CrossMark

RESEARCH PAPER

Multiobjective and multi-physics topology optimization using an updated smart normal constraint bi-directional evolutionary structural optimization method

David J. Munk[1] · Timoleon Kipouros[2] · Gareth A. Vio[1] · Geoffrey T. Parks[2] ·
Grant P. Steven[1]

Abstract To date the design of structures using topology optimization methods has mainly focused on single-objective problems. Since real-world design problems typically involve several different objectives, most of which counteract each other, it is desirable to present the designer with a set of Pareto optimal solutions that capture the trade-off between these objectives, known as a smart Pareto set. Thus far only the weighted sums and global criterion methods have been incorporated into topology optimization problems. Such methods are unable to produce evenly distributed smart Pareto sets. However, recently the smart normal constraint method has been shown to be capable of directly generating smart Pareto sets. Therefore, in the present work, an updated smart Normal Constraint Method is combined with a Bi-directional Evolutionary Structural Optimization (SNC-BESO) algorithm to produce smart Pareto sets for multiobjective topology optimization problems. Two examples are presented, showing that the Pareto solutions found by the SNC-BESO method make up a smart Pareto set. The first example, taken from the literature, shows the benefits of the SNC-BESO method. The second example is an industrial design problem for a micro fluidic mixer. Thus, the problem is multi-physics as well as multiobjective, highlighting the applicability of such methods to real-world problems. The results indicate that the method is capable of producing smart Pareto sets to industrial problems in an effective and efficient manner.

✉ David J. Munk
   david.munk@sydney.edu.au

[1] The University of Sydney, Sydney, 2006, Australia

[2] University of Cambridge, Cambridge, CB2 1PZ, UK

1 Introduction

Engineering design often involves several considerations, usually with conflicting requirements, that cannot be simplified to a single objective function. In such cases, more than one solution may exist that meets the design goals. For these multiobjective problems, the Pareto frontier of the entire design space is the most valuable tool a designer can have to select the most appropriate designs. The Pareto frontier is defined as the set of all solutions for which no other solution is better in all objectives (Pareto 1964). Such solutions are known as Pareto-optimal. Therefore, Pareto sets give the trade-off relationships between the particular objectives in a multiobjective problem.

1.1 Pareto set generation methods

A variety of different algorithms for generating Pareto sets are to be found in the literature. This section summarises the most popular algorithms, comparing their efficiency and effectiveness in representing the design space. The interested reader is advised to seek out the latest review articles (Marler and Arora 2004; Ruzika and Wiecek 2005) on the topic, which provide in-depth surveys of multiobjective optimization methods.

The most common approach in the multiobjective optimization literature is the weighted sums method in which all objectives are combined to form a single function, known as the aggregate objective function. Following the introduction of the weighted sums method by Zadeh (1963), many authors have addressed its pitfalls with respect to identifying

the Pareto-optimal set, for example see Koski (1985), Stadler (1995), Athan and Papalambros (1996), Das and Dennis (1997) and Messac et al. (2000a). There are three main difficulties associated with the weighted sums method. First, although many methods for determining weights exist, a satisfactory, *a priori* selection of weights does not guarantee an acceptable final solution will be obtained (Marler and Arora 2010). In this case new weights need to be defined to resolve the problem. Messac and Hattis (1996) showed that weights must be a function of the original objectives, not constants, in order for a weighted sum to mimic a preference function accurately. Second, the weighted sums method is not able to capture solutions on the non-convex regions of the Pareto frontier. Theoretical derivations for this deficiency are given in Das and Dennis (1997) and Messac et al. (2000b). Finally, varying the weights consistently and continuously does not guarantee an even distribution of Pareto solutions and an accurate, complete representation of the Pareto set. However, the weighted sums method is easy to implement and does generate solutions of Pareto attribute.

The compromise programming method (Chen et al. 1999) has been proven to overcome some of the drawbacks of the weighted sums method; namely, it is able to find solutions on the non-convex regions of the Pareto frontier. Messac and Ismail-Yahaya (2001) showed that a relationship exists between the order of the aggregate objective function and that of the Pareto frontier for the compromise programming method to successfully generate Pareto solutions. Therefore, the success of the method is highly dependent on the order of the aggregated objective function. Nevertheless, although the compromise programming method is able to produce a Pareto frontier for convex and non-convex regions, it still shares some of the deficiencies associated with the weighted sums method. Most notably, when using an even distribution of weights it is unable to generate a set of evenly distributed solutions. Hence, the weighted sums and compromise programming methods are suitable for obtaining Pareto solutions, but ill-suited for the creation of Pareto sets (Messac et al. 2003).

The physical programming method was initially introduced by Messac and Hattis (1996) to provide a means of incorporating preferences without the need to define relative weights (Chen et al. 2000). Objective functions, constraints and goals are treated equally and combined into *design metrics*. Each design metric is associated with a type of utility function, which is distinguished by its general form, such as: monotonically decreasing, monotonically increasing or unimodal. The decision-maker specifies the numerical ranges, corresponding to different degrees of preference (i.e. desirable, tolerable, etc.), for each metric. As the design process evolves, these ranges may change. Messac et al. (2004) show that because of the way these

utility functions are defined, physical programming is able to effectively optimize objective functions with varying orders of magnitude. However, the requirement that the decision-maker needs to quantitatively classify different ranges of values for each metric suggests that physical programming requires significant knowledge about each objective and constraint before the optimization process. Nevertheless, the physical programming method is superior to the weighted sums and compromise programming methods, since it is able to represent the complete Pareto set with an even distribution of points (Chen et al. 2000; Messac 2000; Messac et al. 2001). Furthermore, Martinez et al. (2001) demonstrate the ability of the method to handle non-convex Pareto-optimal surfaces. Messac et al. (2001) prove that the physical programming method provides a sufficient condition for Pareto-optimality, while (Messac and Mattson 2002) demonstrate how physical programming can be used as a necessary condition for Pareto-optimality, obtaining all Pareto-optimal points. However, if an algorithmic implementation of the method is not available, then the application of physical programming becomes difficult. Physical programming also requires significant knowledge of the problem functions, since the decision-maker is required to specify large amounts of information. Thus, physical programming can incorporate more information about a problem; however, this comes at the cost of being inherently more involved compared with the weighted sums and compromise programming methods.

In an effort to overcome the deficiencies of the weighted sums approach, Das and Dennis (1998) proposed the Normal Boundary Intersection (NBI) method. This method introduced a new parameter that provided a means for obtaining an even distribution of Pareto-optimal points, even for a non-convex Pareto set, provided a consistent variation in this parameter is defined. Although the NBI method has been shown to generate an even distribution of Pareto solutions representing the complete Pareto set, the method does not provide a sufficient condition for Pareto-optimality and therefore may produce non-Pareto-optimal points. Further, for multiobjective problems of dimension $n > 2$ the NBI method overlooks some Pareto-optimal points, i.e. it does not explore the full design space. Das (1999) proposed a modified NBI method whereby more Pareto points in the nonlinear portions of the Pareto surface are generated.

Messac et al. (2003) developed the Normal Constraint (NC) method as an alternative to the NBI method with further improvements. The authors showed that when the normal constraint method is used with normalized objective functions, often referred to as the Normalized Normal Constraint (NNC) method, and with a Pareto filter, which eliminates non-Pareto or locally Pareto-optimal points, this approach provides a set of evenly spaced Pareto-optimal

points in the criterion space. Its performance is independent of design objective scales, and it has been shown to be more computationally stable than the NBI method, and less likely to produce non-Pareto or locally Pareto-optimal solutions. Messac et al. (2003) show that the NC method overcomes the deficiencies of the NBI method because of the different structure of their formulations. The former uses inequality constraints whereas the latter uses equality constraints. The method used in this paper to generate the Pareto sets is based upon the NC method. Hence, the specific literature concerning improvements in this method is discussed in Section 1.2.

1.2 Normal constraint method

Since the inception of the NC method (Ismail-Yahaya and Messac 2002) many variants and improvements have been proposed. Shortly after introducing the method the original authors proposed the NNC variant, which alleviates objective scaling issues by operating on the normalized design objective space (Messac et al. 2003). A number of authors have suggested means for improving the distribution of Pareto solutions by modifying the utopia plane (Martinez et al. 2007; Martinez et al. 2009a; Motta et al. 2012). Furthermore, methods for avoiding local optima have been proposed by Martinez et al. (2009b). The authors propose a hybrid algorithm, which combines the NNC method with genetic algorithms.

The NNC method has been shown to capture the entire Pareto frontier for problems of dimension $n = 2$ (Messac et al. 2003). However, similarly to the NBI method, parts of the design space could be left unexplored for problems of higher dimension (Section 1.1). Messac and Mattson (2004) proposed that, for problems where $n > 2$, the utopia plane should be extended to include not just the region bounded by anchor points, but rather the entire region of the utopia plane that could produce a Pareto point. This extended region is bounded by the anchor points as well as the perpendicular projections of the anti-anchor points. Without the use of this extended region for the utopia plane, one cannot guarantee that the generated set will represent the complete Pareto frontier for problems with $n > 2$.

A design space containing a disjointed Pareto set may yield the same Pareto point, even when performing multiple single-objective optimizations. Boyce and Mattson (2008) proposed a method for identifying which utopia plane points will produce redundant Pareto points, to avoid these single-objective optimization runs. This is achieved by recognizing when at least one of the normal constraints used in generating a point is not active. All the utopia plane points that lie in the region between the normal constraint that first generated the replicated point and the parallel normal constraint that would be generated directly through the given point can then be removed.

Martinez et al. (2007) proposed the uniform NNC method. This method uses the distribution of known Pareto points to help guide the NNC method in finding a new set of Pareto solutions. The authors showed that this improved on the NNC method's ability to generate a more uniformly distributed Pareto frontier. This idea was recently taken further by Hancock and Mattson (2013), developing the smart normal constraint (SNC) method. Additional linear constraints, known as smart constraints (Haddock et al. 2008), are used to determine which approximate point is most likely to produce a smart Pareto solution. The authors show that the SNC method alleviates the need for a Pareto filter to generate smart Pareto sets. This improves the computational expense of the algorithm, since the Pareto filter first generates many solutions, then reduces the set by removing solutions that are considered insignificantly different from other Pareto points (Hancock and Mattson 2013). Such an approach is computationally inefficient as a large number of designs, which are later removed from consideration, are generated. Hancock and Mattson (2013) apply the SNC method to three numerical examples from the literature: the TNK problem (Tanaka et al. 1995), a gear box design (Huang et al. 2006) and the WATER problem (Ray et al. 2001). While these examples do demonstrate the effectiveness and efficiency of the SNC method in generating smart Pareto sets, they all have objective functions that depend only on a small number of design variables (2, 7 and 3, respectively). In all three cases, no sensitivity analysis of the system is required since the objectives are given as continuous functions of the design variables. These problems therefore represent a niche set, with a restricted problem size.

In contrast, topology optimization has become a highly developed tool, which is extensively used in the mechanical, automotive and aerospace industries (Sigmund 2011). Gradient-based topology optimization has been shown to efficiently solve fine-resolution problems with thousands or even millions of design variables with only a few hundred function evaluations. However, the topology optimization literature shows a lack of multiobjective algorithms, limiting their application to single-objective problems.

In this paper, we propose a multiobjective gradient-based topology optimization algorithm which uses an updated SNC method and couples it with a Bi-directional Evolutionary Structural Optimization (BESO) algorithm. The BESO algorithm uses the method of Lagrange multipliers to convert the constraints into continuous variables such that multiple objectives can be considered. The SNC method is modified to take advantage of BESO's ability to minimize an objective while satisfying all the constraints. The second illustrative application presents a multi-physics problem, where the objectives are from different disciplines. This highlights the method's ability to tackle real-world

problems, which are characterized by multiple conflicting objectives and often multiple disciplines, especially in the aerospace and automotive industries. It must be noted that, in real-world design, multiple constraint problems are unavoidable, with a common example being the maximum stress of a structure not exceeding its limit value. However, there are different types of constraints, namely geometrical and physical, which are each handled in a different way. Usually, one treats physical constraints as penalty terms in the calculation of the objective function. However, this study focuses on multiple objectives instead of constraints. Nevertheless, the method proposed in this article could treat physical constraints by adding penalty terms in the objective functions accordingly. Furthermore, geometrical constraints can be handled even before the calculation of the sensitivities. This is already demonstrated in this study for volume and non-design constraints - parts of the design domain which are constrained to be either solid or void. Furthermore, this approach facilitates the use of high-fidelity analysis methods in the conceptual design stage for real-world problems.

1.3 A quick history of topology optimization

The field of topology optimization was born over a century ago with the publication of a paper that derived the optimality criteria for the least weight layout of trusses (Michell 1904). This did not spark immediate interest, since the first general theory of topology optimization, known as optimal layout theory, was not formulated until some 70 years later by Prager and Rozvany (1977), and it was the seminal paper by Bendsoe and Kikuchi (1988), which developed the first material distribution method, that revolutionized the field of structural optimization, making it applicable to real-world engineering problems. Two methods, namely, Solid Isotropic Material with Penalization (SIMP) (Bendsoe 1989; Rozvany et al. 1992) and BESO (Xie and Steven 1993; Yang et al. 1999) have now reached the stage of application in single-objective industrial problems (Rozvany 2009). This work is concerned with the latter, proposing a method for effectively and efficiently producing smart Pareto sets for multiobjective topology optimization (MOTO) problems.

Compared to other types of structural optimization, topology optimization of continuum structures, through material distribution methods, is the most challenging; however, it is the most rewarding economically as there are no restrictions on the design. The main idea is to find the optimal distribution of the material in a predefined design domain considering an objective function and constraints. Since its introduction in 1988 (Bendsoe and Kikuchi 1988), topology optimization through material distribution methods have seen an exponential increase in publications (Munk

et al. 2015). A wide variety of objective functions have been used with topology optimization algorithms, diversifying their application to almost all fields of engineering and design (Sigmund 2001; Steven et al. 2000). However, compared with the extensive research on single-objective optimization, there has been significantly less work concerned with topology optimization for multiobjective problems. The most recent topology optimization review articles have highlighted this gap in the literature (Rozvany 2009; Sigmund and Maute 2013; Deaton and Grandhi 2014; Munk et al. 2015), with only one or two references concerning multiobjective problems and no section dedicated to the topic. This lack of literature is also seen in the latest books on topology optimization (Bendsoe and Sigmund 2004; Huang and Xie 2010; Rozvany and Lewinski T 2013) with no reference to multiobjective optimization problems. Recently, Sigmund and Maute (2013) identified the handling of multiple constraints as one of the main future challenges of topology optimization.

