem is formed, which is made up of upper and lower levels. The feasible space of the upper level is determined by the Pareto optimal solutions of the lower level. Then, the upper level corresponds to the system level and implements multiobjective optimization to search for the ultimate solutions among the system level objectives. The lower level includes all the subsystems and aims to find trade-off solutions (Pareto solutions) according to the objectives of all subsystems. Other practical bilevel optimization problems can be found in the field of transportation (Alizadeh, Marcotte and Savard 2013; Assadipour, Ke and Verma 2016), Stackelberg games (Sinha *et al.* 2013, 2014), and industrial location problems (Qu and Jiang 2013).

In a bilevel multiobjective optimization problem, the Pareto set on the lower level can be a decision set for the upper level. Thus, a bilevel optimization technique needs to search for the lower level Pareto solutions first and then find the optimal solutions for the upper level on the basis of the lower level trade-off solutions. This seems to be the great challenge in handling the bilevel optimization problem.

At present, there are not so many studies on bilevel multiobjective optimization problems, and relevant approaches are quite limited. Deb and Sinha (2009) have come up with an evolutionary approach with the use of NSAG-II to handle bilevel multiobjective optimization problems. To coordinate the processing of populations between the upper and lower levels, they maintain identical upper level variable values in subpopulations. This interactive upper and lower level population processing strategy is able to steer the search close to the Pareto set of the overall problem. Afterwards, they also proposed a viable and hybrid evolutionary-local-search algorithm as a solution methodology for bilevel multiobjective optimization (Deb and Sinha 2010). In this algorithm, the population sizing and termination criteria are made self-adaptive, so that no additional parameters need to be supplied by the user. Then, an automatic update of the key parameters from generation to generation is also allowed. Meanwhile, Sinha, Malo and Deb (2015) took bilevel decision making into account in bilevel multiobjective optimization.

In addition, a classical approach has been proposed by Eichfelder (2007). In this approach, a uniformly distributed set of points of upper level design variable $\boldsymbol{x}_u$ is generated at the beginning. Afterward, for each $\boldsymbol{x}_u$, the lower level Pareto solutions are found using a classical multiobjective optimization algorithm. Then, non-dominated solutions in terms of the upper level multiobjective optimization are chosen from these solutions, and an approximate upper level Pareto frontier is formed. Next, the chosen $\boldsymbol{x}_u$ vectors are refined in their vicinities and the lower level optimizations are repeated until a good approximation of the upper level Pareto frontier is obtained. The disadvantage of this approach is that, when $\boldsymbol{x}_u$ is of high dimension, generating and refining $\boldsymbol{x}_u$ is quite time-consuming.

The objective of this article is to develop an efficient classical approach for bilevel multiobjective optimization problems on the basis of the DSD algorithm. The article is structured as follows. In Section 2, key concepts and definitions about single-level and bilevel MOO problems are introduced. Then, the main principles and procedures of the DSD algorithm including DSD-II are explained in Section 3. The proposed classical bilevel multiobjective optimization approach is described in Section 4. The approach is evaluated in Section 5 according to simulation results of four test cases. Discussion as well as conclusions are presented in Section 6.

2. Bilevel multiobjective optimization problems

First, the basic concepts as well as definitions of single-level and bilevel multiobjective optimization problems are introduced.

2.1. Multiobjective optimization problems

A general MultiObjective Optimization (MOO) problem can be described as follows:

$$\begin{aligned} \text{Min} \quad & \boldsymbol{y}(\boldsymbol{x}) = (y_1(\boldsymbol{x}), y_2(\boldsymbol{x}), \ldots, y_n(\boldsymbol{x})), \\ \text{subject to} \quad & \boldsymbol{x} \in \boldsymbol{D}^*, \end{aligned} \tag{1}$$

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

In general, the solution to 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 2.1: *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 $y_i(\boldsymbol{x}) \leq y_i(\boldsymbol{x}^*)$ for all $i = 1, \ldots, n$ and $y_j(\boldsymbol{x}) < y_j(\boldsymbol{x}^*)$ for at least one j $(1 \leq j \leq n)$.*

Then, in the objective space, the vector $\boldsymbol{y}(\boldsymbol{x}^*)$ 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 2.2: *In the objective space $\boldsymbol{Y}$, an anchor point $\boldsymbol{\mu}_i$ represents the minimum of an ith objective function subject to all the constraints.*

Sometimes, no unique anchor point is obtained according to the above definition. Consequently a lexicographic-based prioritization is introduced as follows (Utyuzhnikov, Fantini and Guenov 2009).

Definition 2.3: *In the feasible objective space $\boldsymbol{Y}^* = \{\boldsymbol{y}(\boldsymbol{x}) \mid \boldsymbol{x} \in \boldsymbol{D}^*\}$, the anchor point $\boldsymbol{\mu}_i$ of an ith objective function is determined in the circular order: min y_i, min y_{i+1}, ..., min y_n, min y_1, min y_2, ..., min y_{i-1}. Any successive minimization problem: min y_{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 2.4: *A hyperplane that contains all the anchor points is called the utopia hyperplane.*

The above definitions are illustrated in Figure 1.

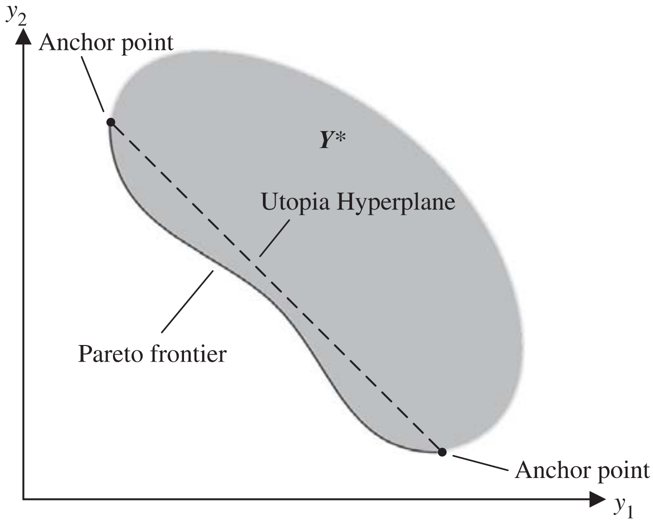

Figure 1. Basic definitions of MOO in objective space.

2.2. Bilevel multiobjective optimization problems

A standard bilevel multiobjective optimization problem consists of two, upper level and lower level, multiobjective optimization problems. The variable vector of the whole problem $\boldsymbol{x}$ consists of two parts: the upper level design variable vector $\boldsymbol{x}_u \in \mathbb{R}^{m_u}$ and the lower level design variable vector $\boldsymbol{x}_l \in \mathbb{R}^{m_l}$, such that $\boldsymbol{x} = (\boldsymbol{x}_u, \boldsymbol{x}_l) \in \mathbb{R}^{m_u} \times \mathbb{R}^{m_l}$. Then the bilevel multiobjective optimization problem can be modelled as follows:

$$\begin{aligned} \text{Min} \quad & \boldsymbol{F}(\boldsymbol{x}) = (F_1(\boldsymbol{x}), F_2(\boldsymbol{x}), \ldots, F_{n_u}(\boldsymbol{x})), \\ \text{s.t.} \quad & \begin{cases} \boldsymbol{G}(\boldsymbol{x}) = (G_1(\boldsymbol{x}), G_2(\boldsymbol{x}), \ldots, G_{K_u}(\boldsymbol{x})) \le 0, \\ \boldsymbol{x}_l \text{ solve} \begin{cases} \text{Min} \quad \boldsymbol{f}(\boldsymbol{x}) = (f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots, f_{n_l}(\boldsymbol{x})), \\ \text{s.t.} \quad\;\; \boldsymbol{g}(\boldsymbol{x}) = (g_1(\boldsymbol{x}), g_2(\boldsymbol{x}), \ldots, g_{K_l}(\boldsymbol{x})) \le 0, \end{cases} \end{cases} \end{aligned} \tag{2}$$

where $\boldsymbol{F} : \mathbb{R}^{m_u} \times \mathbb{R}^{m_l} \to \mathbb{R}^{n_u}$, $\boldsymbol{G}(\boldsymbol{x})$ are upper level constraints, $\boldsymbol{f} : \mathbb{R}^{m_u} \times \mathbb{R}^{m_l} \to \mathbb{R}^{n_l}$, and $\boldsymbol{g}(\boldsymbol{x})$ are lower level constraints.

It is shown in problem (2) that the feasible space of the upper level optimization problem is determined by the lower level Pareto solutions, while the lower level optimization is implemented with a constant vector $\boldsymbol{x}_u$ from the upper level, as mentioned above. The key problem of bilevel multiobjective optimization is to deal with the interaction between these two levels.

3. The DSD algorithm

DSD is a classical multiobjective optimization algorithm for generating a well-distributed Pareto set in multiobjective optimization problems. Then, Erfani, Utyuzhnikov and Kolo (2013) proposed a modified directed search domain algorithm called DSD-II to further improve the optimization efficiency.

3.1. The original DSD algorithm

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

Step 3.0. The objective functions are to be scaled if necessary.
Step 3.1. Search for the anchor points or modified anchor points.
Step 3.2. Form the interior of the utopia hyperplane P by

$$\begin{aligned} &P = \sum_{i=1}^{n} \alpha_i \boldsymbol{\mu}_i, \\ &\sum_{i=1}^{n} \alpha_i = 1, \\ &0 \leq \alpha_i \leq 1 \ (i = 1, \ldots, n), \end{aligned} \tag{3}$$

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

Step 3.3. Generate a set of evenly distributed reference points $\mathcal{M}$ on the utopia hyperplane by varying α_i in Equation (3).