Meta-heuristic algorithms have, thus far, dominated multiobjective methods, as shown by the latest review articles (Zhou et al. 2011; Kunakote and Bureerat 2011) and books (Coello et al. 2007; Deb 2009) in this field. For problems with many variables, such as those in topology optimization, these techniques require orders of magnitude more function evaluations for low resolution, coarse mesh, problems (Sigmund and Maute 2013; Sigmund 2011). These methods solely rely on objective function values, based on random processes, to search the design space and progress the solution. Thus, the handling of multiple objectives can be achieved in an easy, non-intrusive, manner. Comparatively, topology optimisation algorithms compute the gradients of the objective functions to determine the sensitivities of the objectives to each design variable. Therefore, the added difficulty is how one should combine the different gradients for the different objectives such that the desired trade-off in each objective can be found. Moreover, once the method of determining the sensitivities of multiple objectives is found, a rigorous convergence criterion must be established. In the literature, especially for discrete methods, convergence of topology optimisation algorithms has been shown to be difficult (Munk et al. 2017). Therefore, the addition of multiple objectives only makes convergence harder. The development of gradient-based topology optimization algorithms that can solve multiobjective problems would therefore be of considerable benefit.

Early efforts were made to extend the Evolutionary Structural Optimization (ESO) method to encompass multiobjective problems (Proos et al. 2001a). The authors used a weighted sums method and a global criterion method to incorporate multiple criteria into the ESO process, showing that this approach is able to produce a range of options, of Pareto attribute, for a multiobjective problem. However,

the weighted sums method has been criticized for its deficiencies in depicting the Pareto-optimal set (Section 1.1). The global criterion method is only able to produce a single result, usually similar to that produced with equally weighted objectives (Proos et al. 2001a), thus the Pareto set is not defined. The same authors have also applied this method to two different objectives (Proos et al. 2001b), namely, minimum compliance and maximum specific inertia. Kim et al. (2006) developed a multiobjective structural optimization method for a three-dimensional (3D) thermal protection system design using an ESO algorithm. They again used a weighted sums method with the objective of minimizing maximum thermal stress and maximizing the fundamental frequency.

This review shows the small amount of literature available on applying ESO methods to multiobjective problems. Only the weighted sums method has been successfully added to the ESO method. Only the conventional ESO algorithm, i.e. a monotonic algorithm, has been applied to multicriteria problems, thereby avoiding convergence issues, since the solution is found once a certain number of elements have been removed.

This paper aims to extend this work on multiobjective ESO algorithms in two main ways. First, by using a more rigorous BESO, rather than ESO, algorithm. Modern BESO algorithms are convergent and mesh-independent (Huang and Xie 2007), simultaneously removing and including material in the design domain until the constraints and a convergence criterion are satisfied. Second, by adopting a multiobjective algorithm which can more efficiently and effectively represent the Pareto frontier. Thus far, only the weighted sums method has been adopted. While this method does produce Pareto-optimal solutions, it is not able to identify solutions on non-convex regions of the Pareto frontier. It is also unable to generate an evenly distributed Pareto front or guarantee a smart Pareto set will be obtained. Therefore, the multiobjective algorithm adopted for this work is a modified SNC method, which has been shown to capture the entire design domain (Messac and Mattson 2004), produce evenly distributed Pareto solutions (Martinez et al. 2007) and efficiently obtain a smart Pareto set (Hancock and Mattson 2013).

The remainder of this paper is organized as follows: Section 2 outlines the multiobjective optimization problem formulation and the SNC method for generating a smart Pareto set. Here the update to the original algorithm (Hancock and Mattson 2013) is presented. This is followed in Section 3 by a review of the BESO method and how it is applied to the multiobjective formulation. In Section 4 the performance of the SNC-BESO method is illustrated through two numerical examples, the latter being a real-world multi-physics engineering design problem. Finally, Section 5 concludes the paper.

2 The smart normal constraint method

In this section the mathematical definition of a multiobjective optimization problem (MOP) is introduced. The original SNC method is then outlined. Finally, the update to the SNC method used in this analysis is given. For an in-depth analysis of the original SNC and NC methods the reader should seek out (Hancock and Mattson 2013) and Ismail-Yahaya and Messac (2002), respectively.

2.1 Multiobjective optimization

A generic MOP is usually stated mathematically as:

$$\begin{aligned}
\min(\mathbf{x}) \quad & \{\mu_1(\mathbf{x}), \mu_2(\mathbf{x}), \ldots, \mu_n(\mathbf{x})\} \\
\text{subject to: } & n \geq 2 \\
& \mathbf{g}(\mathbf{x}) \leq 0 \\
& \mathbf{h}(\mathbf{x}) = 0 \\
& \mathbf{x}_l \leq \mathbf{x} \leq \mathbf{x}_u
\end{aligned} \tag{1}$$

where $\mathbf{x}$ is the design variable vector, such that the vectors $\mathbf{x}_l$ and $\mathbf{x}_u$ are the corresponding lower and upper bounds of the design variables. μ is the design objective vector, $\mathbf{g}$ and $\mathbf{h}$ are inequality and equality constraint vectors, respectively, and n is the dimension of the problem or number of objectives. Solution of the problem (1) produces a Pareto set of optimal solutions. Section 2.2 reviews the SNC method, which is used in this work to generate the Pareto set.

There exist two important types of points in the design space of the MOP which are pivotal to the understanding of the SNC method. Therefore, they are first defined for clarity.

Anchor points are the points in the feasible design space, the subset of designs satisfying the constraints of the MOP (1), that correspond to the minimum value of one of the objectives. The anchor point for the ith objective is given by:

$$\mu^{i*} = \left[\mu_1(\mathbf{x}^{i*}), \mu_2(\mathbf{x}^{i*}), \ldots, \mu_n(\mathbf{x}^{i*}) \right]^T \tag{2}$$

where $\mathbf{x}^{i*}$ is defined as the design variable vector that gives the minimum value for the ith objective.

Anti-anchor points are the opposite of anchor points in that they correspond to the points in the feasible design domain with a maximum value of one of the objectives. The anti-anchor point for the ith objective is thus defined as:

$$\mu^{i\circ} = \left[\mu_1(\mathbf{x}^{i\circ}), \mu_2(\mathbf{x}^{i\circ}), \ldots, \mu_n(\mathbf{x}^{i\circ}) \right]^T \tag{3}$$

where $\mathbf{x}^{i\circ}$ is the design variable vector that gives the maximum value for the ith objective.

2.2 Review of the smart normal constraint method

The SNC method converts the MOP into a series of single-objective optimization (SOO) problems, with additional,

different, linear constraints calculated to produce a Pareto solution in a particular region of the design space. The SNC method can be divided into 7 steps, which will be outlined in this section, using a bi-objective optimization problem (Fig. 1). For problems with $n > 2$, the lines described in these steps are replaced by their higher dimensional counterparts, i.e. planes for $n = 3$ or hyperplanes for higher dimensions. Steps 2–7 are repeated until there are no more approximate regions of the Pareto surface that are capable of yielding a smart Pareto point. Further details on the method can be found in Hancock and Mattson (2013).

Step 1: Generating the reference points

First, the anchor and anti-anchor points must be located inside the design domain. These are used as the vertices on the edges of the Pareto frontier approximation to guarantee convergence of the entire Pareto set (Messac and Mattson 2004). Using (2) and (3) the anchor and anti-anchor points are found, respectively.

Step 2: Connecting the approximation vertices

The vertices created by the anchor and anti-anchor points are divided into approximation segments or planes to approximate the Pareto frontier. For the bi-objective case, each vertex is connected to the neighbouring vertices on either side of it, as shown in Fig. 1. For problems with $n > 2$, the connectivity of approximation vertices is found by linearly projecting them onto the utopia plane and finding the Delaunay triangulation (Barber et al. 1996) of the projected set.

The lines that connect the anchor points are known as the utopia lines. Therefore, a utopia line vector, N_j, is found using the equation:

$$N_j = \mu^{j*} - \mu^{n*} \qquad \forall j \in (1, 2, \ldots, n-1) \qquad (4)$$

Hence, $n - 1$ utopia line vectors are defined, all of which point to the anchor point corresponding to the nth dimension, μ^{n*}.

Step 3: Approximating the Pareto frontier

Evenly spaced approximation points (Fig. 1) are generated along each approximation plane, through the following relation:

$$S_i = \sum_{j=1}^{n} \alpha_i^j P_j \qquad (5)$$

where S_i is the ith approximation point and P_j is the jth approximation vertex. The non-dimensional parameter α_i^j satisfies the constraints given by:
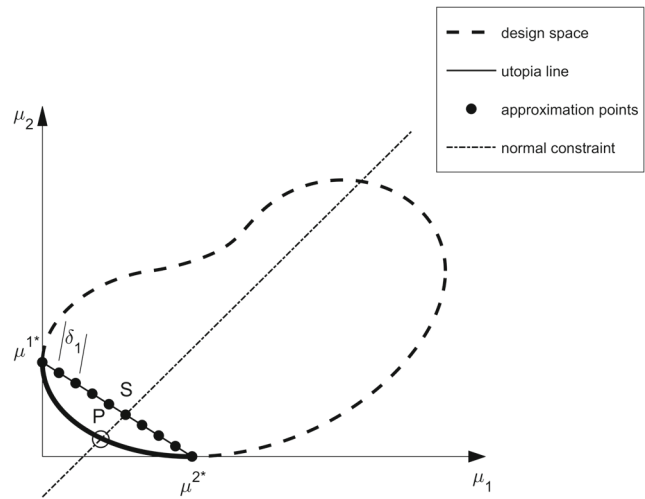
$$0 \le \alpha_i^j \le 1 \qquad (6)$$

Fig. 1 Initial set-up of the SNC method before constrained SOO for a bi-objective problem

and

$$\sum_{j=1}^{n} \alpha_i^j = 1 \qquad (7)$$

α^j is varied from 0 to 1 with a fixed increment of δ_j such that an even distribution of approximation points over the entire Pareto frontier is obtained. Traditionally the value of δ_j was arbitrary and only depended on how close to each other the designer wanted the approximation points to be. However, close approximation points may result in several converging to the same region of the Pareto frontier, and hence being discarded. Hancock and Mattson (2013) found that, in practice, it is simple and effective to set δ_j equal to the shortest Euclidean distance between a centre point and the Practically Insignificant Trade-off (PIT) region that defines one smart distance around it. Further discussion of the PIT region and smart distances is given in Step 5 (Section 2.2). Thus, the increment δ_j can be found by:

$$\delta_j = \left\| \min_{\mathbf{d}} ||\mathbf{d}|| \right\| \qquad (8)$$

where $\mathbf{d}$ is a vector of distances between the centre point of the PIT region and any second point on the boundary of the PIT region. Here d is constrained by setting the smart distance to unity, $s = 1$.

A smaller increment, δ_j, will result in more approximation points. Computations performed on the approximation points are relatively inexpensive, as more approximation points will not result in more functional calls, because only one point is used per iteration. Therefore, in contrast to the NC method, the efficiency of this algorithm depends little on the value of δ_j.

Step 4: Removing restricted approximation points

Unavoidably, some SOOs will result in the generation of solutions that will not produce new smart Pareto points. There may also be regions where the Pareto frontier is discontinuous and, therefore, solutions cannot exist. In Step 7 (Section 2.2), these *restricted* regions are recorded, and avoided in further iterations. In Step 4 approximation points that lie within one smart distance of already existing smart Pareto points are removed from further consideration.

Step 5: Calculating the smart distances

In this step the smart distance between each approximation point and all the existing approximation vertices is found. Mattson et al. (2004) first introduced the idea of a smart Pareto set, based on the assumption that: *"when the trade-off is significant a designer is willing to give up an insignificant amount in one objective to gain significantly in another"*. To achieve such a set, Mattson et al. (2004) introduced the smart Pareto filter, use of which has been shown to be computationally prohibitive (Section 1.2). The fundamental concept of the smart Pareto filter is to remove any duplicate Pareto solutions which fall inside a user-defined shape, the PIT region, surrounding each Pareto solution. The PIT region is defined by two control parameters, $\Delta\mathbf{r}$ and $\Delta\mathbf{t}$, which determine the significant amount of change in an individual objective and the trade-off amount that is considered insignificant, as shown in Fig. 2a. One advantage of this method is that it can be used alongside any algorithm capable of producing a well-distributed Pareto set.

In the SNC method, the direct generation of a smart Pareto set is facilitated by the smart distance between points in the design space. The shape of the PIT region around a point is called a Lamé curve in 2D or a hyper-Lamé curve in nD (Fig. 2b).

Therefore, the PIT region is defined as the area that lies on or within the Lamé curve (Fig. 2b). All points inside the PIT region have a smart distance $s \leq 1$ from the centre point. Since, by definition all members of a smart Pareto set do not lie within the PIT regions of any other member of the set, each will have a smart distance $s > 1$ with respect to all other members of the set. The smart distance between two points is found by the following formula:
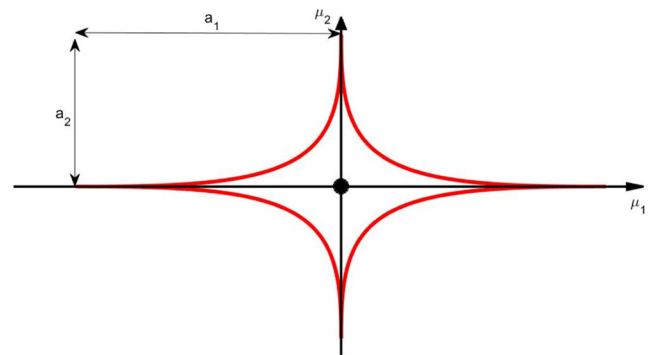
$$s = \|\mathbf{Ad}\|_p \quad \text{for } (0 \leq p \leq 2) \tag{9}$$

where:

$$\mathbf{A} = \begin{bmatrix} \frac{1}{a_1} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \frac{1}{a_n} \end{bmatrix} \quad \text{for } (\mathbf{a} > 0) \tag{10}$$
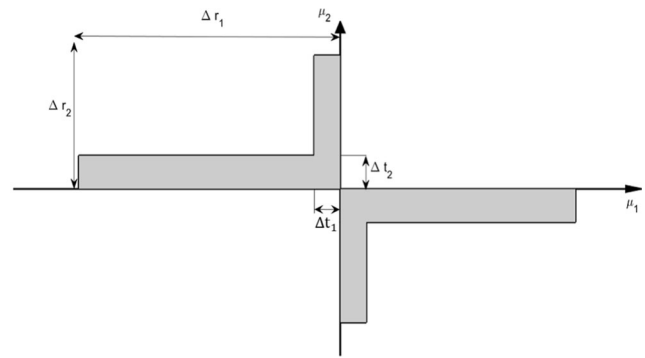
(a) Using the smart Pareto filter (Mattson et al (2004))

(b) Using the smart distance formulation

Fig. 2 The user-defined PIT region

$\mathbf{d}$ is a vector between the two points and $\|\mathbf{Ad}\|_p$ is the p-norm of the vector $\mathbf{Ad}$, which follows that given by Rynne (2007), where for this case it is given by:

$$\|\mathbf{Ad}\|_p = \left(\sum_{i=1}^{n} |A_{i,i} d_i|^p \right)^{\frac{1}{p}} \tag{11}$$

The variables $\mathbf{a}$ and p determine the distribution of the smart Pareto points that will be generated for a particular problem. The values a_i, which make up the diagonal of matrix $\mathbf{A}$, correspond to the ith objective and can be considered as the amount of change in the ith objective that would constitute a significant difference between two points if all other objectives remain practically unchanged. Therefore, any Pareto point that lies within a distance a_i of another Pareto point and does not have a significant trade-off in one or more other objectives will fall within the PIT region and be discarded. Larger values of the matrix $\mathbf{A}$ will thus correspond to fewer points in the smart Pareto set. The value of p determines the amount of curvature of the PIT region, and hence controls the amount of trade-off between objectives that is required in order for two points near each other to both remain in the smart Pareto set. The SNC method will work for values of p between 0 and 2; however, it is assumed that for most cases: $0 \leq p \leq 1$, since this will

result in a trade-off that is equal to or less than the significant change in a single objective. Since the values for **a** and *p* are user-defined, it is therefore the user's preferences that determine the distribution of that set.

This method of smart distances is unique in that it only requires a single scalar value, the smart distance, to define a PIT region. This allows the algorithm to identify whether or not a new point is a smart Pareto point. It can also determine to what extent the point is smart. Thus, the SNC method is able to search the design space more efficiently and determine which point is most likely to generate a new smart Pareto solution. It is this ability that makes the SNC method more efficient than the NC method, which iteratively cycles through all the utopia line points.

Step 6: Generating the new Pareto point

The approximation point with the largest smart distance to its nearest known Pareto point is selected to construct the SOO problem given by:

$$
\begin{aligned}
\min(\mathbf{x}) \quad & \mu_1(\mathbf{x}) \\
\text{subject to:} \quad & n \geq 2 \\
& \mathbf{g}(\mathbf{x}) \leq 0 \\
& \mathbf{h}(\mathbf{x}) = 0 \\
& \mathbf{x}_l \leq \mathbf{x} \leq \mathbf{x}_u \\
& N_j(\mu(\mathbf{x}) - S_j)^T \leq 0 \quad \forall j(1, 2, \ldots, n-1)
\end{aligned}
\tag{12}
$$

This SOO problem has an additional linear constraint (12), which excludes all points found above the line that intersects the approximation point and is orthogonal to the approximation line. Thus, for every approximation point that is deemed likely to produce a smart Pareto solution, a corresponding point on the Pareto frontier is found. Figure 1 shows the Pareto point *P* that is produced by solving (12) using the approximation point *S*. In this case, the approximation points are closer to the Pareto front than the utopia line points that are obtained with the NC method. This usually results in fewer function calls per SOO for the SNC method compared to the NC method.

Step 7: Confirming the new Pareto point belongs to the smart Pareto set

The new Pareto point, found by solving the SOO problem of (12), may not lie in the smart Pareto set. Therefore, if this is the case, a restriction enabling the removal of future approximation points in these regions, which are known to be unable to produce smart Pareto points, must be added. Hancock and Mattson (2013) identify three criteria to test whether the new point lies inside the smart Pareto set. If the point meets one of these criteria, then it is not a smart Pareto point.

First, a dominated point may be produced by solving the topology optimisation problem of (12) when there are local minima or maxima in the design space. When using gradient-based algorithms, local optima can be perceived as global optima by the optimizer. A dominated solution is one which is locally optimal, but not globally optimal, since there exists at least one other solution where one of the objective functions can be improved in value, compared to the dominated point, without degrading the other objective values. If any of the other objective values deteriorate, then the points are Pareto-equivalent - neither dominates the other. For one point to dominate another, it needs to be better with respect to at least one objective and not worse with respect to any other objective. Therefore, a solution is called Pareto-optimal if there does not exist another solution that dominates it. These solutions can be identified and omitted with a Pareto filter, as suggested by Messac et al. (2003). By passing the new points through a Pareto filter, the algorithm can avoid using dominated points as approximation vertices. Including such points would decrease the accuracy of its approximation of the true Pareto frontier.

Second, a redundant point can be produced when the new solution falls within the PIT region of another, already present, Pareto point. This may occur because the true shape of the Pareto frontier is unknown, rather it is just approximated by the already existing Pareto solutions that have been obtained. Therefore, simply by selecting an approximate point which does not lie inside the PIT of another Pareto point does not guarantee, once the optimisation problem has been solved (12), that the obtained Pareto point also won't lie inside the PIT of another Pareto point. If this occurs, the new Pareto point can still be used as an extra approximation vertex to better approximate the Pareto frontier for future points; however, it cannot be kept in the smart Pareto set. No approximation point that lies along the normal constraint line that resulted in a redundant solution can be used for future SOOs. The reader is directed to Fig. 6 for an example of a redundant point being produced by the original SNC method.

Finally, a point that is separated from the normal constraint lines or planes, which were used in the SOO (12) that created it, is referred to as a separated point. This separation indicates that there is a region in the design space in which all SOOs will converge to the same solution. This occurs when the Pareto frontier is discontinuous in this region. Using the normal constraint line that produced the separated point and a parallel normal constraint line that intersects the final solution, a restricted region can be created. This restricted region is kept for the remainder of the optimization process, to avoid the generation of redundant points.

Thus, each new solution is a smart Pareto point if it is not dominated, redundant, separated or any combination

there of. For more detail on the restrictions of smart Pareto sets, the reader is advised to consult (Hancock and Mattson 2013).

2.3 Updated smart normal constraint method

Here the modifications to the SNC method (Hancock and Mattson 2013) that improve its coupling with the BESO algorithm (Section 3) are described. The aim is to generate smart Pareto points in a computationally efficient way. The method is graphically represented in Fig. 3.

Figure 3a shows that the anchor points have been found and a line connecting them created. The line is divided equally into a series of approximation points. This is the main difference between the SNC and NC methods. For the SNC method, the line represents an approximation of the Pareto frontier, which, in the first iteration, is clearly not a good approximation. However, as the solution progresses the approximation becomes more accurate, improving the method's ability to locate smart Pareto points.

The main difference between the original SNC method (Hancock and Mattson 2013) and the one used in this work is also illustrated in Fig. 3a. Hancock and Mattson (2013) keep the normal constraint of the SOO problem (12) the same as that used in the NC method. However, in this work, the normal constraint is modified to take advantage of the BESO algorithm's ability to satisfy multiple constraints while minimizing the objective. Instead of defining a large region, approximately half the design space for the first iteration (Fig. 1), a small band (the shaded area in Fig. 3a) is defined, using the following two normal constraints:

$$N_j(\mu(\mathbf{x}) - S_r)^T \leq 0 \quad \forall j(1, 2, \ldots, n - 1) \tag{13}$$

and

$$-N_j(\mu(\mathbf{x}) - S_l)^T \leq 0 \quad \forall j(1, 2, \ldots, n - 1) \tag{14}$$

where S_r and S_l are the approximation points on either side of the approximation point that is determined to be most likely to produce a smart Pareto solution (Section 2.2 Step 6). In this way, the solution is guaranteed to fall at the intersection of the Pareto curve and the normal line, which intersects the approximation point that is most likely to produce a smart Pareto solution. Therefore, this method is less likely to produce dominated or redundant points, since the solution must fall inside the constrained region. However, there is one restriction that is not automatically handled by the updated SNC method. Separated points may still occur if the Pareto frontier is discontinuous over the entire constrained region. Furthermore, it is possible for a local optimum to exist inside the narrow region defined by the two constraints (13) and (14). Therefore, a dominated point in this region could still be produced. This is unavoidable

when using gradient-based methods, since local optima can be obtained. However, the BESO method ranks all new solutions against the previously determined ones. Therefore, if a local optimum is found, it is removed from the Pareto set and the constrained region that produced it is omitted for the entire duration of the optimization process.

Separated points are handled using the conventional SNC method, i.e. by identifying solutions that converge to the same point even when different approximation points are used (Section 2.2). In the updated method, there are two ways of identifying a separated point. First, if the Pareto frontier is discontinuous, but the design space is not, then a solution will be found that is locally Pareto-optimal. If this occurs, then the point is not added to the smart Pareto set, and the constrained region that was used to find that solution is omitted for the entire duration of the optimization process. Second, if the design space happens to be discontinuous over the entire constrained region, then no solution will exist to the SOO problem with the constraints given by (13) and (14). Alternatively, with the addition of an extra constraint it is possible that the feasible design space can become limited such that the algorithm struggles to converge. In both these cases, the objective will oscillate back and forth over either side of the discontinuity, and hence the constrained region as well. This is found by tracking the objective and stopping the solution if this oscillatory behaviour is observed. Again, the constrained region is omitted for the entire duration of the optimization process.

Figure 3b shows how the approximation of the Pareto frontier is improved after the first Pareto point is found. Now the approximation curve is made up of two segments of piecewise linearly varying points, instead of one as used in the NC method. This updated approximation of the Pareto frontier provides the algorithm with more information about where the remainder of the smart Pareto set is most likely to be found. Before each SOO (12) the nearest Pareto point, in terms of its smart distance, to each approximation point is found. The approximation point that has the largest smart distance to its nearest known Pareto point is identified as the most likely point to produce the next smart Pareto solution. The constrained region is then constructed by using the points on either side of this approximation point (S_l and S_r). An SOO is then performed with the normal constraints defined by (13) and (14). This process is then repeated multiple times iteratively to construct a more accurate approximation of the Pareto frontier.

Figure 3c shows the smart Pareto set obtained once no approximation point has a minimum smart distance of $s > 1$ from any discovered Pareto points. Once this occurs, the algorithm terminates and the smart Pareto set has been found.

The strategy of approximating the Pareto frontier through an approximation curve is identical to that used in the original
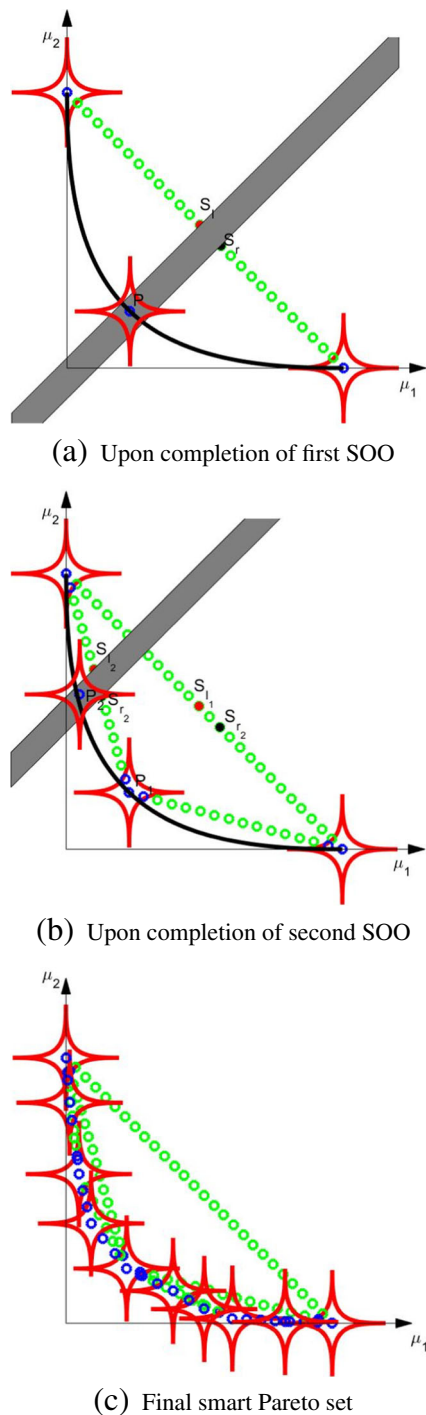
(a) Upon completion of first SOO

(b) Upon completion of second SOO

(c) Final smart Pareto set

Fig. 3 The updated SNC method for a bi-objective case

SNC method (Hancock and Mattson 2013). Likewise, the general method for creating individual Pareto solutions, i.e. solving a constrained SOO problem, is the same for the NC and SNC method used here. However, in this work, the constraint is tightened by defining a narrow band, through two parallel normal constraints, in which the solution must lie. Therefore, the solution of the SOO does not have to

be passed through a Pareto filter, nor does the smart distance from the current solution to the already determined Pareto points need to be calculated, to determine if the solution is dominated or redundant. These restrictions are actively handled in the optimization process. Thus, the SNC method used here retains all the improvements of the original (Hancock and Mattson 2013), since the algorithm dynamically adjusts the spacing between the constructed constraint regions according to the final distribution, as described by the user through the definition of the PIT regions.

Finally, as with all optimisation algorithms, a poor initial parameter selection may unavoidably result in the generation of unfavourable solutions. However, this is not new to topology optimisation, nor specific to it, and will be automatically handled by Step 7 (Section 2.2) where the solution's Pareto optimality is assessed.

3 The bi-directional evolutionary structural optimization method

This work uses the BESO algorithm to solve the SOO (12) of the SNC method. In this section the developments of the algorithm are outlined, followed by the method of Lagrange multipliers, which is used to implement the multiple constraints of the SOO problem.