Step 3.4. Shrink the search domain. For each reference point $\boldsymbol{M} = (M_1, \ldots, M_n) \in \mathcal{M}$, the following single-objective optimization problem is formulated similar to the physical programming (Messac 1996; Messac and Mattson 2002):

$$\begin{aligned} \text{Min} \quad & \sum_{i=1}^{n} y_i(\boldsymbol{x}), \\ \text{s.t.} \quad & y_i(\boldsymbol{x}) \leq M_i, \\ & \boldsymbol{x} \in \boldsymbol{D}^*. \end{aligned} \tag{4}$$

In problem (4), the constraints $y_i(\boldsymbol{x}) \leq M_i$ 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 degrade the evenness of the Pareto set. To overcome this problem in the 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 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 (4) is modified as

$$\hat{y}_i(x) \leq \sum_{j=1}^{n} M_j B_{ji} \ (i = 1, \ldots, n), \tag{5}$$

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

Step 3.5. Solve the new single-objective optimization based on the shrunk search domain for each reference point $\boldsymbol{M} = (M_1, \ldots, M_n) \in \mathcal{M}$:

$$\begin{aligned} \text{Min} \quad & \sum_{i=1}^{n} y_i(\boldsymbol{x}), \\ \text{s.t.} \quad & \hat{y}_i(x) \le \sum_{j=1}^{n} M_j B_{ji}, \\ & \boldsymbol{x} \in \boldsymbol{D}^*. \end{aligned} \tag{6}$$

Sometimes there may no feasible solution found in the search domain, when the boundary of the feasible objective space is non-convex (Messac and Mattson 2004; Utyuzhnikov, Fantini and Guenov 2009; Erfani and Utyuzhnikov 2011). In this case, the search domain should be flipped to the opposite side of the utopia hyperplane to search the points on the Pareto frontier by reversing the inequalities in optimization problem (6) as

$$\hat{y}_i(x) \ge \sum_{j=1}^{n} M_j B_{ji} \ (i = 1, \ldots, n). \tag{7}$$

Meanwhile, it is possible that the orthogonal projection of the utopia hyperplane does not fully cover the entire Pareto frontier (Messac and Mattson 2004; Utyuzhnikov, Fantini and Guenov 2009; Erfani and Utyuzhnikov 2011). In order to capture undiscovered regions, the search domain should be rotated if the reference point $\boldsymbol{M}$ is located on the edge of the hyperplane.

Step 3.6. Eliminate local Pareto solutions by a filtering procedure.

3.2. The modified DSD algorithm: 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

$$\boldsymbol{v} = \boldsymbol{M}_c - \boldsymbol{M}, \tag{8}$$

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

$$\gamma = \arccos \left| \frac{\boldsymbol{v} \cdot \boldsymbol{n}_h}{\|\boldsymbol{v}\| \, \|\boldsymbol{n}_h\|} \right| \le \theta, \tag{9}$$

where $\boldsymbol{n}_h$ is a vector normal to the utopia hyperplane. By setting the value of θ, the search domain based on each reference point is shrunk. Thus, two advantages are brought for improvement of computational efficiency. On one hand, the shrinking calculation process is simplified as 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.

Meanwhile, DSD-II also proposes a modified utopia hyperplane to remove the rotating procedure for further enhancing the efficiency. However, this is achieved by some increase of computational time in comparison with the original DSD algorithm. The reader is referred to the article by Erfani, Utyuzhnikov and Kolo (2013) for details of modified utopia hyperplane generation.

4. An extension of the DSD approach to bilevel multiobjective optimization

Thus, each Pareto point of the upper level corresponds to a decision vector $\boldsymbol{x} = (\boldsymbol{x}_u, \boldsymbol{x}_l)$ in the decision space, and the image of the $\boldsymbol{x}$ in the lower level objective space is a Pareto point of the lower level. Therefore, each Pareto point of the upper level corresponds to a Pareto point of the lower level, depicted in Figure 2, and they share the same decision vector $\boldsymbol{x} = (\boldsymbol{x}_u, \boldsymbol{x}_l)$.

In the bilevel optimization problem, there exists a variety of decision making strategies on the lower level. As the decision making strategy changes, the upper level Pareto frontier obtained can be different (Sinha, Malo and Deb 2015). In this article, the optimistic formulation is considered. In such a context, the best solution for the upper level is selected from the lower level Pareto solutions.

Then, in contrast to the upper level multiobjective optimization, the lower level optimization only needs to find out an optimal solution rather than generate the whole Pareto frontier. To realize this goal, lower level reference point coefficients are applied and optimized. A double-layer optimizer is designed in such a way that both $\boldsymbol{x}_u$ and the vector of lower level reference point coefficients $\boldsymbol{\alpha}_l$ are considered as design variables at the same time. Along with the use of the DSD algorithm, the proposed Bilevel DSD (BDSD) approach aims to solve the bilevel multiobjective optimization problem efficiently as well as to guarantee an evenness of Pareto set distribution.

4.1. Double-layer optimizer

A double-layer optimizer plays a key role in the proposed algorithm. It is focused on bilevel single-objective optimization to find the Pareto solutions based on the reference points on the upper level. In this subsection, the definition of design variables in the double-layer optimizer is firstly discussed, and then the major implementation steps of this optimizer are introduced.

4.1.1. Design variables of the double-layer optimizer

In the bilevel multiobjective optimization, the upper level multiobjective optimization problem is broken down into a number of single-objective optimization problems under the framework of the DSD algorithm. For each of these single-objective optimization problems, an optimal solution $\boldsymbol{x} = (\boldsymbol{x}_u, \boldsymbol{x}_l)$ is found. All these optimal solutions, except the local optimal ones, form the set of Pareto solutions of the upper level. As a part of the Pareto solution $\boldsymbol{x}$, the optimal $\boldsymbol{x}_u$ is acquired after the upper level optimization, while the optimal $\boldsymbol{x}_l$ is obtained after the lower level optimization in which the optimal $\boldsymbol{x}_u$ is fixed. In order to search for this optimal $\boldsymbol{x}_l$ in the lower level optimization, the concept of the reference points and reference point coefficients in the DSD algorithm are utilized.

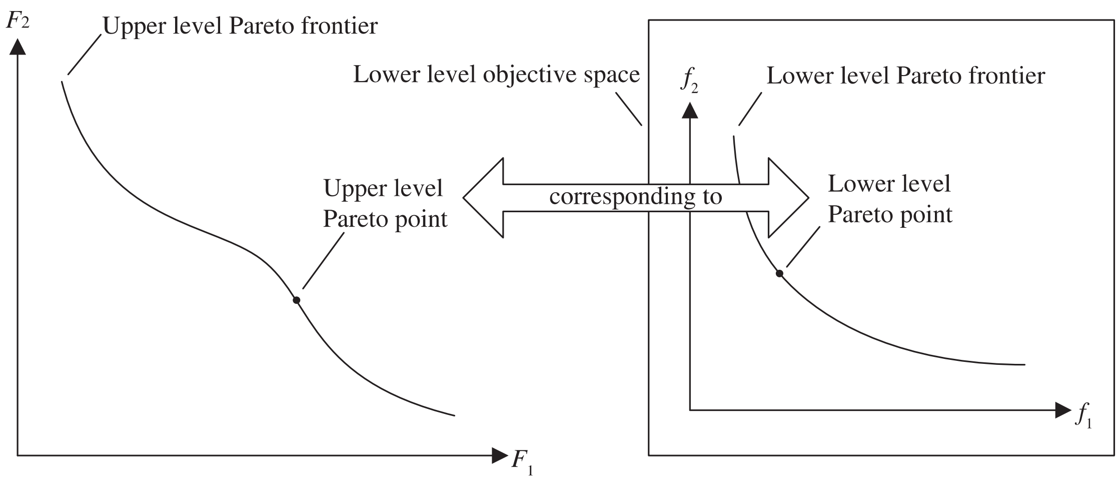

Figure 2. Relationship between Pareto points of upper and lower levels.

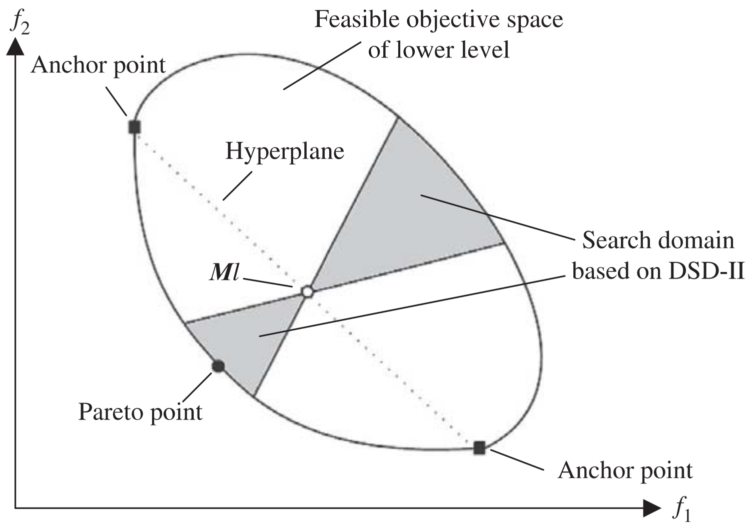

Figure 3. Relationship between the Pareto point and the reference point.

In the bilevel multiobjective optimization under the framework of the DSD algorithm, each lower level optimal solution corresponds to a lower level reference point $\boldsymbol{M}_l$, as shown in Figure 3. Consequently, searching for the optimal $\boldsymbol{x}_l$, which corresponds to a lower level Pareto solution with respect to an upper level function, can be transformed to the problem of finding the optimal $\boldsymbol{\alpha}_l$.