3.1 Developments of the BESO algorithm

BESO is a variant of ESO with a bi-directional formulation, i.e. elements can be included in, as well as removed from, the design space. The idea behind the original ESO algorithm (Xie and Steven 1993) was that by gradually removing *inefficient* material from an initial, over-sized, structure; the design would then evolve towards the optimum. However, this monotonic algorithm is limited in two main ways: first, only material can be removed from the structure, and consequently the initial model must be significantly over-designed. Second, if structure is prematurely removed, it cannot be recovered. Thus far, only the ESO algorithm has been applied to MOTO problems (Section 1.3); therefore, these deficiencies are present in the algorithms used.

Modern BESO algorithms are mesh-independent and convergent (Huang and Xie 2007), removing and including material in the design domain until the constraints and a convergence criterion are satisfied. Recently, a further improvement of the BESO algorithm introduced the use of *soft* material, to model void elements (Huang and Xie 2009). This method, known as *soft-kill* BESO, is used in the work presented in this paper. The SOO (12) can be transformed into the general topology optimization problem for

soft-kill BESO, implementing the updated constraints ((13) and (14)) with two objectives, by:

$$
\begin{aligned}
\min(\mathbf{x}) \quad & \mu_1(\mathbf{x}) \\
\text{subject to:} \quad & [\mathbf{K}]\mathbf{u} = \mathbf{f} \\
& \sum_{i=1}^{N_e} \leq V \\
& N_1(\mu(\mathbf{x}) - S_l)^T \leq 0 \\
& -N_1(\mu(\mathbf{x}) - S_r)^T \leq 0 \\
& \mathbf{x} = [x_{min}, 1]
\end{aligned}
\tag{15}
$$

where $\mathbf{f}$ and $\mathbf{u}$ are the nodal force and displacement vectors, respectively, and $[\mathbf{K}]$ is the global stiffness matrix of the structure. V is a predefined volume fraction and N_e is the total number of elements in the model design space. Therefore, this part of the problem formulates the volume constraint. The design variables $\mathbf{x}$ are discrete, where $x_i = x_{min} = 10^{-4}$ represents void material and $x_i = 1$ defines solid material.

3.2 Sensitivity analysis

In this study, three different objectives are considered in the test cases presented in Section 4. The first objective is minimum compliance or maximum stiffness, used for stiffness optimization. In finite element analysis the change in the stiffness of the structure due to the removal of an element is equal to the element strain energy (Chu et al. 1996). This change is defined as the element sensitivity for compliance minimization:

$$
\alpha_e^{cmp} = \frac{\partial c}{\partial x_i} = \frac{1}{2} p x_i^{p-1} \mathbf{u}_e^T [\mathbf{K}]_e^0 \mathbf{u}_e
\tag{16}
$$

where c is the compliance, $p = 3$ is the penalization factor, a superscript of 0 indicates solid values and a subscript of e represents elemental values. The element sensitivity (16) takes advantage of the SIMP material model (Bendsoe and Sigmund 1999), where the Young's modulus, E, is defined by the following power law penalization:

$$
E(x_i) = E^0 x_i^p
\tag{17}
$$

The second objective is the maximization of the fundamental frequency, used for dynamic optimization of structures. The element sensitivity for natural frequency maximization is calculated by the following:

$$
\alpha_e^{frq} = \frac{\partial \omega_n}{\partial x_i} = \frac{1}{2\omega_n} \Phi_{e_n} \left(\frac{1 - x_{min}}{1 - x_{min}^p} p x_i^{p-1} [\mathbf{K}]_e^0 - \omega_n^2 [\mathbf{M}]_e^0 \right) \Phi_{e_n}
\tag{18}
$$

where Φ_{e_n} is the element eigenvector that corresponds to the nth mode, ω_n is the nth natural frequency, where ω_n^2 is the eigenvalue that corresponds to the nth mode, and $[\mathbf{M}]$ is the global mass matrix of the structure. In order to avoid localized fictitious modes in the soft material,

Huang et al. (2010) proposed a modified SIMP material model, described by:

$$
\begin{aligned}
\rho(x_i) &= x_i \rho^0 \\
E(x_i) &= \left[\frac{x_{min} - x_{min}^p}{1 - x_{min}^p} (1 - x_i^p) + x_i^p \right] E^0
\end{aligned}
\tag{19}
$$

where ρ is the density of the material. When maximizing a given frequency, adjacent eigenmodes often converge towards each other, becoming multiple by having the same or very similar frequencies. The BESO method handles this numerical difficulty by taking the average of the sensitivities of the relative modes (Zuo et al. 2010). For a complete derivation of the frequency sensitivity number (18) and the material model (19) the reader is advised to seek out (Huang et al. 2010). Further discussion on the numerical difficulties and alternative material models for dynamic optimization can be found in Pedersen (2000) and Du and Olhoff (2007).

The final objective considered in this work is vorticity maximization. The last analysis in Section 4 is a multi-physics and multiobjective topology optimization problem. One of the objectives is to increase the mixing of two fluid species for a given Reynolds number. This objective is imperative to the operation of micro fluidic mixers since their function is to efficiently mix two fluid species. Thus, as the flows have low Reynolds numbers, typically lower than 1000, vorticity is an efficient measure of the degree of mixing. Recently, the authors of this work developed a soft-kill BESO method for the vorticity maximization of fluids using the Lattice Boltzmann Method (LBM) (Munk et al. 2016b). The sensitivity number was derived using the circulation method for vorticity (Abrahamson and Lonnes 1995) and the shape derivative given in Kasumba and Kunisch (2012). Therefore, the sensitivity number for vorticity maximization is given by:

$$
\alpha_e^{vrt} = \max(\vec{\omega}) - \Delta\gamma_e^T x_i^{p-1} \Delta\gamma_e
\tag{20}
$$

where $\vec{\omega}$ is the vorticity of the flow and $\Delta\gamma_e$ is the change of the element velocity vector defined as:

$$
\gamma_e = \left\{ \Delta\gamma_x, \Delta\gamma_y, \Delta\gamma_z, \Delta W_x, \Delta W_y, \Delta W_z \right\}^T
\tag{21}
$$

where γ_x, γ_y and γ_z are the spatial components and W is the circulation. For the element sensitivity number given in (20), the presence of fluid and solid are defined by a design variable of $x_i = 1$ and $x_i = x_{min}$, respectively.

3.3 Handling of multiple constraints by the method of Lagrange multipliers

To solve a multi-constrained optimization problem, Zuo et al. (2012) proposed transforming the problem formulation (15) into an equivalent formulation using a Lagrange relaxation. This is achieved by first defining slack variables S_k^2 to

transform inequality constraints into equality constraints, as follows:

$$g(\mathbf{x})_k + S_k^2 = 0 \tag{22}$$

where $g(\mathbf{x})_k$ represents the kth inequality constraint. Then, by using Lagrange multipliers λ_k, the relaxed objective function can be found as the following Lagrangian:

$$L(\mathbf{x}, \lambda, S^2) = \mu_1(\mathbf{x}) + \sum_{k=1}^{m} \lambda_k \left[g(\mathbf{x})_k + S_k^2 \right] \tag{23}$$

where m is the number of inequality constraints.

Hence, the sensitivity of the Lagrangian due to a change in design variable x_i is found by:

$$\alpha_e = \frac{\partial L}{\partial x_i} = \frac{\partial \mu_1}{\partial x_i} + \sum_{k=1}^{m} \lambda_k \frac{\partial g_k}{\partial x_i} \tag{24}$$

Thus, comparing the sensitivity function (24) with the single-objective sensitivity functions (Section 3.2), it is seen that the Lagrange multipliers are utilized to compromise the original objective function and constraints (Zuo et al. 2012).

The equivalent formulation requires not only the design variables $\mathbf{x}$ to be found, but also the Lagrange multipliers λ and slack variables S^2, in order for a solution to the optimization problem to be determined. The sensitivity of the Lagrangian to the additional variables is expressed as:

$$\frac{\partial L}{\partial \lambda_k} = g(\mathbf{x})_k + S_k^2 \tag{25}$$

and

$$\frac{\partial L}{\partial S_k} = 2\lambda_k S_k \tag{26}$$

The Kuhn-Tucker necessary conditions for optimality state that the additional sensitivities ((25) and (26)) need to equal zero. Therefore, in this way, an update trend for the Lagrange multipliers can be determined based on the Kuhn-Tucker conditions.

For the inequality constraints $g(\mathbf{x})$, if the constraint value, $g(\mathbf{x})_k$ is positive, then the additional sensitivity (25) must also be positive. Therefore, λ_k needs to be decreased for minimization of the Lagrangian. However, for (26) to equal zero, $\lambda_k = 0$ if $S_k \neq 0$, and, according to (22), if $S_k \neq 0$ then $g(\mathbf{x})_k \leq 0$. Therefore, if an inequality constraint is satisfied, the corresponding Lagrange multiplier is zero, and the constraint is not considered in the minimization of the Lagrangian.

The additional variables are continuous in an infinite domain. Thus, it is computationally infeasible to search such a domain for a solution with a direct method. Zuo et al. (2012) therefore suggest defining λ_k through a scaling function of replacement factors φ_k that range in a reduced domain $[0, 1)$, given by:

$$\lambda_k = \frac{\varphi_k}{1 - |\varphi_k|} \quad \varphi \in [0, 1) \tag{27}$$

However, in this case, the reduced domain must be extended to $(-1, 1)$, since the problem is multiobjective and thus it must be possible to increase, as well as decrease, the objectives from their initial values. Hence, the Lagrange multipliers λ_k are represented in the whole range by the replacement factors φ_k, since $\varphi_k = 0 \implies \lambda_k = 0$ and $\lim_{\varphi_k \to 1} \lambda_k = \infty$. Thus, the Lagrange multipliers can be increased or decreased by increasing or decreasing the corresponding replacement factors. In this way, the Lagrange multipliers which satisfy the Kuhn-Tucker conditions are determined by searching for replacement factors within $(-1, 1)$ using an increment programming algorithm.

Using this formulation the optimization problem (15) for the stiffness and dynamic objectives becomes:

$$
\begin{array}{ll}
\min(\mathbf{x}) & \frac{1}{2} \mathbf{u}^T [\mathbf{K}] \mathbf{u} \\
\text{subject to:} & [\mathbf{K}] \mathbf{u} = \mathbf{f} \\
& \sum_{i=1}^{N_e} \leq V \\
& C(\mathbf{x}) - C_l \leq 0 \\
& \omega_{l_n} - \omega_n(\mathbf{x}) \leq 0 \\
& C_r - C(\mathbf{x}) \leq 0 \\
& \omega_n(\mathbf{x}) - \omega_{r_n} \leq 0 \\
& \mathbf{x} = [x_{min}, 1]
\end{array}
\tag{28}
$$

where $C(\mathbf{x})$ is the compliance of the structure, ω_n is the nth natural frequency of the structure, C_l and ω_{l_n} are the constraints imposed by S_l, and C_r and ω_{r_n} are the constraints imposed by S_r. The constraints given by (13) and (14) are expanded in (28). Thus, using the method of Lagrange multipliers from this section to construct the Lagrangian for the inequality constraints of the compliance and natural frequency, the sensitivity number is given by:

$$\alpha_e = \alpha_e^{cmp} - \lambda_1 \alpha_e^{cmp} + \lambda_2 \alpha_e^{frq} \tag{29}$$

The overall element sensitivity number is therefore a combination of the element sensitivities of the mean compliance, α_e^{cmp}, and n^{th} natural frequency, α_e^{frq}. The amount each objective is considered is determined by the two normal constraints in the updated SNC method (Section 2), since the Lagrange multipliers are updated depending on whether or not the constraints are satisfied.

The Lagrange multipliers start from zero and are gradually updated according to the corresponding objectives. For example, the replacement factor φ_k for Lagrange multiplier λ_k is increased if the corresponding constraint is not satisfied. The increment defined for the replacement factor is set to a small value (1% in this study), if the constraint is not close to being satisfied. As the constraint approaches zero, i.e. becoming satisfied, the increment value gets smaller according to the difference between the constraint value and the imposed value. This is expressed mathematically, for the problem given in (28), by Algorithm 1.

Algorithm 1 Lagrange multipliers updating scheme

if $N_1(\mu(\boldsymbol{x}) - S_l)^T \leq 0$ *and* $-N_1(\mu(\boldsymbol{x}) - S_r)^T \leq 0$ **then**
    All constraints are satisfied.;
    else
        if $N_1(\mu(\boldsymbol{x}) - S_l)^T > 0$ **then**
            if $\frac{C(\boldsymbol{x}) - C_l}{C_l} > \varepsilon_1$ **then**
                $\varphi_1 = \varphi_1 + \varepsilon_1$
                else
                    $\varphi_1 = \varphi_1 + \frac{C(\mathbf{x}) - C_l}{C_l}$
                end
            end
            if $\frac{\omega_{l_n} - \omega_n(\boldsymbol{x})}{\omega_{l_n}} > \varepsilon_2$ **then**
                $\varphi_2 = \varphi_2 + \varepsilon_2$
                else
                    $\varphi_2 = \varphi_2 + \frac{\omega_{l_n} - \omega_n(\mathbf{x})}{\omega_{l_n}}$
                end
            end
        end
        if $-N_1(\mu(\boldsymbol{x}) - S_r)^T > 0$ **then**
            if $\frac{C_r - C(\boldsymbol{x})}{C_r} > \varepsilon_1$ **then**
                $\varphi_1 = \varphi_1 - \varepsilon_1$
                else
                    $\varphi_1 = \varphi_1 - \frac{C_r - C(\mathbf{x})}{C_r}$
                end
            end
            if $\frac{\omega_n(\boldsymbol{x}) - \omega_{r_n}}{\omega_{r_n}} > \varepsilon_2$ **then**
                $\varphi_2 = \varphi_2 - \varepsilon_2$
                else
                    $\varphi_2 = \varphi_2 - \frac{\omega_n(\mathbf{x}) - \omega_{r_n}}{\omega_{r_n}}$
                end
            end
        end
    end
end

In Algorithm 1, ϵ_1 and ϵ_2 are small numbers, which are set to $\epsilon_1 = \epsilon_2 = 0.01$ for this study to restrict the amount by which the Lagrange multipliers can change between optimisation iterations.

3.4 Mesh independence and convergence

In order to guarantee that a solution to the topology optimization problem (28) exists, some restriction on the resulting design must be introduced (Sigmund and Petersson 1998). A filter scheme is used to smooth the element sensitivities (29) across the entire domain. This alleviates the problems of mesh-dependency and checkerboarding, which are a result of the sensitivity numbers becoming discontinuous across the element boundaries. The filter scheme requires the nodal sensitivity numbers, which are defined as
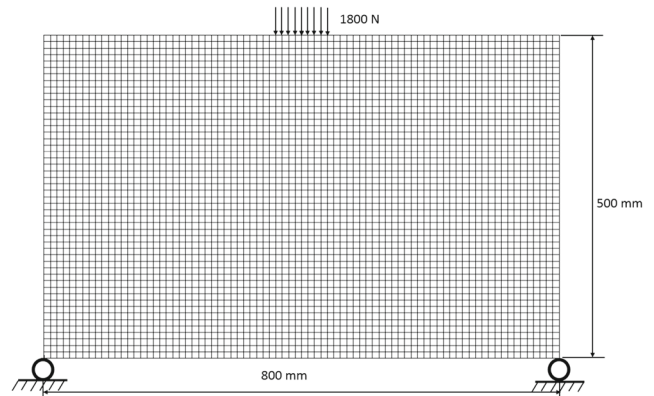

Fig. 4 Initial rectangular plate design domain (Proos et al. 2001a)

the average of the sensitivities of the elements connected to the node. It follows that:

$$\alpha_n = \sum_{i=1}^{M} w_i \alpha_{e_i} \tag{30}$$

where M is the number of elements connected to the node. The weighting factor of the ith element, w_i, is defined with respect to its distance from the jth node, r_{ij}, as:

$$w_i = \frac{1}{M-1}\left(1 - \frac{r_{ij}}{\sum_{i=1}^{M} r_{ij}}\right) \tag{31}$$

where r_{ij} is the distance from the center of the ith element to the jth node.

The nodal sensitivity numbers (30) are then converted to smooth element sensitivities using a mesh-independency filter. A sub-domain, Ω, is defined by the filter radius, r_{min}, which identifies the nodes that the element sensitivities are spread over. The value of r_{min} must be large enough that Ω covers at least one element; for the purpose of this study, $r_{min} = 3$. Therefore, nodes located inside Ω contribute to the smoothing of the element sensitivity, by:

$$\alpha_e = \frac{\sum_{j=1}^{N} w(r_{ij}) \alpha_{n_j}}{\sum_{j=1}^{N} w(r_{ij})} \tag{32}$$

where N is the total number of nodes in the sub-domain, Ω, and $w(r_{ij})$ is a linear weighting factor defined as:

$$w(r_{ij}) = r_{min} - r_{ij} \quad j = 1, 2, \ldots, N \tag{33}$$

The filter scheme smooths the elemental sensitivity numbers over the entire design domain, including void regions. Therefore, it effectively addresses the mesh-dependency and checkerboard problems. However, the objective function and corresponding topology may not be convergent. In order to overcome this problem, Huang and Xie (2007) showed
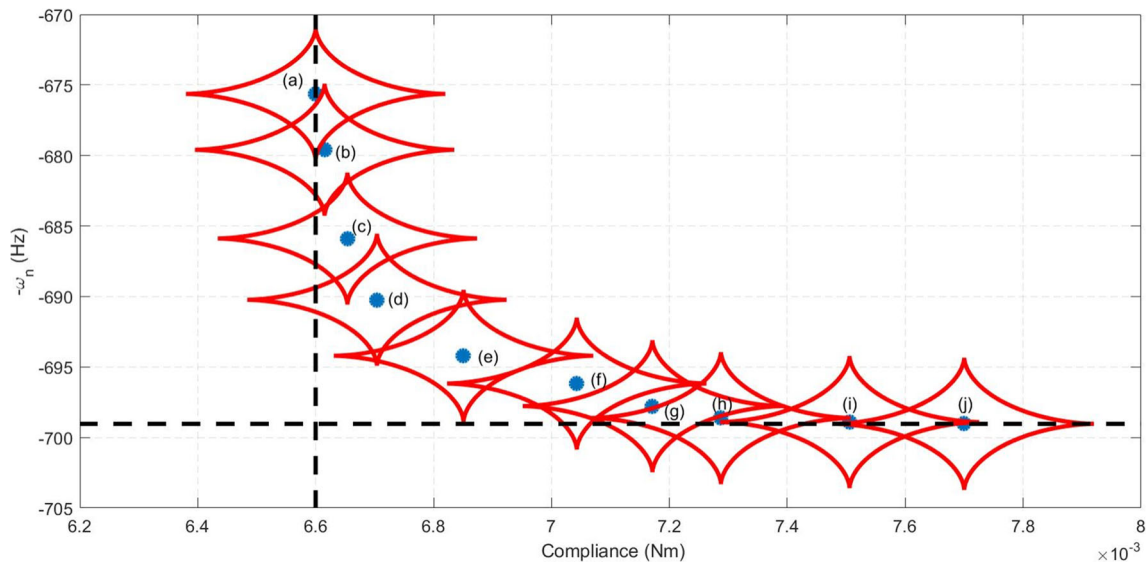
Fig. 5 Pareto frontier of the first mode natural frequency and mean compliance for the roller-supported rectangular plate found using the SNC-BESO method

that the sensitivity numbers (29) should be averaged over their history, thus:

$$\alpha_e = \frac{\alpha_e^{itr} + \alpha_e^{itr-1}}{2} \tag{34}$$

where itr is the current iteration number. Therefore, the updated sensitivity number includes the history of the sensitivity information from the previous iterations.

The BESO method defines a target volume for each iteration, given by:

$$V_{itr+1} = V_{itr}(1 \pm ER) \tag{35}$$

where ER, known as the evolutionary ratio, is a percentage of the current structural volume and increases or decreases V_{itr+1} towards the desired volume constraint V. Hence, this in turn sets the threshold, α_{th}, of the sensitivity numbers. Solid elements are switched to void when:

$$\alpha_e \leq \alpha_{th} \tag{36}$$

and void elements are switched to solid when:

$$\alpha_e > \alpha_{th} \tag{37}$$

The amount by which the volume of the structure can increase between iterations, AR, is restricted by a maximum
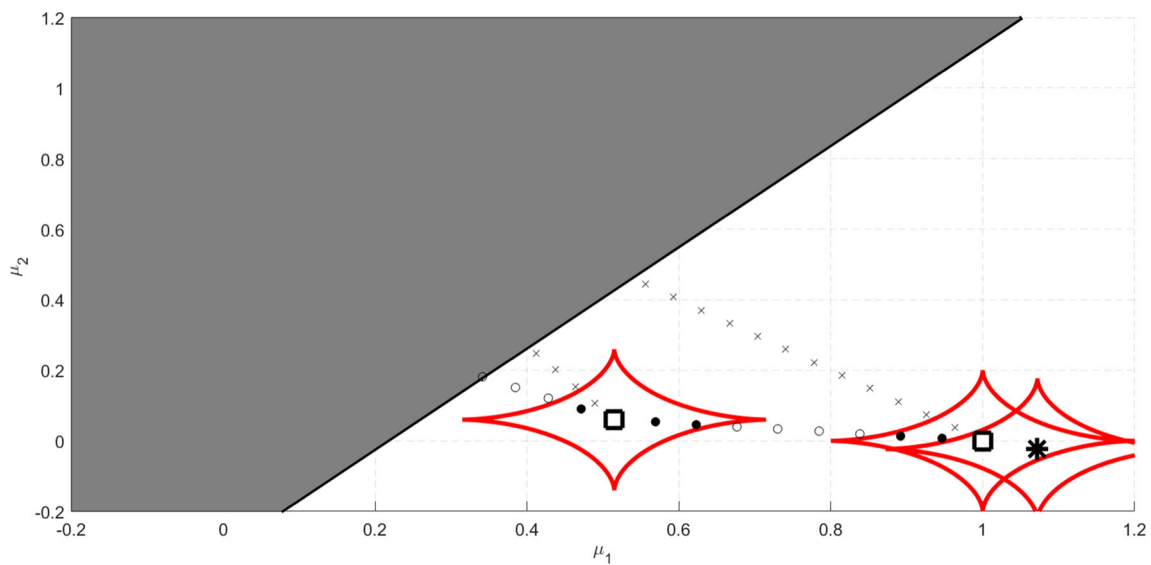
Fig. 6 Redundant point obtained by solving the rectangular plate with roller-supports problem using the original SNC formulation. Redundant points are shown as *

addition ratio, AR_{max}. Once $AR > AR_{max}$, only some of the elements (those with the highest sensitivity numbers) are added, such that $AR = AR_{max}$. Then the elements with the lowest sensitivity numbers are removed, in order to satisfy the target volume V_{itr+1}. Void elements can have higher sensitivities than solid elements due to the soft material model, which adds a small fictitious stiffness to these elements. Due to the mesh-dependency filter, void elements near solid regions with high sensitivity numbers have their sensitivities increased by the elements inside their sub-domain Ω.

Once the volume constraint is satisfied, the iteration target volume remains constant at V. The topology evolves until a convergence criterion and the normal constraints are satisfied. The convergence criterion is defined as:

$$\Delta C = \frac{\sum_{h=0}^{4} C_{itr-h} - \sum_{h=0}^{4} C_{itr-5-h}}{\sum_{h=0}^{4} C_{itr-h}} \leq \delta \tag{38}$$

where δ is a predefined error tolerance. (38) evaluates the change in the objective for the last 10 solutions. The solution is deemed to be converged if the change in the objective is minimal and all constraints are satisfied. More details on evolutionary topology and shape optimization algorithms can be found in the latest textbooks (Huang and Xie 2010) and review articles (Munk et al. 2015) on the subject.

4 Results and discussion

This section presents the results for applications of the new SNC-BESO algorithm developed in this study. First, a 2D plane stress problem taken from the work of Proos et al. (2001a) is analysed. This allows the performance of the algorithm to be compared with a previous method used for multiobjective topology optimization. This validates the new method and shows its benefits compared to the current literature. Then, an industrial problem, which is both multiobjective and multi-physics, the design of a micro fluidic mixer, is considered. This second example shows the ability of such methods to be applied to real-world problems.

4.1 Rectangular plate with roller supports (Proos et al. 2001a)

Proos et al. (2001a) present a problem where a structure must be designed to support nine point loads, each having a magnitude of 200 N, distributed at 0.01 m intervals. The structure has roller supports at the bottom two corners (Fig. 4). The design domain has a width of 0.8 m, a height of 0.5 m and a thickness of 0.01 m. A discretization of 80×50 four-node square elements is used to model the design domain (Fig. 4). The material properties used for the elements are: a Young's modulus of $E = 200$ GPa, a Poisson's ratio of
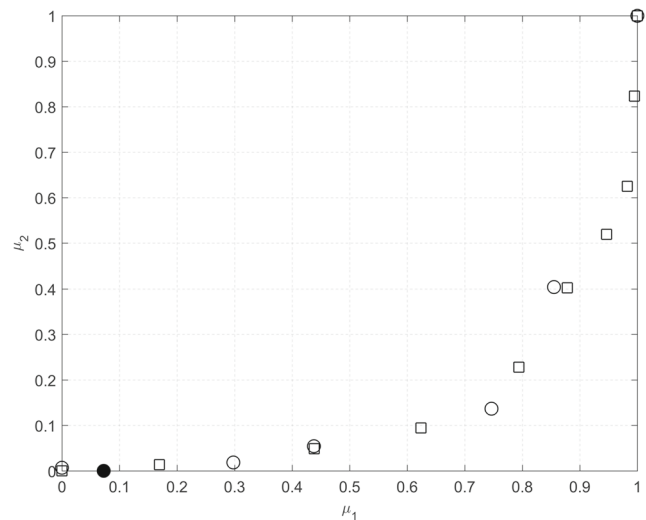

Fig. 7 Normalised Pareto frontier, ∘ samples taken from Proos et al. (2001a) and □ samples from this study

$\nu = 0.3$, and a density of $\rho = 7000$ kg m^{-3}. Throughout the analysis 2D plane-stress conditions are assumed.

The objectives of this optimization problem are to minimize the mean compliance and to maximize the first mode natural frequency. A volume constraint of $V = 0.7$ is also applied to the problem. The MOTO problem is then solved using the SNC-BESO method presented in this paper. The amount of change in either objective that would constitute a significant difference between two points, if all other objectives remain practically unchanged, is set to 5%. The amount of curvature of the PIT region is defined as $p = 0.6$. The corresponding Pareto curve of the first mode natural frequencies and the minimum mean compliance terms is presented in Fig. 5. Points marked (a)–(j) correspond to the topologies shown in Fig. 8. The dashed lines correspond to the limits of both objectives, found by solving a single-objective topology optimization problem for each objective.