Therefore, $\boldsymbol{x}_u$ and $\boldsymbol{\alpha}_l$ are design variables to be optimized simultaneously in the double-layer optimizer.

4.1.2. Implementation steps of the double-layer optimizer

During the implementation process of the double-layer optimizer, the following steps are needed (as shown in Figure 4).

Step 4.1.1. Set original values for $\boldsymbol{x}_u$ and $\boldsymbol{\alpha}_l$. For the lower level multiobjective optimization problem, which contains n_l objective functions as determined in problem (2), $\boldsymbol{\alpha}_l$ is defined as

$$\begin{aligned} &\boldsymbol{\alpha}_l = (\alpha_{l1}, \alpha_{l2}, \ldots, \alpha_{l(n_l-1)}, \alpha_{ln_l}), \\ &\alpha_{ln_l} = 1 - (\alpha_{l1} + \alpha_{l2} + \cdots + \alpha_{l(n_l-1)}), \\ &0 \le \alpha_{li} \le 1 \ (i = 1, \ldots, n_l), \end{aligned} \tag{10}$$

where α_{li} are reference point coefficients needed for generating a reference point on the lower level.

Step 4.1.2. Find lower level (modified) anchor points $\boldsymbol{\mu}_{li}$ $(i = 1, 2, \ldots, n_l)$ by solving the single-objective optimization problem for each i from 1 to n_l:

$$\begin{aligned} \text{Min} \quad & f_i(\boldsymbol{x}), \\ \text{s.t.} \quad & \boldsymbol{g}(\boldsymbol{x}) \le 0. \end{aligned} \tag{11}$$

Step 4.1.3. Generate a lower level reference point $\boldsymbol{M}_l$ on the utopia hyperplane according to the lower level reference point coefficients α_{li}, $i = (1, \ldots, n_l)$.

$$\boldsymbol{M}_l = \sum_{i=1}^{n_l} \alpha_{li} \boldsymbol{\mu}_{li}. \tag{12}$$

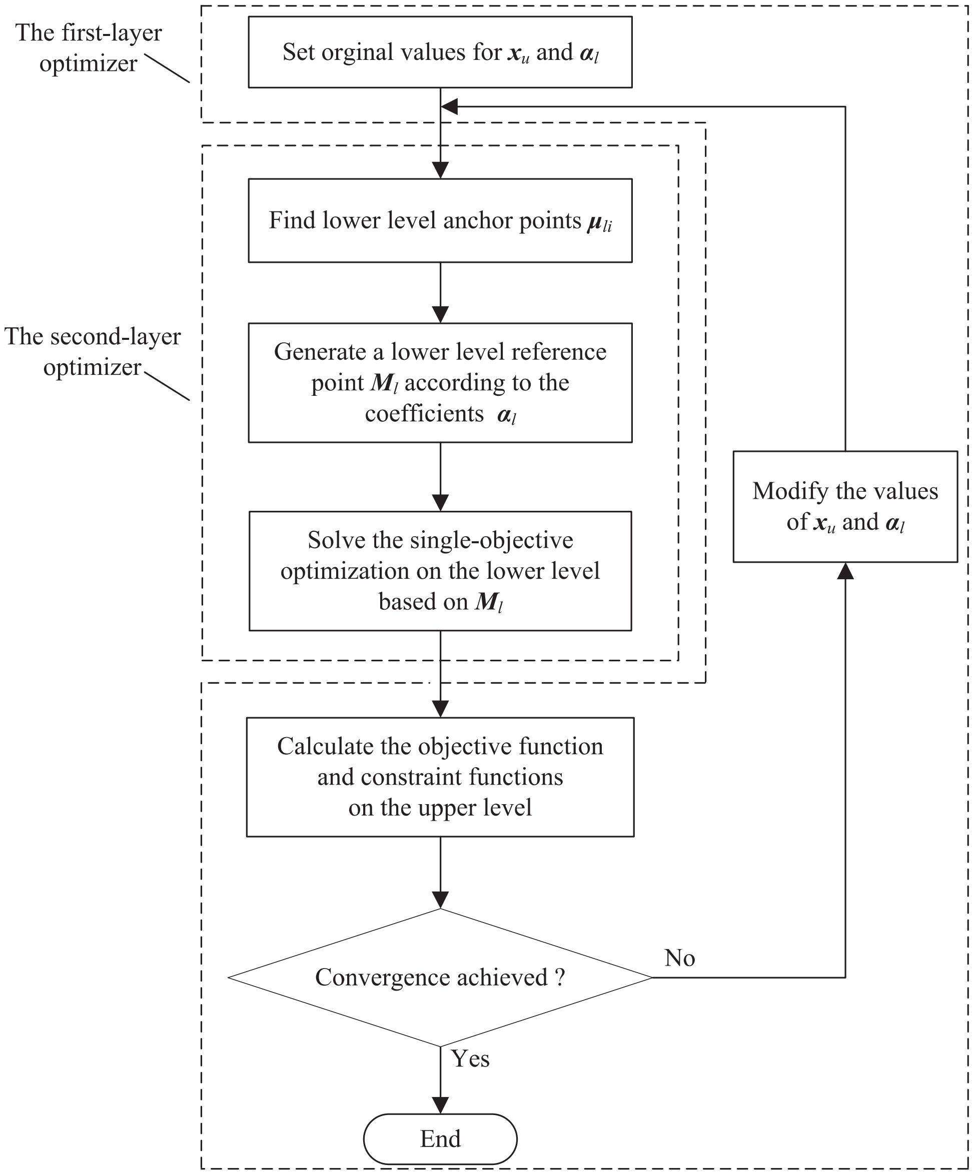

Figure 4. Implementation steps of the double-layer optimizer.

Step 4.1.4. For the reference point $\boldsymbol{M}_l = (M_{l1}, \ldots, M_{ln_l})$, solve the single-objective optimization formed by the DSD-II shrinking strategy in the lower level:

$$\begin{aligned} \text{Min} \quad & \sum_{i=1}^{n_l} f_i(\boldsymbol{x}), \\ \text{s.t.} \quad & \boldsymbol{g}(\boldsymbol{x}) \leq 0, \\ & \gamma_l = \arccos \left| \frac{\boldsymbol{v}_l \cdot \boldsymbol{n}_{lh}}{\|\boldsymbol{v}_l\| \|\boldsymbol{n}_{lh}\|} \right| \leq \theta_l, \end{aligned} \tag{13}$$

where similarly $\boldsymbol{v}_l = \boldsymbol{M}_{lc} - \boldsymbol{M}_l$, $\boldsymbol{M}_{lc} = \boldsymbol{f}(\boldsymbol{x})$ with $\boldsymbol{x}$ to be searched in the shrunk search domain of the lower level, $\boldsymbol{n}_{lh}$ is a vector normal to the utopia hyperplane of the lower level, and θ_l is the value set for shrinking the search domain of the lower level.

Step 4.1.5. Calculate the objective function and constraint functions of the upper level single-objective optimization problem. It depends on whether the double-layer optimizer is used either to find the upper level anchor points or to search for the upper level Pareto solution based on the reference point.

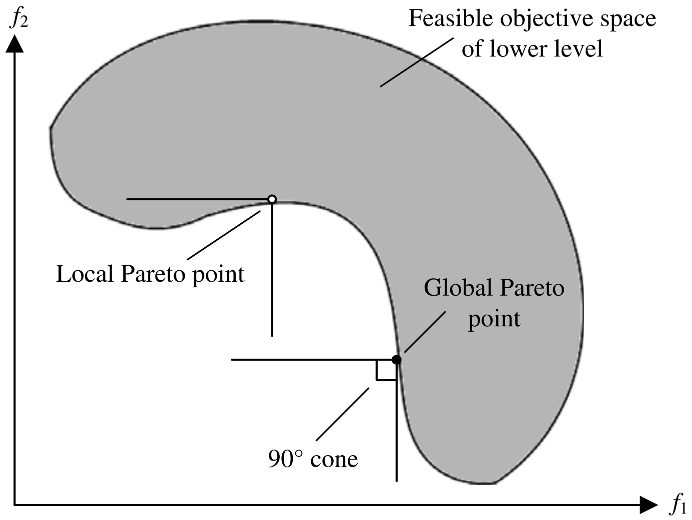

Figure 5. Illustration of local and global Pareto points.

Step 4.1.6. Analyse whether the convergence of the upper level single-objective optimization problem has been achieved. If convergence is achieved, then the double-layer optimization is completed. Otherwise, the values of $\boldsymbol{x}_u$ and $\boldsymbol{\alpha}_l$ should be further modified.

Steps 4.1.2–4.1.4 construct the second-layer optimizer to implement lower level multiobjective optimization. The other steps make up the first-layer optimizer which is directly relevant to the single-objective optimization on the upper level. Both optimizers make up the entire double-layer optimizer.

4.1.3. Filtering strategy in lower level multiobjective optimization

Sometimes, the above double-layer optimizer can finally find an upper level Pareto solution which is a local Pareto solution on the lower level. Then, the filtering strategy based on the search domain (Erfani and Utyuzhnikov 2011) should be applied. For a lower level Pareto candidate point $\boldsymbol{P}_{sl} = (P_{sl1}, P_{sl2}, \ldots, P_{sln_l})$ obtained in step 4.1.4, the search domain is a 90° cone in the lower level objective space as follows:

$$C_{90^\circ}^{\boldsymbol{P}_{sl}} = \{\boldsymbol{f}(\boldsymbol{x}) \mid f_i(\boldsymbol{x}) \le P_{sli}, \forall\, i = 1, \ldots, n_l\}. \tag{14}$$

Denote the feasible objective space on the lower level by $\boldsymbol{Y}_l^*$ as

$$\boldsymbol{Y}_l^* = \{\boldsymbol{f}(\mathbf{x}) \mid \boldsymbol{g}(\boldsymbol{x}) \le 0\}. \tag{15}$$

In the lower level objective space, if $C_{90^\circ}^{\boldsymbol{P}_{sl}} \cap \boldsymbol{Y}_l^* = \{\boldsymbol{P}_{sl}\}$ only, then $\boldsymbol{P}_{sl}$ is a global Pareto point on the lower level. Otherwise, $\boldsymbol{P}_{sl}$ is dominated by another solution in the feasible objective space on the lower level, which means $\boldsymbol{P}_{sl}$ is a local Pareto point on the lower level. A two-dimensional case is illustrated in Figure 5.

Therefore, the global Pareto constraint should be added to problem (13) in step 4.1.4 as follows:

$$C_{90^\circ}^{\boldsymbol{P}_{sl}} \cap \boldsymbol{Y}_l^* = \{\boldsymbol{P}_{sl}\}. \tag{16}$$

If $\boldsymbol{P}_{sl}$ is a local Pareto point of the lower level, the convergence in step 4.1.6 cannot be achieved. Thus, a new iteration from step 4.1.2 to step 4.1.6 is still required with modified $\boldsymbol{x}_u$ and $\boldsymbol{\alpha}_l$.

4.2. Overall optimization procedures of the BDSD algorithm

This approach takes advantage of the double-layer optimizer, which is made up of the first-layer and second-layer optimizers. On the basis of the DSD-II algorithm, the overall procedure for tackling the bilevel multiobjective optimization problem is as follows (as depicted in Figure 6).

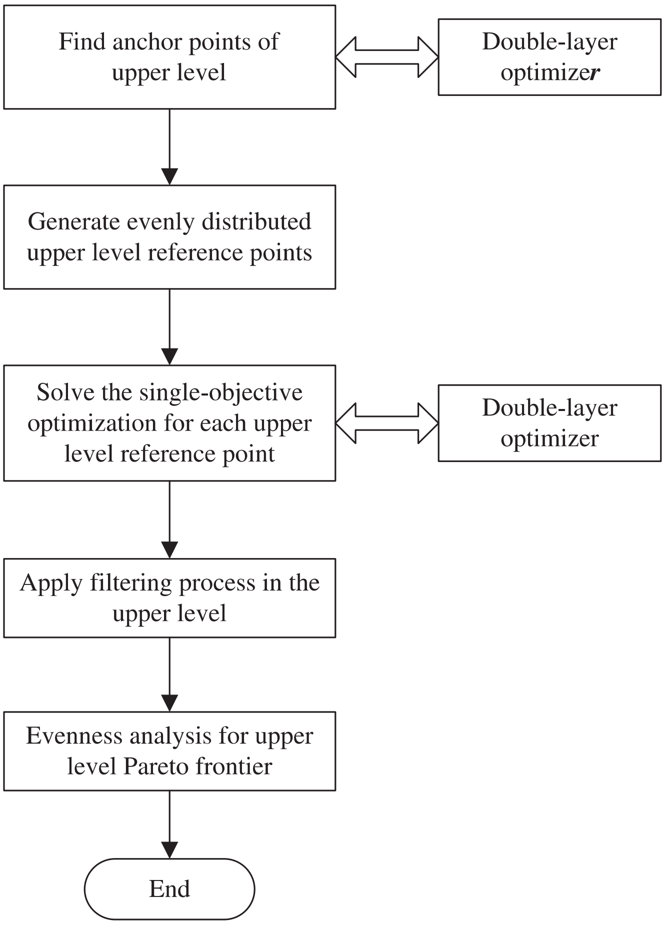

Figure 6. Overall optimization procedures of the BDSD algorithm.

Step 4.2.1. Find the anchor points of upper level $\boldsymbol{\mu}_{ui}$ $(i = 1, 2, \dots, n_u)$ with the use of the double-layer optimizer by solving the following single-objective optimization problem for each i from 1 to n_u:

$$\begin{aligned} \text{Min} \quad & F_i(\boldsymbol{x}), \\ \text{s.t.} \quad & \boldsymbol{x}_l \in \operatorname{argmin}_{(\boldsymbol{x}_l)}\{\boldsymbol{f}(\boldsymbol{x}) = (f_1(\boldsymbol{x}), \dots, f_{n_l}(\boldsymbol{x})) \mid \boldsymbol{g}(\boldsymbol{x}) \le 0\}, \\ & \boldsymbol{G}(\boldsymbol{x}) \le 0. \end{aligned} \tag{17}$$

Step 4.2.2. Generate evenly distributed upper level reference points $\mathcal{M}_u$ on the utopia hyperplane by varying α_{ui}:

$$\begin{aligned} & P_u = \sum_{i=1}^{n_u} \alpha_{ui} \boldsymbol{\mu}_{ui}, \\ & \sum_{i=1}^{n_u} \alpha_{ui} = 1, \\ & 0 \le \alpha_{ui} \le 1 \ (i = 1, \dots, n_u), \end{aligned} \tag{18}$$

where α_{ui} are reference point coefficients for generating reference points.

Step 4.2.3. For each upper level reference point $\boldsymbol{M}_u = (M_{u1}, \dots, M_{un_u}) \in \mathcal{M}_u$, solve the single-objective optimization formed by the DSD-II shrinking strategy with the use of the double-layer optimizer.

$$
\begin{aligned}
\text{Min} \quad & \sum_{i=1}^{n_u} F_i(\boldsymbol{x}), \\
\text{s.t.} \quad & \boldsymbol{G}(\boldsymbol{x}) \le 0, \\
& \boldsymbol{x}_l \in \text{argmin}_{(\boldsymbol{x}_l)}\{\boldsymbol{f}(\boldsymbol{x}) = (f_1(\boldsymbol{x}), \ldots, f_{n_l}(\boldsymbol{x})) \mid \boldsymbol{g}(\boldsymbol{x}) \le 0\}, \\
& \gamma_u = \arccos\left|\frac{\boldsymbol{v}_u \cdot \boldsymbol{n}_{uh}}{\|\boldsymbol{v}_u\|\|\boldsymbol{n}_{uh}\|}\right| \le \theta_u,
\end{aligned} \tag{19}
$$

where $\boldsymbol{v}_u = \boldsymbol{M}_{uc} - \boldsymbol{M}_u$, $\boldsymbol{M}_{uc} = \boldsymbol{F}(\boldsymbol{x})$ with $\boldsymbol{x}$ to be searched in the shrunk search domain of the upper level, $\boldsymbol{n}_{uh}$ is a vector normal to the utopia hyperplane of the upper level, and θ_u is the value set for shrinking the search domain of the upper level.

Step 4.2.4. Utilize a filtering process to remove local upper level Pareto solutions.

Step 4.2.5. Evenness analysis for upper level Pareto set.

5. Test cases

The proposed BDSD approach is validated on four test cases of which the first two are from Deb and Sinha (2009). The test cases include both convex and non-convex Pareto frontiers on the upper level. The results obtained by BDSD are compared against the classical reference technique (Eichfelder 2007) with respect to the evenness of the upper level Pareto set and the time consumption of the whole bilevel optimization process. In the approach of Eichfelder (2007), the DSD-II algorithm is used to search for the lower level Pareto solutions.

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

$$
\boldsymbol{d}^i = (\|\boldsymbol{P}_{su}^i - \boldsymbol{P}_{su}^1\|, \ldots, \|\boldsymbol{P}_{su}^i - \boldsymbol{P}_{su}^{i-1}\|, \|\boldsymbol{P}_{su}^i - \boldsymbol{P}_{su}^{i+1}\|, \ldots, \|\boldsymbol{P}_{su}^i - \boldsymbol{P}_{su}^{n_{pu}}\|), \tag{20}
$$

where n_{pu} is the number of Pareto points (including n_u anchor points) obtained in the upper level Pareto set. Then, the effective distance vector $\boldsymbol{d}_{\text{eff}}^i$ for $\boldsymbol{P}_{su}^i$ is defined as

$$
\boldsymbol{d}_{\text{eff}}^i = (\min(\boldsymbol{d}^i), \text{second} \min(\boldsymbol{d}^i), \ldots, n_u\text{th} \min(\boldsymbol{d}^i)), \tag{21}
$$

where $n_u\text{th} \min(\boldsymbol{d}^i)$ is the n_uth smallest value of $\boldsymbol{d}^i$. Then, the coefficient of evenness for upper level E_u is defined as

$$
E_u = \frac{\max(\boldsymbol{d}_{\text{eff}}^1, \boldsymbol{d}_{\text{eff}}^2, \ldots, \boldsymbol{d}_{\text{eff}}^{n_{pu}-n_u})}{\min(\boldsymbol{d}_{\text{eff}}^1, \boldsymbol{d}_{\text{eff}}^2, \ldots, \boldsymbol{d}_{\text{eff}}^{n_{pu}-n_u})}. \tag{22}
$$

If $E_u = 1$, then the upper level Pareto set is completely even. If E_u increases, then the evenness of the Pareto set becomes worse.

In the whole bilevel 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 basis optimization algorithm.