The first observation is that the points given in the Pareto frontier of Fig. 5 make up a smart Pareto set: the points are evenly spaced along the Pareto front, such that no point lies within the PIT of any other. Furthermore, all PIT regions intersect their neighbouring point's PIT region in some location, demonstrating that the entire design space has been searched. Therefore, no new design can be found that would be of interest to the designer. This is a particularly important improvement over the previous methods used in multiobjective topology optimisation, as the designer has the minimum amount of information needed to give all the possibilities for the problem under consideration. Thus, the SNC-BESO method used is able to access the entire design domain, produce evenly distributed Pareto solutions and efficiently obtain a smart Pareto set, showcasing its ability to solve

Fig. 8 Optimal designs of the
roller-supported rectangular
plate found using the
SNC-BESO method
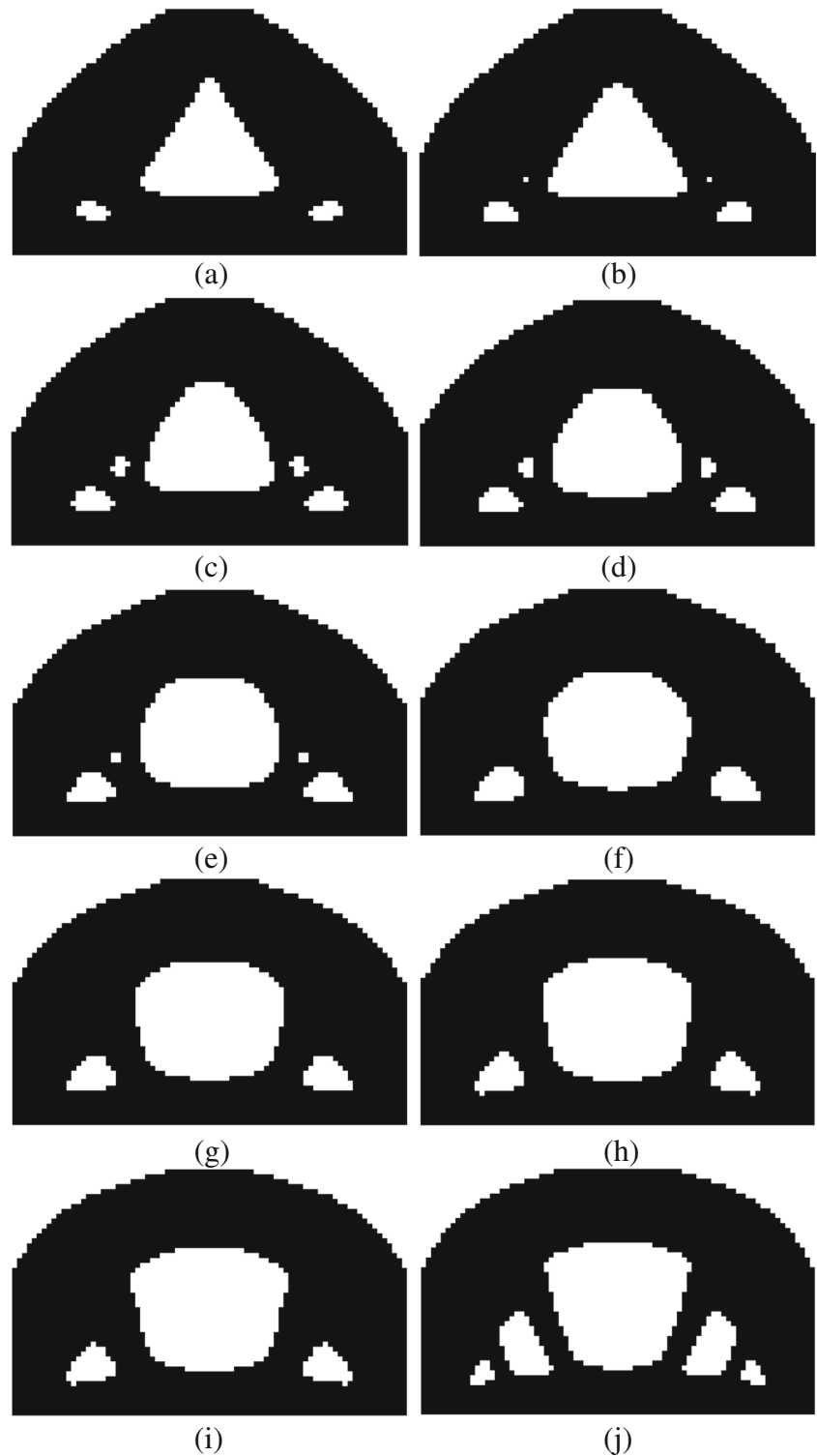
multiobjective optimization problems in an efficient and effective manner.

It is worth noting that all approximation points chosen produced a smart Pareto solution. Hence, only 10 optimization runs, the minimum for the given problem, were required to obtain the full smart Pareto set (Fig. 5). The same does not occur when using the original SNC algorithm (Hancock and Mattson 2013), as redundant points are produced (Fig. 6). Figure 6 shows the design domain for the sixth Pareto solution using the original SNC method with

(a) Fluid domain (Tsotskas et al. (2015))

(b) Layout of micro-reactor model (Tsotskas et al. (2015))

(c) Multi-holed baffle plate
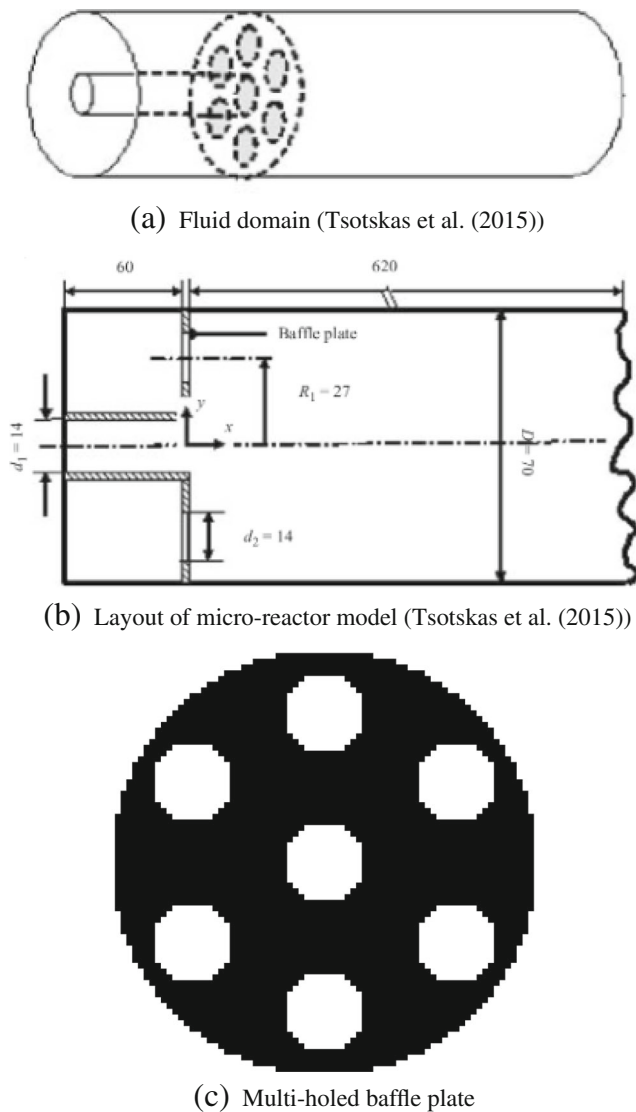
**Fig. 9** Initial design domain of micro fluidic mixer (Munk et al. 2016b)

the BESO algorithm. The grey zone shows the restricted region enforced by the normal constraint, with the black line being orthogonal to the approximated Pareto curve. The ○ symbols are the current approximation points, which the algorithm can choose from to find a new Pareto solution, whereas the × symbols are the previous approximation points from earlier iterations. The ● symbols are current approximation points that lie inside the PIT of a previously found Pareto solution and hence are not considered by the algorithm. Pareto solutions that belong to the smart Pareto set are depicted by □ symbols in Fig. 6, whereas the Pareto solutions that break one or more of the criteria in Step 7 (Section 2.2) are depicted by ∗ symbols. The solution found using the original SNC method falls inside the PIT of another Pareto solution. Therefore, it is a redundant point

and has not added to the smart Pareto set. Hence, this point would be removed. Thus, the updated SNC-BESO method is more efficient compared with the original SNC method on this problem. This demonstrates the benefit of the update made to the SNC algorithm: namely, that the penalties on the sensitivity function (29) are no longer monotonic, i.e. they can be increased and decreased, whereas without (14) they can only be increased. This can lead to the constraints dominating the solution and result in redundant points, as shown by Fig. 6.

Comparatively, the work of Proos et al. (2001a) shows the deficiencies of the weighted sums method: evenly distributed weights are prescribed, but an even distribution of points is not obtained (Fig. 7). Furthermore, the points are concentrated around the *knee* region of the Pareto curve, where larger areas of trade-off are observed. Proos et al. (2001a) found that their analysis, for this particular problem, did not lead to designs that showed any improvement in one criterion leading to a clear trade-off with the others, i.e. increasing the weights of one objective did not necessarily lead to an improvement in that objective with a corresponding reduction in the others. The authors found that the solution produced with a stiffness criterion weighting of 90% (depicted by a ● symbol in Fig. 7) had a lower natural frequency than the solution produced with a stiffness criterion weighting of 100%. However, the mean compliance was lower for the solution with a stiffness criterion weighting of 100%. Thus, it is optimal in terms of being the stiffest design; the authors had found a locally Pareto-optimal solution (Fig. 7), i.e. a dominated point was produced. This is an example of how a satisfactory, *a priori* selection of weights does not guarantee an acceptable final solution will be obtained (Marler and Arora 2010). These problems are not evident for the SNC-BESO method of this work.

The final topologies (Fig. 8) produced using the SNC-BESO method are not affected by numerical instabilities, such as mesh-dependency and checkerboarding. Clear holes have been created in each solution, with a uniform transition between the two anchor points. Conversely, the topologies produced by Proos et al. (2001a) contain some checkerboarding, with several small holes formed and elements connected at only two corners. This shows the benefits of implementing an updated BESO method compared to the ESO method. The mesh-independency filter employed in this work spreads the sensitivities across the entire design domain such that these instabilities do not occur. Furthermore, the topologies produced in this work (Fig. 8) are convergent, whereas the ESO method does not have a rigorous convergence criterion. The SNC-BESO method is thus better able to find smart Pareto sets of MOTO problems.

Fig. 10 Pareto frontier of negative vorticity and mean compliance for the micro fluidic mixer baffle plate found using the SNC-BESO method
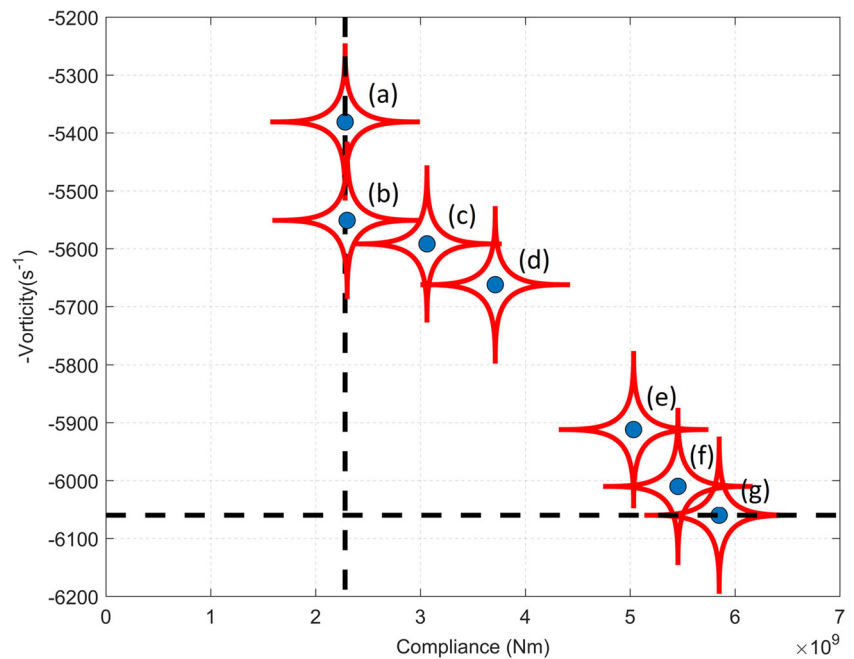
4.2 Micro fluidic mixer

The authors of the present study analysed a baffled micro reactor to develop a BESO algorithm for multi-physics optimization with design-dependent pressure loads (Munk et al. 2016b). The model consists of a tubular vessel fitted with a fuel inlet tube, located co-axially in the main vessel, and a multi-holed baffle plate through which the oxidizer is introduced. The fluid domain and layout of the micro-reactor model are shown in Fig. 9.

The dimensions of the fluid domain (Fig. 9b) are given in LBM nodes, where the dimensions of the lattice are $680 \times 73 \times 73$ lattice units, with additional nodes used for the wall, in the x, y and z directions, respectively. The baffle is located 60 lattice units downstream of the flow inlet (Fig. 9b). The imposed inlet conditions are the velocities of the flow in the fuel inlet tube and the annulus area. At the outlet, a convective boundary condition is applied, based on the velocity. The no-slip boundary condition is implemented at the walls, by modelling them as full-way bounce back in the LBM. The mass flow rate between the inner tube and annulus is set to 5% to mimic the experiments performed by Moghtaderi et al. (2006).

In this section, the SNC-BESO algorithm is applied to the multi-holed baffle plate simultaneously to maximize its stiffness and the vorticity of the two flows for a given volume fraction, $V = 0.58$, and Reynolds number, $Re = 100$, chosen to match the previous single-objective optimisation runs (Munk et al. 2016a; Munk et al. 2016b). The baffle is modelled by four-node quadrilateral plate elements with all six degrees-of-freedom active. Hence, membrane, bending and transverse shear stresses are present. A clamped boundary condition, i.e. all six degrees of freedom are restrained, is applied along the boundary of the baffle. The boundary of the central hole is designated as non-designable material for the topology optimization, since this is determined by the fuel line and inlet conditions, which have been constrained in the fluid domain (Fig. 9a) to be identical to previous numerical (Munk et al. 2016b) and experimental (Moghtaderi et al. 2006) studies. The amount of change in either objective that would constitute a significant difference between two points if all other objectives remain practically unchanged is set to 5%. The amount of curvature of the PIT region is defined as $p = 0.4$. The corresponding Pareto curve of the negative vorticity of the flow and the mean compliance of the structure is presented in Fig. 10. Points marked (a)–(g) correspond to the topologies shown in Fig. 11. The dashed lines correspond to the limits of both objectives, found by solving a single-objective topology optimization problem for each objective.

For this multiobjective and multi-physics problem, it is clear that there are two Pareto fronts, separated by a small gap, between solutions (d) and (e) (Fig. 10). This suggests that, for this problem, the Pareto front is discontinuous. This conclusion is further supported by the resulting topologies, which portray a rapid change between solutions (d) and (e) where they become progressively similar to their respective single-objective solutions. This is not surprising since the two objectives produce very different final topologies and are based on different physical models, structures and fluids.

The authors of this paper found in a previous study that certain topological features resulted from the physics of the two single-objective problems (Munk et al. 2016b). For the compliance minimization problem, the structural symmetry
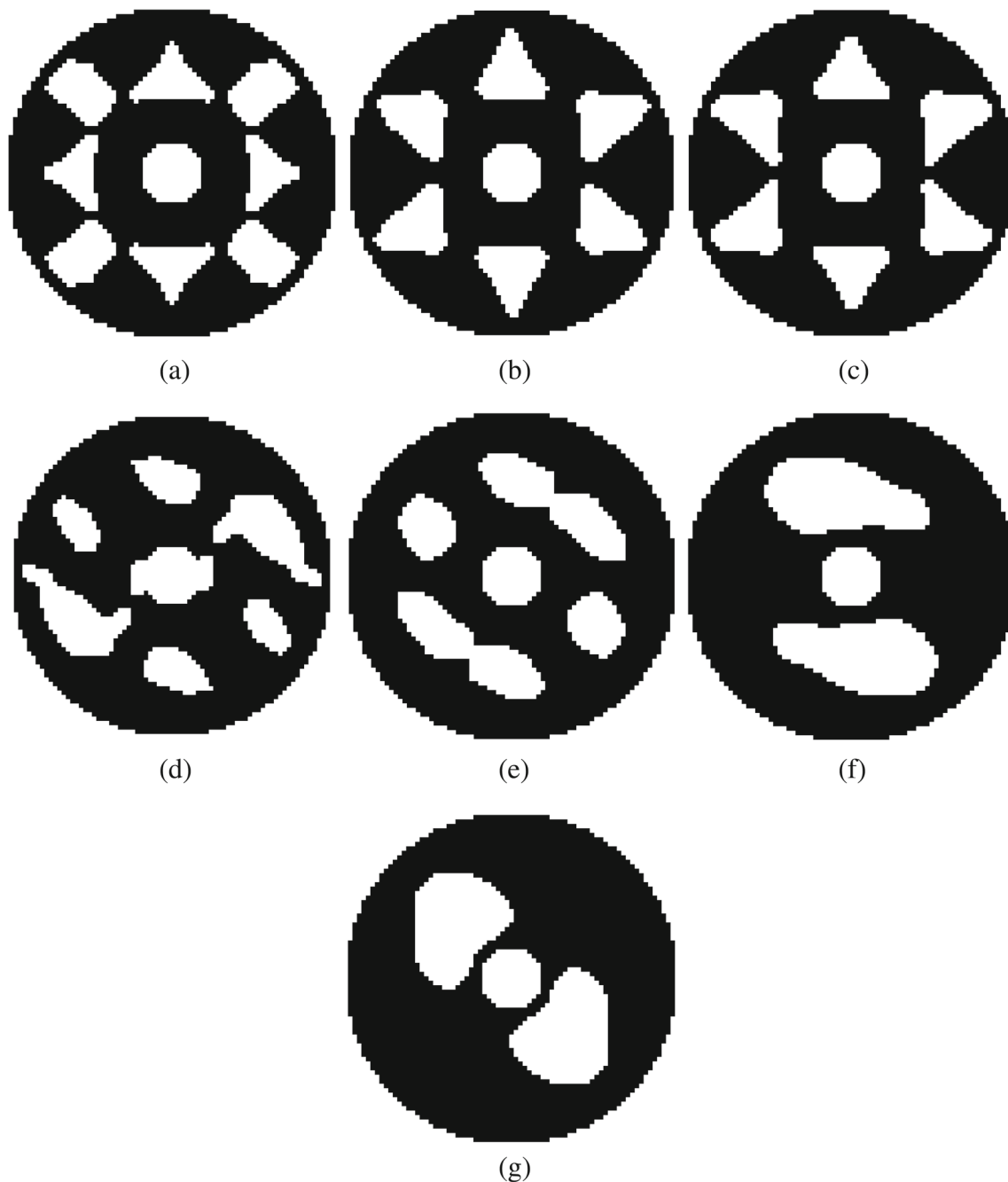
Fig. 11 Optimal micro fluidic mixer baffle plate designs found using the SNC-BESO method

about the x- and y-axes becomes identical (Fig. 11a). It was shown that this is physically reasonable, since there is no physical difference about the horizontal and vertical axes. Therefore, there is nothing physically present to introduce asymmetry into the topology, and it makes physical sense for the compliance SOO problem to exhibit 4-fold rotational symmetry, SO(4), about the center point of the baffle. Conversely, it was shown that vorticity is not a symmetrical phenomenon (Munk et al. 2016b). Therefore, for the vorticity SOO solution the topology is not SO(4), but

displays some symmetry about the ±45° diagonals. Hence, the topologies produced by the two SOO problems are considerably different. The change in the topology from being SO(4) to symmetric about the ±45° diagonals is clearly noticeable in the different geometries shown in Fig. 11.

The strain energy distributions for the different Pareto solutions are given in Fig. 12. Moving progressively from solution (a) to solution (g), clear strain energy concentrations begin to appear around solution (d). Solutions (a)–(c) show comparatively low strain energy distributions, with an
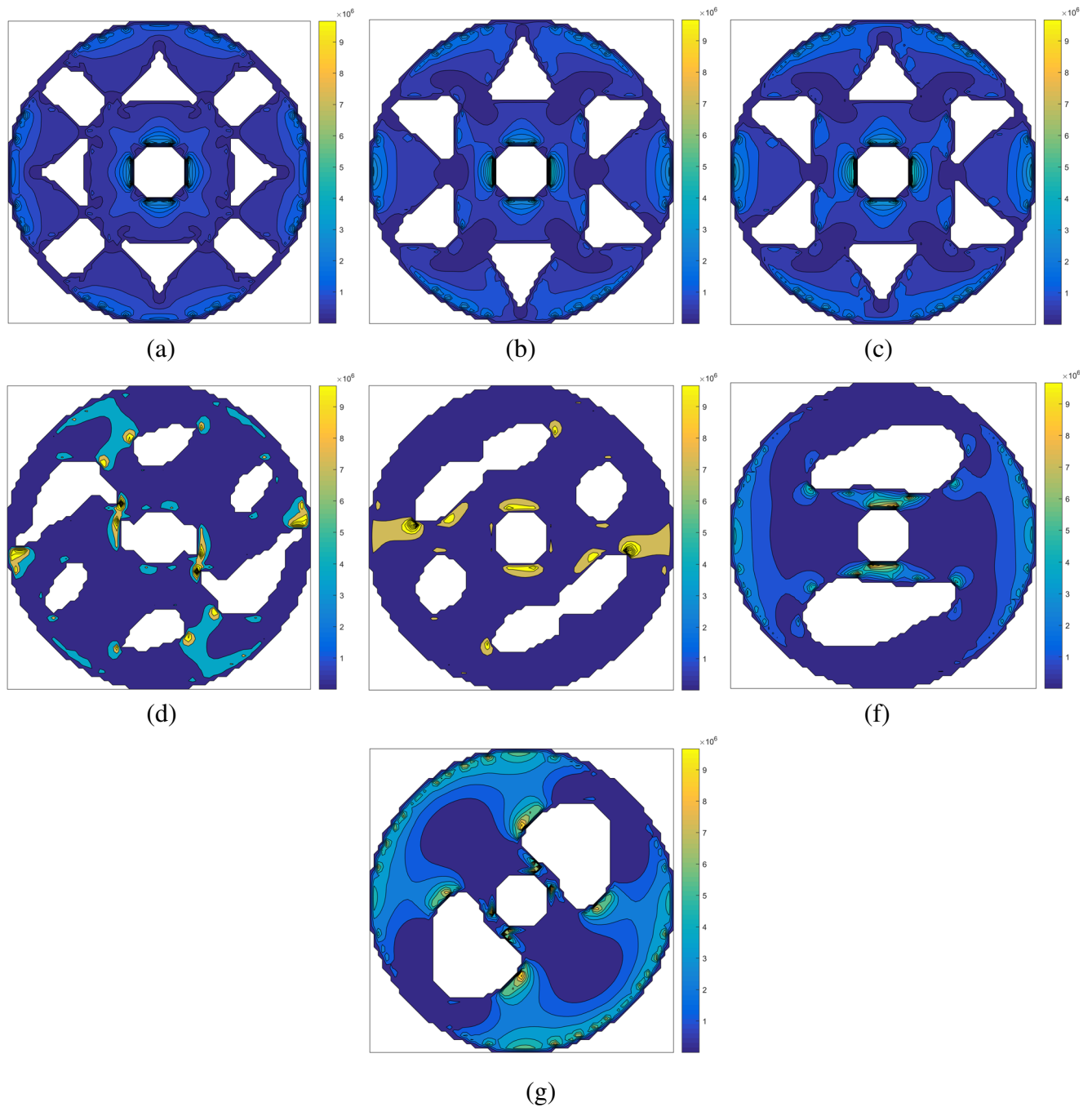
Fig. 12 Strain energy density for optimal micro fluidic mixer baffle plate designs

almost uniform distribution, especially considering that the centre hole is non-designable. Solution (g) has large regions of relatively high strain energy due to the large pressure difference present across the baffle.

The efficiency for each baffle design in mixing can be seen by the velocity and streamline plots (Fig. 13). Clearly, the latter solutions ((e)–(g)) promote mixing by increasing the velocity of the flow. The flow characteristics of the earlier designs ((a)–(c)) are difficult to distinguish for the chosen scale. This is because the low compliance designs try to minimise the pressure difference across the baffle, and hence the velocity increase is low compared with the high vorticity designs. Furthermore, by comparing the streamlines, it is apparent that the low compliance designs do

not produce significant mixing. The streamlines show small circulation zones immediately after the baffle. However, by looking at solutions (e)–(g), it is seen that mixing is significantly promoted by large areas of circulation downstream of the baffle.

For the high vorticity designs ((e)–(g)), mixing occurs mainly along the 45° diagonal that does not contain the holes, whereas, in the low compliance designs ((a)–(d)), mixing occurs around the holes. The area of mixing is larger in the high vorticity designs, as the spacing between the two/four holes is larger than that between the six holes of the low compliance designs. By comparing the flow paths (Fig. 13) of the Pareto solutions, it is seen that solutions (e)–(g) have considerably higher recirculation zones, indicating a bigger mixing zone. Conversely, the six smaller holes reduce the compliance of the structure, spreading the strain energy more evenly over the baffle. However, the larger holes result in increased strain energy concentrations, resulting in a structure with a high compliance.

To further verify the Pareto solutions found for the multiobjective and multi-physics micro fluidic baffle plate optimization problem, a Tabu Search (TS) algorithm (Jaeggi et al. 2008) was applied to the problem. The TS algorithm used a level-set parametrisation to maximize the vorticity and minimize the pressure difference of the flow. While minimizing the pressure of the flow is not identical to the minimum compliance objective, it is similar to maximizing the stiffness of the baffle, albeit, as a fluid objective. This is because it reduces the fluid loads on the baffle and hence also the compliance of the baffle. Again, the center hole is designated as non-designable material. The resulting Pareto front is displayed in Fig. 14.

The Pareto front found (Fig. 14) is again found to be discontinuous. There is, however, a small difference between the optimization algorithms used: the volume fraction of the baffle can change in the TS algorithm, but not in the SNC-BESO algorithm. Therefore, the results are not comparable below a vorticity of $5400s^{-1}$, as this region does not fall into the design space of the topology optimization problem of this paper. As was shown by Munk et al. (2016b), the topology optimization algorithm is able to find a more optimal solution to the vorticity maximization problem, thereby increasing the design space slightly at the maximum vorticity end.

The Pareto front found using the SNC-BESO algorithm of this work is able to identify the different options available to the designer for the multiobjective and multi-physics optimization problem. These results indicate that the SNC-BESO method is capable of producing smart Pareto sets to industrial problems, which, to the best of the authors' knowledge, has not been shown before in the BESO literature.
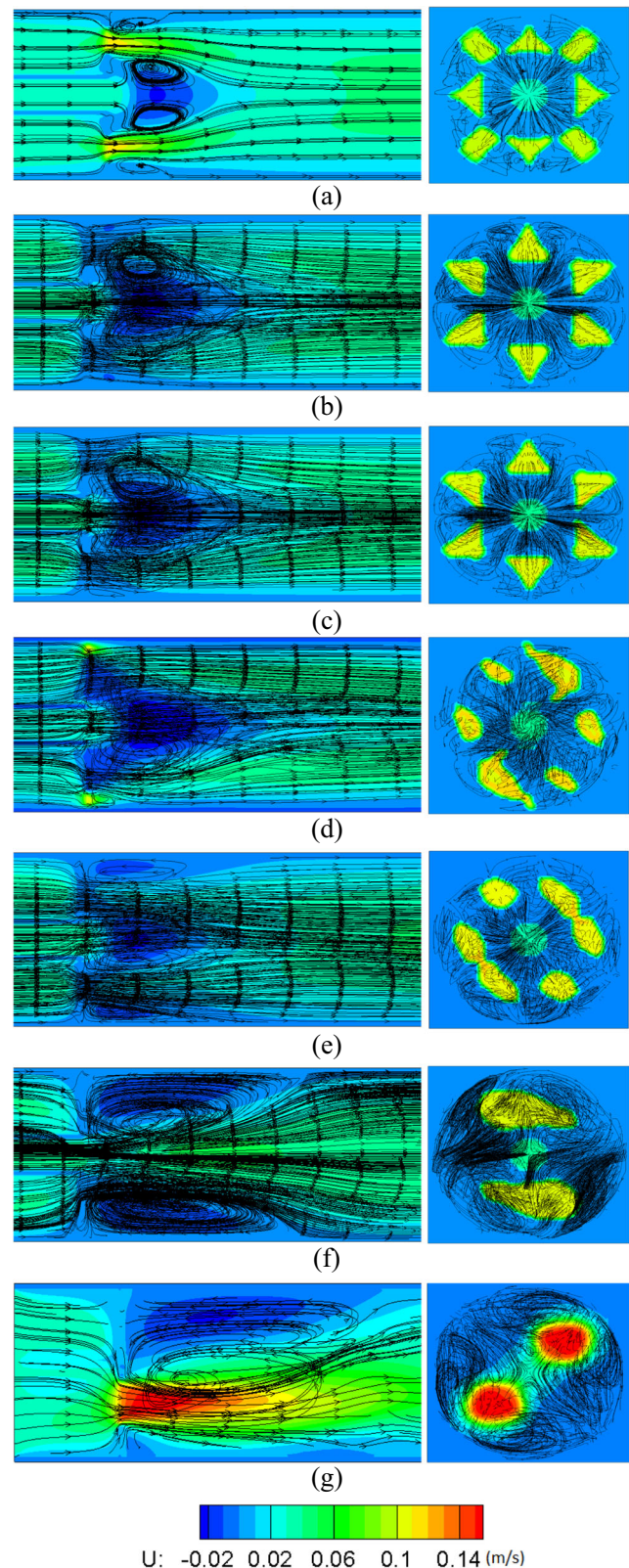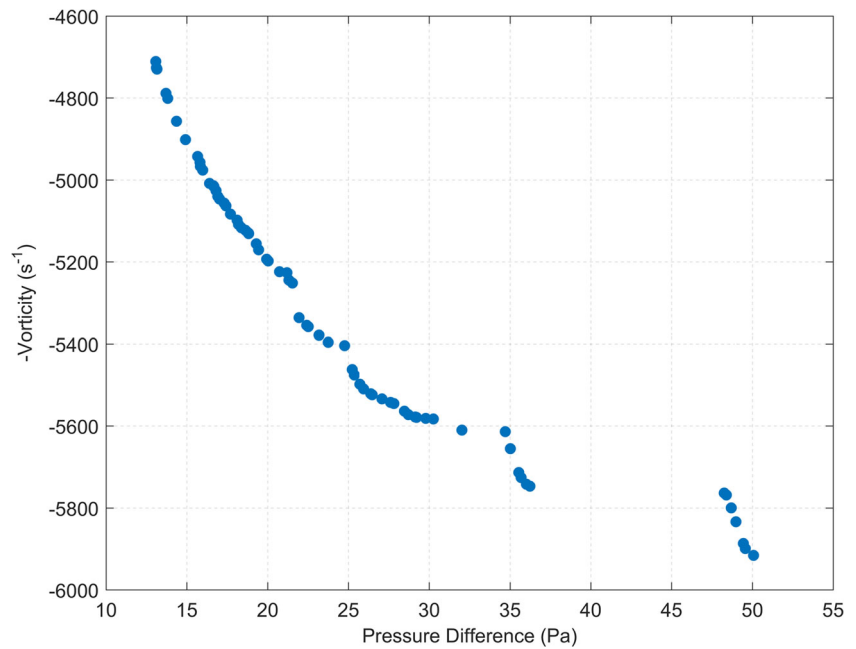

Fig. 13 Streamlines for the optimal micro fluidic mixer baffle plate designs

Fig. 14 Pareto frontier of
pressure difference and negative
vorticity for the micro fluidic
mixer baffle plate found using
the TS algorithm (data produced
by Dr Tiziano Ghisu)

5 Conclusions

A novel multiobjective topology optimization algorithm, termed SNC-BESO, which uses an updated smart constraints method combined with a bi-directional evolutionary optimization algorithm, has been presented. The literature survey showed that, thus far, topology optimization methods have mainly focused on single-objective problems. For BESO/ESO type algorithms adapted for multiobjective optimization, only the weighted sums and global criterion methods have been incorporated. It is known that such methods are unable to produce smart Pareto sets. Two MOTO problems were solved using the SNC-BESO method, the first taken from the limited MOTO literature and the second an industrial problem concerning the design of a micro fluidic mixer.