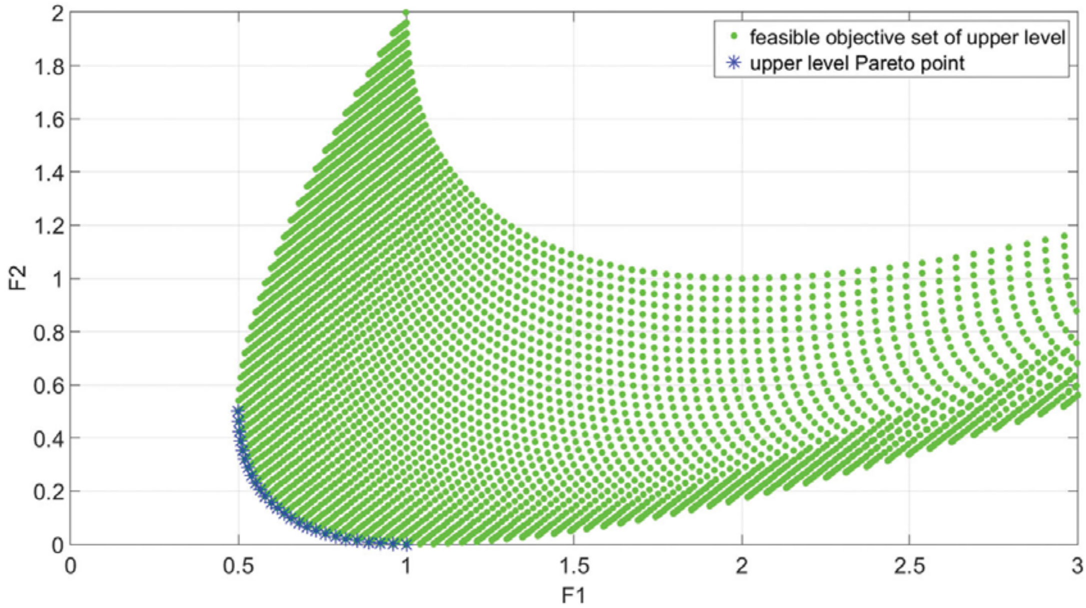

Figure 7. Upper level Pareto points ($n_{pu} = 26$) of Problem 1 by BDSD and the Eichfelder (2007) approach.

5.1. Problem 1

First, a simplistic bilevel two-objective optimization problem is considered in which $\boldsymbol{x}_u = (x_{u1})$ and $\boldsymbol{x}_l = (x_{l1}, x_{l2})$:

$$
\begin{aligned}
\text{Min} \quad & \boldsymbol{F}(\boldsymbol{x}) = (F_1(\boldsymbol{x}), F_2(\boldsymbol{x})), \\
\text{s.t.} \quad & \boldsymbol{x}_l \in \text{argmin}_{(\boldsymbol{x}_l)} \{\boldsymbol{f}(\boldsymbol{x}) = (f_1(\boldsymbol{x}), f_2(\boldsymbol{x}))\}, \\
& 0 \leq x_{u1} \leq 2, \quad -1 \leq x_{l1}, x_{l2} \leq 2, \\
& \text{where} \\
& F_1(\boldsymbol{x}) = (x_{l1} - 1)^2 + x_{l2}^2 + x_{u1}^2, \\
& F_2(\boldsymbol{x}) = (x_{l1} - 1)^2 + x_{l2}^2 + (x_{u1} - 1)^2, \\
& f_1(\boldsymbol{x}) = x_{l1}^2 + x_{l2}^2, \\
& f_2(\boldsymbol{x}) = (x_{l1} - x_{u1})^2 + x_{l2}^2.
\end{aligned} \tag{23}
$$

The upper level Pareto sets acquired with the BDSD approach and the algorithm of Eichfelder (2007) are identical, as shown in Figure 7. To illustrate the feasible objective space of the upper level, a set of feasible non-Pareto solutions is generated, and the corresponding feasible objective set is depicted in Figure 7. In addition, a lower level frontier obtained with the approach of Eichfelder is shown in Figure 8. Comparison between these two approaches on the evenness E_u, as well as time consumption t, is presented in Table 1. Here, η_t is the ratio of time consumption by BDSD to that by the Eichfelder (2007) approach.

It can be inferred that in this case the BDSD approach drastically reduces optimization time by about 75–80%. In addition, the more upper level Pareto solutions obtained, the greater is the effect in saving computational time achieved.

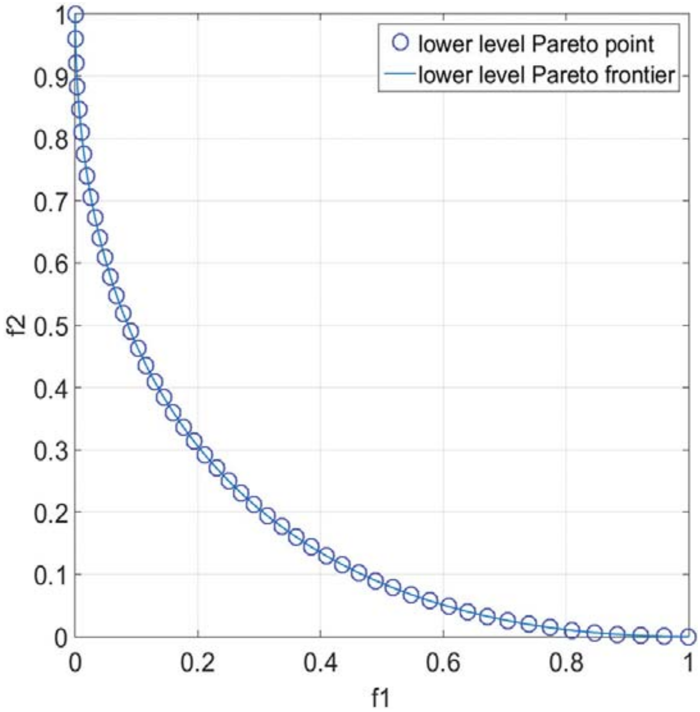

Figure 8. A lower level Pareto frontier of Problem 1 ($x_{u1} = 1$).

Table 1. Comparison of optimization performance in Problem 1.

	BDSD		Eichfelder (2007) approach		
n_{pu}	E_u	$t(s)$	E_u	$t(s)$	η_t (%)
26	1.39	12.5	1.39	47.6	26.3
51	1.40	21.5	1.40	94.7	22.7
101	1.41	40.7	1.41	189.5	21.5

5.2. Problem 2

Problem 2 also has $\boldsymbol{x}_u = (x_{u1})$ and $\boldsymbol{x}_l = (x_{l1}, x_{l2})$.

$$
\begin{aligned}
\text{Min} \quad & \boldsymbol{F}(\boldsymbol{x}) = (F_1(\boldsymbol{x}), F_2(\boldsymbol{x})), \\
\text{s.t.} \quad & \boldsymbol{x}_l \in \operatorname{argmin}_{(\boldsymbol{x}_l)}\{\boldsymbol{f}(\boldsymbol{x}) = (f_1(\boldsymbol{x}), f_2(\boldsymbol{x}))|g_1(\boldsymbol{x}) = x_{l1}^2 + x_{l2}^2 - x_{u1}^2 \le 0\}, \\
& G_1(\boldsymbol{x}) = -(x_{l1} + x_{l2} + 1) \le 0, \\
& 0 \le x_{u1} \le 1, \quad -1 \le x_{l1}, x_{l2} \le 1, \\
& \text{where} \\
& F_1(\boldsymbol{x}) = x_{l1} - x_{u1}, \\
& F_2(\boldsymbol{x}) = x_{l2}, \\
& f_1(\boldsymbol{x}) = x_{l1}, \\
& f_2(\boldsymbol{x}) = x_{l2}.
\end{aligned} \tag{24}
$$

In contrast to Problem 1, Problem 2 has inequality constraints both on the upper and lower levels. The upper level Pareto sets obtained with BDSD and the Eichfelder algorithm are shown in Figures 9 and 10, respectively. A lower level frontier obtained with the approach of Eichfelder is shown in Figure 11. These approaches are compared with each other in Table 2.

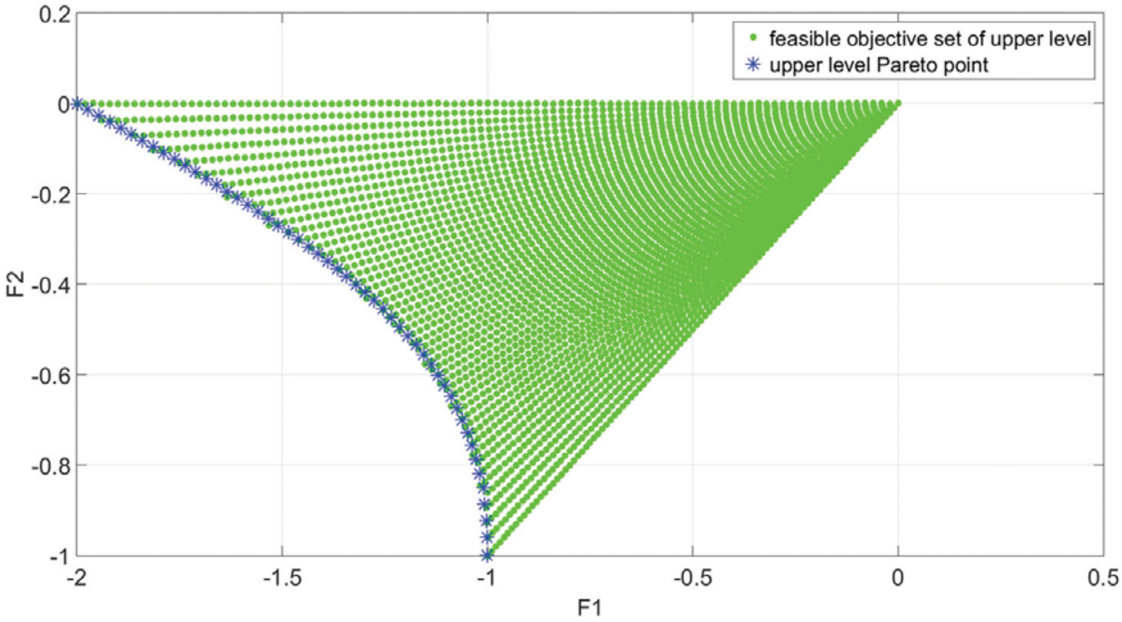

Figure 9. Upper level Pareto points ($n_{pu} = 51$) of Problem 2 by BDSD approach.

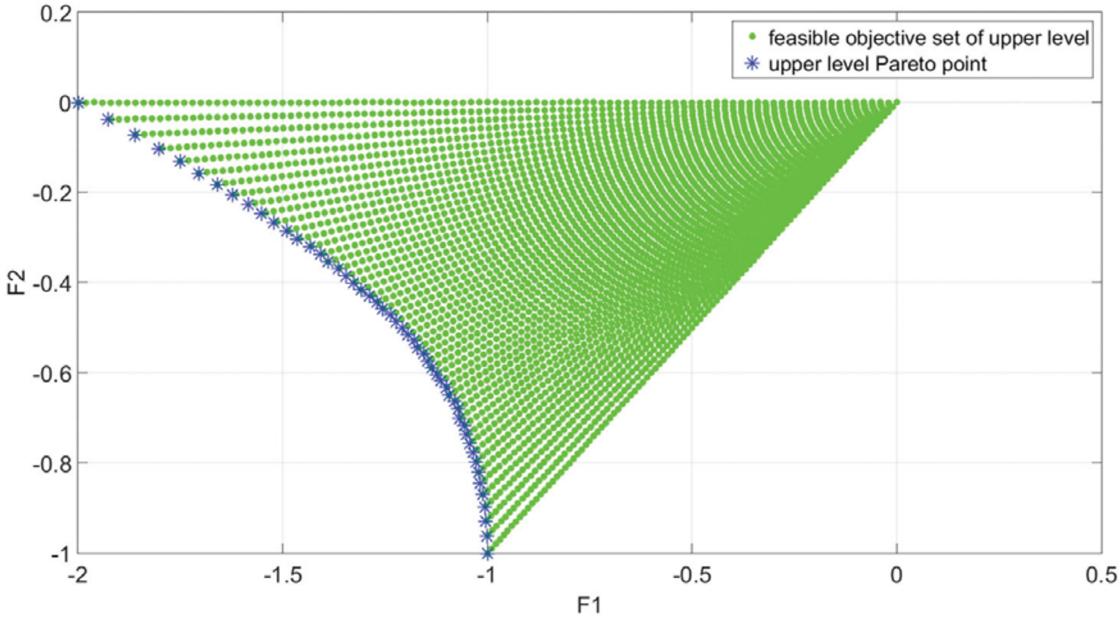

Figure 10. Upper level Pareto points ($n_{pu} = 51$) of Problem 2 by approach (Eichfelder 2007)

In Problem 2, BDSD essentially cuts down the time consumption by nearly half, whilst significantly improving the evenness of the upper level Pareto set. Similar to Problem 1, the impact on computational time reduction increases as the number of upper level Pareto points generated rises.

5.3. Problem 3

In Problem 3, the upper and lower level design variables contain two elements: $\boldsymbol{x}_u = (x_{u1}, x_{u2})$ and $\boldsymbol{x}_l = (x_{l1}, x_{l2})$, respectively.

$$\begin{aligned} \text{Min} \quad & \boldsymbol{F}(\boldsymbol{x}) = (F_1(\boldsymbol{x}), F_2(\boldsymbol{x})), \\ \text{s.t.} \quad & \boldsymbol{x}_l \in \operatorname{argmin}_{(\boldsymbol{x}_l)} \{\boldsymbol{f}(\boldsymbol{x}) = (f_1(\boldsymbol{x}), f_2(\boldsymbol{x}))\}, \end{aligned}$$

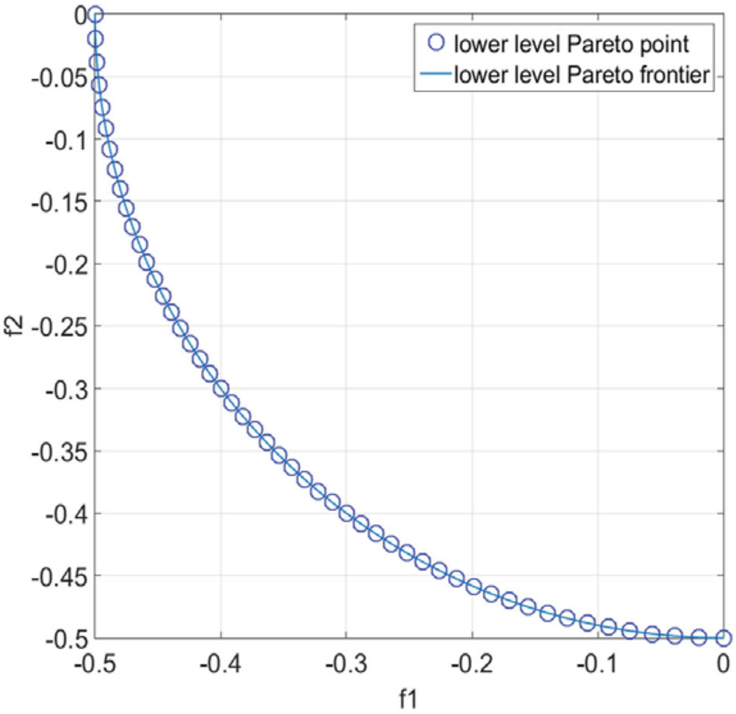

Figure 11. A lower level Pareto frontier of Problem 2 ($x_{u1} = 0.5$).

Table 2. Comparison of optimization performance in Problem 2.

	BDSD		Eichfelder (2007) approach		
n_{pu}	E_u	$t(s)$	E_u	$t(s)$	η_t (%)
26	1.37	89.2	4.22	173.5	51.4
51	1.42	161.0	4.47	329.6	48.8
96	1.46	303.3	5.79	647.0	46.9

$$0 \le x_{u1}, x_{u2} \le 2, \quad -1 \le x_{l1}, x_{l2} \le 2,$$

where

$$\begin{aligned} F_1(\boldsymbol{x}) &= (x_{l1} - 1)^2 + x_{l2}^2 + x_{u1}^2 + (x_{u2} - 1)^2, \\ F_2(\boldsymbol{x}) &= (x_{l1} - 1)^2 + x_{l2}^2 + (x_{u1} - 1)^2 + x_{u2}^2, \\ f_1(\boldsymbol{x}) &= x_{l1}^2 + (x_{l2} - x_{u1})^2, \\ f_2(\boldsymbol{x}) &= (x_{l1} - x_{u2})^2 + x_{l2}^2. \end{aligned} \tag{25}$$

In this case, the coupling relationship between the upper level and the lower level is stronger. The upper level Pareto sets generated by the two approaches are almost identical, as shown in Figure 12, and a lower level frontier obtained with the approach of Eichfelder is shown in Figure 13. Then, comparison between the two approaches is presented in Table 3.

In Problem 3, the evenness of the upper level Pareto set can still be maintained at a high level by both approaches. However, as the dimension of the upper level design variable vector $\boldsymbol{x}_u$ increases, the whole optimization time is reduced by almost 90% or more with the use of the BDSD approach. In other words, the optimization efficiency is enhanced as much as one order of magnitude or even more. Hence, the BDSD approach becomes more effective and efficient as the dimension of $\boldsymbol{x}_u$ increases.

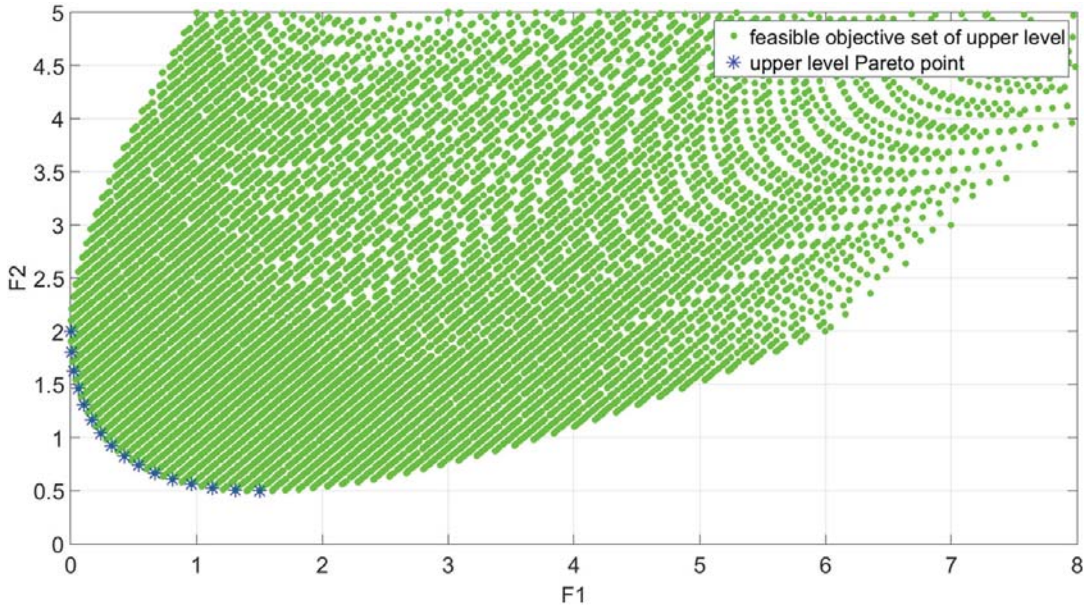

Figure 12. Upper level Pareto points ($n_{pu} = 16$) of Problem 3 by BDSD and the Eichfelder (2007) approach.

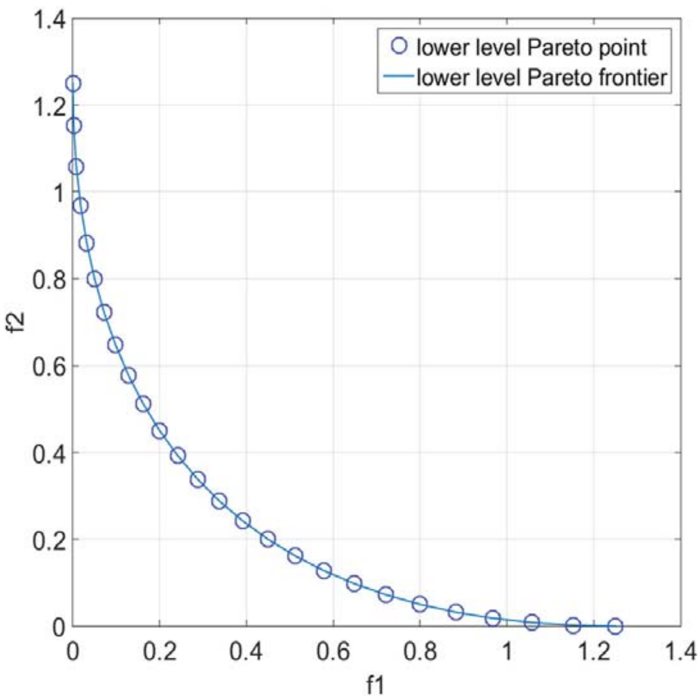

Figure 13. A lower level Pareto frontier of Problem 3 ($x_{u1} = 0.5$, $x_{u2} = 1$).

5.4. Problem 4

In Problem 4, one constraint is added on the upper level on the basis of Problem 3.

$$
\begin{aligned}
\text{Min} \quad & \boldsymbol{F}(\boldsymbol{x}) = (F_1(\boldsymbol{x}), F_2(\boldsymbol{x})), \\
\text{s.t.} \quad & \boldsymbol{x}_l \in \operatorname{argmin}_{(\boldsymbol{x}_l)}\{\boldsymbol{f}(\boldsymbol{x}) = (f_1(\boldsymbol{x}), f_2(\boldsymbol{x}))\}, \\
& G_1(\boldsymbol{x}) = -(x_{u1} + x_{u2} - 2) \leq 0, \\
& 0 \leq x_{u1}, x_{u2} \leq 2, \quad -1 \leq x_{l1}, x_{l2} \leq 2, \\
& \text{where} \\
& F_1(\boldsymbol{x}) = (x_{l1} - 1)^2 + x_{l2}^2 + x_{u1}^2 + (x_{u2} - 1)^2,
\end{aligned}
$$

Table 3. Comparison of optimization performance in Problem 3.

	BDSD		Eichfelder (2007) approach		
n_{pu}	E_u	$t(s)$	E_u	$t(s)$	η_t (%)
16	1.39	22.9	1.35	214.9	10.7
31	1.42	54.7	1.38	851.7	6.4
61	1.43	93.1	1.40	4428.7	2.1

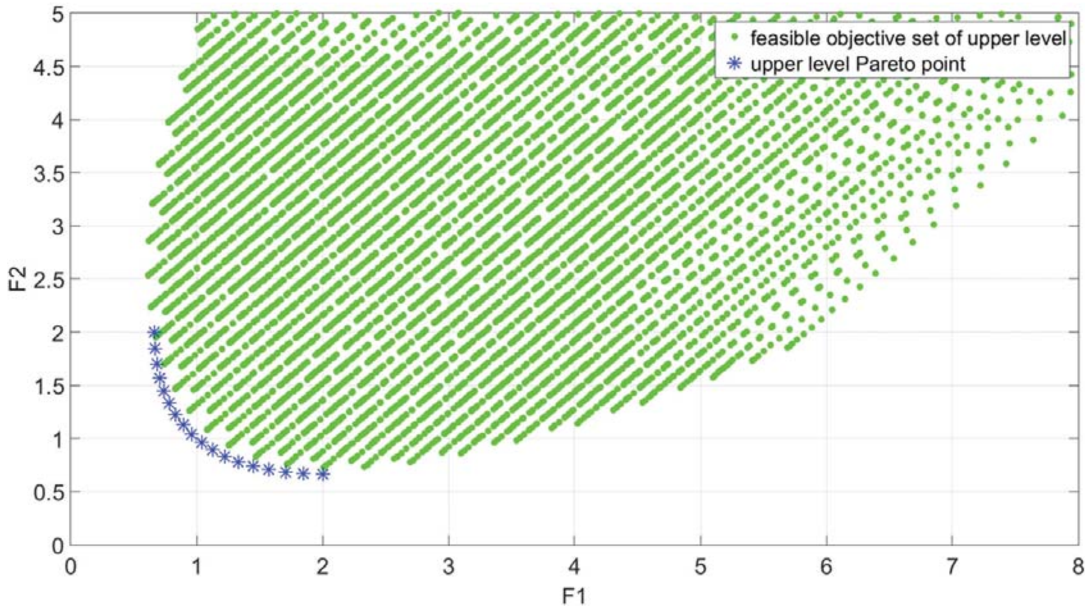

Figure 14. Upper level Pareto points ($n_{pu} = 18$) of Problem 4 by BDSD.

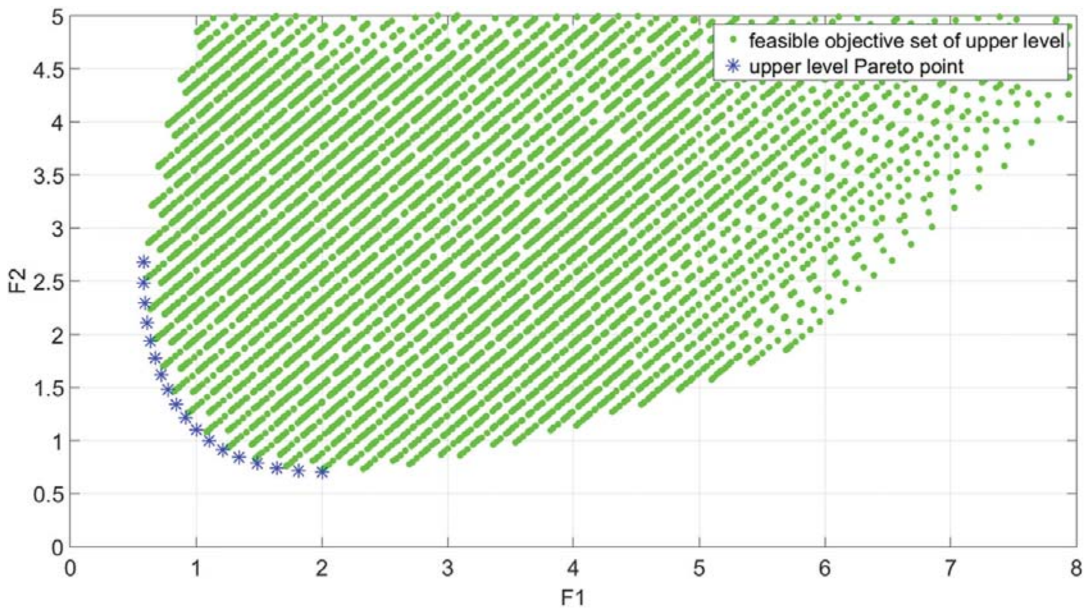

Figure 15. Upper level Pareto points ($n_{pu} = 18$) of Problem 4 by the Eichfelder (2007) approach.

$$
\begin{aligned}
F_2(\boldsymbol{x}) &= (x_{l1}-1)^2 + x_{l2}^2 + (x_{u1}-1)^2 + x_{u2}^2,\\
f_1(\boldsymbol{x}) &= x_{l1}^2 + (x_{l2}-x_{u1})^2,\\
f_2(\boldsymbol{x}) &= (x_{l1}-x_{u2})^2 + x_{l2}^2.
\end{aligned} \tag{26}
$$

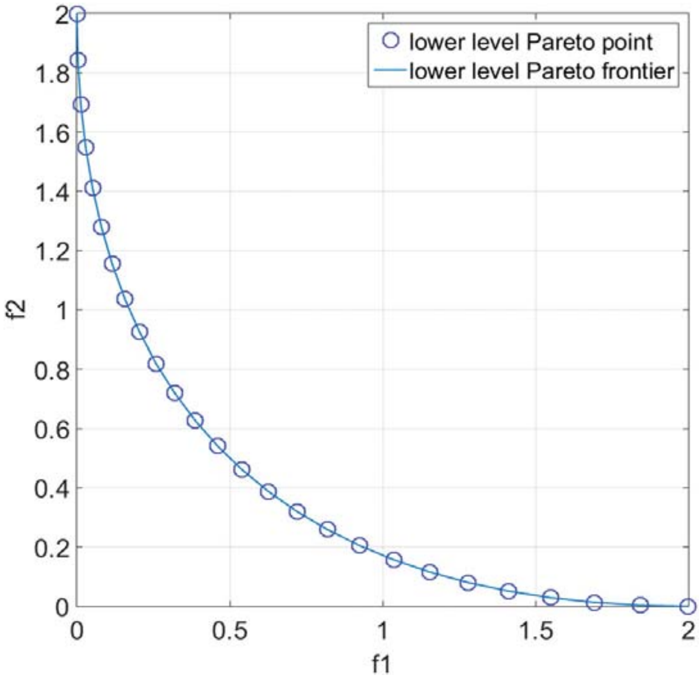

Figure 16. A lower level Pareto frontier of Problem 4 ($x_{u1} = 1$, $x_{u2} = 1$).

Table 4. Comparison of optimization performance in Problem 4.

	BDSD		Eichfelder (2007) approach		
n_{pu}	E_u	$t(s)$	E_u	$t(s)$	η_t (%)
10	1.34	4.6	1.41	217.0	2.1
18	1.37	6.8	1.40	846.7	0.8
35	1.39	12.6	1.40	3556.7	0.35

The upper level Pareto sets obtained with BDSD and the approach of Eichfelder are shown in Figures 14 and 15, respectively. A lower level frontier obtained with the approach of Eichfelder is shown in Figure 16. The approaches are compared with each other in Table 4.

In Problem 4, the evenness of the upper level Pareto set obtained with BDSD is a little better than that acquired with the approach of Eichfelder. However, BDSD drastically cuts down the optimization time by nearly 98% or more. The computational efficiency is improved by as much as two to three orders of magnitude.

6. Conclusion

A classical approach based on the DSD algorithm for bilevel multiobjective optimization problems has been proposed. The new Bilevel DSD approach breaks down the upper level optimization problem into a set of single-objective optimization problems on the upper level. To solve each of these single-objective optimization problems, a double-layer optimizer has been developed. In order to avoid generating the whole Pareto frontier on the lower level, the reference point coefficients vector on the lower level has been utilized as a design variable in the double-layer optimizer. Thus, the optimal lower level design variables can be found directly, and optimization time can be saved.

The approach has been tested on different cases and the performance has been compared with that of the algorithm by Eichfelder. It is shown that the BDSD approach can generate a quasi-evenly distributed Pareto set for the upper level with high computational efficiency. As the number of upper level Pareto points obtained rises, the computational efficiency essentially increases. In addition, for the bilevel problem with higher dimension of the upper level design variables, the BDSD approach becomes more effective and efficient. It is also worth noting that the proposed algorithm can easily be parallelized according to its structure.

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

The first author acknowledges support from the Chinese Scholarship Council (CSC).

References

Alizadeh, S. M., P. Marcotte, and G. Savard. 2013. "Two-Stage Stochastic Bilevel Programming over a Transportation Network." *Transportation Research Part B: Methodological* 58 (4): 92–105.

Assadipour, G., G. Y. Ke, and M. Verma. 2016. "A Toll-Based Bi-Level Programming Approach to Managing Hazardous Materials Shipments over an Intermodal Transportation Network." *Transportation Research Part D: Transport and Environment* 47: 208–221.

Coello Coello, C. A., and M. S. Lechuga. 2002. "MOPSO: A Proposal for Multiple Objective Particle Swarm Optimization." In *Proceedings of the 2002 Congress on Evolutionary Computation (CEC'02)*, Vol. 2, 1051–1056. Piscataway, NJ: IEEE.

Dad, I., and J. E. Dennis. 1997. "A Closer Look at Drawbacks of Minimizing Weighted Sums of Objectives for Pareto Set Generation in Multicriteria Optimization Problems." *Structural and Multidisciplinary Optimization* 14 (1): 63–69.

Dad, I., and J. E. Dennis. 1998. "Normal-Boundary Intersection: A New Method for Generating the Pareto Surface in Nonlinear Multicriteria Optimization Problems." *SIAM Journal on Optimization* 8 (3): 631–657.

Deb, K., A. Pratap, S. Agarwal, and T. Meyarivan. 2002. "A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II." *IEEE Transactions on Evolutionary Computation* 6 (2): 182–197.

Deb, K., and A. Sinha. 2009. "An Evolutionary Approach for Bilevel Multi-Objective Problems." In *Cutting-Edge Research Topics on Multiple Criteria Decision Making*, edited by Yong Shi, Shouyang Wang, Yi Peng, Jianping Li, and Yong Zeng, Vol. 35, 17–24. Berlin Heidelberg: Springer.

Deb, K., and A. Sinha. 2010. "An Efficient and Accurate Solution Methodology for Bilevel Multiobjective Programming Problems Using a Hybrid Evolutionary-Local-Search Algorithm." *Evolutionary Computation* 18 (3): 403–449.

Dehuri, S., A. K. Jagadev, and M. Panda, eds. 2015. *Multi-Objective Swarm Intelligence: Theoretical Advances and Applications*. Vol. 592 of the series Studies in Computational Intelligence. Berlin: Springer-Verlag.

Eichfelder, G. 2007. *Solving Nonlinear Multiobjective Bilevel Optimization Problems with Coupled Upper Level Constraints*. Technical Report Preprint No. 320, Preprint-Series of the Institute of Applied Mathematics, Universität Erlangen-Nürnberg, Germany.

Erfani, T., and S. V. Utyuzhnikov. 2011. "Directed Search Domain: A Method for Even Generation of the Pareto Frontier in Multiobjective Optimization." *Engineering Optimization* 43 (5): 467–484.

Erfani, T., S. V. Utyuzhnikov, and B. Kolo. 2013. "A Modified Directed Search Domain Algorithm for Multiobjective Engineering and Design Optimization." *Structural and Multidisciplinary Optimization* 48 (6): 1129–1141.

Erickson, M., A. Mayer, and J. Horn. 2001. "The Niched Pareto Genetic Algorithm 2 Applied to the Design of Groundwater Remediation Systems." In *Evolutionary Multi-Criterion Optimization*, edited by Eckart Zitzler, Lothar Thiele, Kalyanmoy Deb, Carlos Artemio Coello Coello, and David Corne, Vol. 1993, 681–695. Berlin: Springer-Verlag . doi:10.1007/3-540-44719-9_48.

Hettenhausen, J., A. Lewis, M. Randall, and T. Kipouros. 2013. "Interactive Multi-Objective Particle Swarm Optimisation Using Decision Space Interaction." In *Proceedings of the 2013 IEEE Congress on Evolutionary Computation*, 3411–3418. Piscataway, NJ: IEEE. doi:10.1109/CEC.2013.6557988.

Horn, J., N. Nafpliotis, and D. E. Goldberg. 1994. "A Niched Pareto Genetic Algorithm for Multiobjective Optimization." In *Proceedings of the First IEEE Conference on Evolutionary Computation and IEEE World Congress on Computational Intelligence*, 82–87. Piscataway, NJ: IEEE. doi:10.1109/ICEC.1994.350037.

Messac, A. 1996. "Physical Programming Effective Optimization for Computational Design." *AIAA Journal* 34 (1): 149–158.

Messac, A., A. Ismail-Yahaya, and C. Mattson. 2003. "The Normalized Normal Constraint Method for Generating the Pareto Frontier." *Structural and Multidisciplinary Optimization* 25 (2): 86–98.

Messac, A., and C. Mattson. 2002. "Generating Well-Distributed Sets of Pareto Points for Engineering Design Using Physical Programming." *Optimization and Engineering* 3 (4): 431–450.

Messac, A., and C. Mattson. 2004. "Normal Constraint Method with Guarantee of Even Representation of Complete Pareto Frontier." *AIAA Journal* 42 (10): 2101–2111.

Miettinen, K. M. 1999. *Nonlinear Multiobjective Optimization*. Boston, MA: Kluwer Academic.
Qu, Y., and Z. Jiang. 2013. "Bi-Level Programming Model and Taboo Search Algorithm in Industrial Location Under the Condition of Random Price." In *Emerging Technologies for Information Systems, Computing, and Management*, Vol. 236, 395–403. New York: Springer. doi:10.1007/978-1-4614-7010-6_45.
Sinha, Ankur, Pekka Malo, and Kalyanmoy Deb. 2015. "Towards Understanding Bilevel Multi-Objective Optimization with Deterministic Lower Level Decisions." In *Proceedings of the 8th International Conference on Evolutionary Multi-Criterion Optimization (EMO 2015)*, Part I, Vol. 9018, 426–443. Cham, Switzerland: Springer. doi:10.1007/978-3-319-15934-8_29.
Sinha, A., P. Malo, A. Frantsev, and K. Deb. 2013. "Multi-Objective Stackelberg Game Between a Regulating Authority and a Mining Company: A Case Study in Environmental Economics." In *2013 IEEE Congress on Evolutionary Computation (CEC)*, 478–485. Piscataway, NJ: IEEE. doi:10.1109/CEC.2013.6557607.
Sinha, A., P. Malo, A. Frantsev, and K. Deb. 2014. "Finding Optimal Strategies in a Multi-Period Multi-Leader–Follower Stackelberg Game Using an Evolutionary Algorithm." *Computers & Operations Research* 41: 374–385. doi:10.1016/j.cor.2013.07.010.
Srinivas, N., and K. Deb. 1994. "Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms." *Evolutionary Computation* 2 (3): 221–248. doi:10.1162/evco.1994.2.3.221.
Utyuzhnikov, S. V., P. Fantini, and M. D. Guenov. 2009. "A Method for Generating a Well-Distributed Pareto Set in Nonlinear Multiobjective Optimization." *Journal of Computational and Applied Mathematics* 223 (2): 820–841. doi:10.1016/j.cam.2008.03.011.
Zitzler, E., M. Laumanns, and L. Thiele. 2001. *SPEA2: Improving the Strength Pareto Evolutionary Algorithm*, Technical Report 103, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH).
Zitzler, E., and L. Thiele. 1999. "Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach." *IEEE Transactions on Evolutionary Computation* 3 (4): 257–271.