The first test-case was purely a structural design problem, with stiffness and dynamic design criteria. The problem had previously been solved in the literature using a weighted sums ESO method (Proos et al. 2001a). The Pareto front determined by the SNC-BESO method was found to constitute a smart Pareto set, while the same cannot be said for the Pareto front found using the weighted sums method (Proos et al. 2001a). It was found that the updates made to the SNC method avoided redundant points being produced, a problem that occurs when using the original SNC method. Thus, the updated SNC method is more efficient than the original SNC method when used in topology optimization, and the SNC-BESO method proposed in this work is able to solve MOTO problems in an efficient and effective manner.

The second problem tackled is multi-physics as well as multiobjective, having stiffness (structural) and vorticity (fluids) objectives. To the best of the authors' knowledge, such a problem has not been solved before in the BESO literature. Therefore, a comparison to a similar problem solved using a TS algorithm was given. It was shown that the Pareto front for this problem had a discontinuity present, where the physics of the problem drove the solutions closer towards the anchor points of the Pareto front. This is expected when the objectives counteract each other completely. The SNC-BESO method was able to identify this discontinuity and find a range of solutions, which displayed the different design options for the problem.

The work presented here adds to the literature on using high-fidelity methods, such as Lattice Boltzmann flow simulations, in topology optimization algorithms at the conceptual/preliminary design stages. Furthermore, a topology optimization algorithm that can handle multiple objectives, as well as disciplines, to better optimize real-world applications was demonstrated. This type of analysis is instrumental for the further application of topology optimization to industrial design problems, where the consideration of multiple objectives, as well as disciplines, is a very frequent requirement. Finally, it was mentioned that multiple constraint problems are unavoidable in real-world engineering design. Therefore, extending the SNC-BESO method to also handle multiple constraints is left for planned future work.

References

Abrahamson S, Lonnes S (1995) Uncertainty in calculating vorticity from 2D velocity fields using circulation and least-squares approach. Exp Fluids 20:10–20

Athan TW, Papalambros PY (1996) A note on weighted criteria methods for compromise solutions in multi-objective optimization. Eng Optim 27:155–176

Barber CB, Dobkin DP, Huhdanpaa HT (1996) The quickhull algorithm for convex hulls. ACM Trans Math Softw 22:469–483

Bendsoe MP (1989) Optimal shape design as a material distribution problem. Struct Optim 1:193–202

Bendsoe MP, Kikuchi N (1988) Generating optimal topologies in structural design using a homogenization method. Comput Method Appl Mech Eng 71:197–224

Bendsoe MP, Sigmund O (1999) Material interpolation schemes in topology optimization. Arch Appl Mech 69:635–654

Bendsoe MP, Sigmund O (2004) Topology optimization: theory methods and applications, 2nd edn. Springer, Berlin, Heidelberg, New York

Boyce NO, Mattson CA (2008) Reducing computational time of the normal constraints method by eliminating redundant optimization runs. In: 12th AIAA/ISSMO multidisciplinary analysis and optimization conference. AIAA

Chen W, Wiecek MM, Zhang J (1999) Quality utility - a compromise programming approach to robust design. J Mech Des 121:179–187

Chen W, Sahai A, Messac A, Sundararaj G (2000) Exploration of the effectiveness of physical programming in robust design. J Mech Des 122:155–163

Chu D, Xie YM, Hira A, Steven GP (1996) Evolutionary structural optimization for problems with stiffness constraints. Finite Elem Anal Des 21:239–251

Coello CAC, Lamont GB, Veldhuizen DAV (2007) Evolutionary algorithms for solving multi-objective problems, 2nd edn. Springer, New York

Das I (1999) An improved technique for choosing parameters for Pareto surface generation using normal-boundary intersection. University of Buffalo, Center for Advanced Design, Buffalo

Das I, Dennis JE (1997) A closer look at drawbacks of minimizing weighted sums of objectives for Pareto set generation in multicriteria optimization problems. Struct Optim 14:63–69

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

Deaton JD, Grandhi RV (2014) A survey of structural and multidisciplinary continuum topology optimization: post 2000. Struct Multidiscip Optim 49:1–38

Deb K (2009) Multi-objective optimization using evolutionary algorithms, 1st edn. Wiley, New York

Du J, Olhoff N (2007) Topological design of freely vibrating continuum structures for maximum values of simple and multiple eigenfrequencies and frequency gaps. Struct Multidiscip Optim 34:91–110

Haddock ND, Mattson CA, Knight DC (2008) Exploring direct generation of smart Pareto sets. In: 12th AIAA/ISSMO multidisciplinary analysis and optimization conference. AIAA

Hancock BJ, Mattson CA (2013) The smart normal constraints method for directly generating a smart Pareto set. Struct Multidiscip Optim 48:763–775

Huang HZ, Gu YK, Du X (2006) An interactive fuzzy multi-objective optimization method for engineering design. Eng Appl Artif Intel 19:451–460

Huang X, Xie YM (2007) Convergent and mesh-independent solutions for the bi-directional evolutionary structural optimization method. Finite Elem Anal Des 43:1039–1049

Huang X, Xie YM (2009) Bi-directional evolutionary topology optimization of continuum structures with one or multiple materials. Comput Mech 43:393–401

Huang X, Xie YM (2010) Evolutionary topology optimization of continuum structures, 1st edn. Wiley, Chichester

Huang X, Zuo ZH, Xie YM (2010) Evolutionary topological optimization of vibrating continuum structures for natural frequencies. Comput Struct 88:357–364

Ismail-Yahaya A, Messac A (2002) Effective generation of the Pareto frontier using the normal constraint method. In: 40th AIAA aerospace sciences meeting & exhibit. AIAA

Jaeggi DM, Parks GT, Kipouros T, Clarkson PJ (2008) The development of a multi-objective Tabu search algorithm for continuous optimisation problems. Eur J Oper Res 185:1192–1212

Kasumba H, Kunisch K (2012) Vortex control in channel flows using translation invariant cost functionals. Comput Optim Appl 52:691–717

Kim WY, Grandhi RV, Haney M (2006) Multiobjective evolutionary structural optimization using combined static/dynamic control parameters. AIAA J 44:794–802

Koski J (1985) Defectiveness of weighting method in multicriterion optimization of structures. Commun Appl Numer Methods 1:333–337

Kunakote T, Bureerat S (2011) Multi-objective topology optimization using evolutionary algorithms. Eng Optim 43:541–557

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

Marler RT, Arora JS (2010) The weighted sum method for multi-objective optimization: new insights. Struct Multidiscip Optim 41:853–862

Martinez JSM, Blasco X, Salceo JV (2009a) A new perspective on multiobjective optimization by enhanced normalized normal constraint method. Struct Multidiscip Optim 36:537–546

Martinez M, Sanchis J, Blasco X (2007) Global and well-distributed Pareto frontier by modifying normal normalized constraint methods for bicriterion problems. Struct Multidiscip Optim 34:197–209

Martinez M, Garcia-Nieto S, Sanchis J, Blasco X (2009b) Genetic algorithms optimization for normalized normal constraint method under Pareto construction. Adv Eng Softw 40:260–267

Martinez MP, Messac A, Rais-Rohani M (2001) Manufacturability-based optimization of aircraft structures using physical programming. AIAA J 39:517–525

Mattson CA, Mullur AA, Messac A (2004) Smart Pareto filter: Obtaining a minimal representation of multiobjective design space. Eng Optim 36:721–740

Messac A (2000) From dubious construction of objective functions to the application of physical programming. AIAA J 38:155–163

Messac A, Hattis P (1996) Physical programming design optimization for high speed civil transport (HSCT). J Aircraft 33:446–449

Messac A, Ismail-Yahaya A (2001) Required relationship between objective function and Pareto frontier orders: practical implications. AIAA J 11:2168–2174

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

Messac A, Mattson CA (2004) Normal constraints method with guarantee of even representation of complete Pareto frontier. AIAA J 42:2101–2111

Messac A, Sukam CP, Melachrinoudis E (2000a) Aggregate objective functions and Pareto frontiers: required relationships and practical implications. Optim Eng 1:171–188

Messac A, Sundararaj GJ, Tappeta RV, Renaud JE (2000b) Ability of objective functions to generate points on nonconvex Pareto frontiers. AIAA J 38:1084–1091

Messac A, Puemi-Sukam C, Melachrinoudis E (2001) Mathematical and pragmatic perspectives of physical programming. AIAA J 39:885–893

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

Messac A, Dessel SV, Mullur S, Maria AA (2004) Optimization of large scale rigidified inflatable structures for housing using physical programming. Struct Multidiscip Optim 26:139–151

Michell AGM (1904) The limits of economy of material in frame structures. Philos Mag 8:589–597

Moghtaderi B, Shames I, Djenidi L (2006) Microfluidic characteristics of a multi-holed baffle plate micro-reactor. Int J Heat Fluid Fl 127:1069–1077

Motta RS, Afonso SMB, Lyra PRM (2012) A modified NBI and NC method for the solution of n-multiobjective optimization problems. Struct Multidiscip Optim 46:239–259

Munk D, Vio G, Kipouros T, Parks G (2016a) Computational design for micro fluidic devices using a tightly coupled Lattice Boltzmann and level set-based optimization algorithm. In: Proceedings of the 11th ASMO UK/ISSMO conference on engineering design optimization. ASMO, UK

Munk DJ, Vio GA, Steven GP (2015) Topology and shape optimization methods using evolutionary algorithms: a review. Struct Multidiscip Optim 52(3):613–631

Munk DJ, Kipouros T, Vio GA, Parks GT, Steven GP (2016b) Topology optimization of micro fluidic mixers considering fluid-structure interactions with a coupled Lattice Boltzmann method. J Comput Phys Under review

Munk DJ, Vio GA, Steven GP (2017) A bi-directional evolutionary structural optimization algorithm with an added connectivity constraint. Finite Elem Anal Des 131:25–42

Pareto V (1964) Cour deconomie politique (the first edition in 1896) edn. Librarie Droz-Geneve, Geneva

Pedersen NL (2000) Maximization of eigenvalues using topology optimization. Struct Multidiscip Optim 20:2–11

Prager W, Rozvany GIN (1977) Optimization of the structural geometry. In: Bednarek A, Cesari L (eds) Dynamical systems. Academic Press, pp 265–293

Proos KA, Steven GP, Querin OM, Xie YM (2001a) Multicriterion evolutionary structural optimization using the weighting and the global criterion methods. AIAA J 39(10):2006–2012

Proos KA, Steven GP, Querin OM, Xie YM (2001b) Stiffness and inertia multicriteria evolutionary structural optimization. Eng Comput 18:1031–1054

Ray T, Tai K, Seow KC (2001) An evolutionary algorithm for multiobjective optimization. Eng Optim 33:399–424

Rozvany GIN (2009) A critical review of established methods of structural topology optimization. Struct Multidiscip Optim 37:217–237

Rozvany GIN, Lewinski T (eds) (2013) Topology optimization in structural and continuum mechanics, 1st edn. Springer, Dordrecht

Rozvany GIN, Zhou M, Birker T (1992) Generalized shape optimization without homogenization. Struct Optim 4:250–254

Ruzika S, Wiecek MM (2005) Approximation methods in multiobjective programming. J Optimiz Theory App 126:473–501

Rynne B (2007) Linear functional analysis, 1st edn. Springer, New York

Sigmund O (2001) Design of multiphysics actuators using topology optimization – Part I: One-material structures. Comput Method Appl Mech Eng 190:6577–6604

Sigmund O (2011) On the usefulness of non-gradient approaches in topology optimization. Struct Multidiscip Optim 43:589–596

Sigmund O, Maute K (2013) Topology optimization approaches: a comparative review. Struct Multidiscip Optim 48:1031–1055

Sigmund O, Petersson J (1998) Numerical instabilities in topology optimization: a survey on procedures dealing with checkerboards, mesh-dependencies and local minima. Struct Optim 16:68–75

Stadler W (1995) Caveats and boons of multicriteria optimization. Comput-Aided Civ Infrastruct Eng 10:291–299

Steven GP, Li Q, Xie YM (2000) Evolutionary topology and shape design for generating physical field problems. Comput Mech 26(2):129–139

Tanaka M, Watanabe H, Furukawa Y, Tanino T (1995) GA-based decision support system for multicriteria optimization. In: 1995 IEEE international conference on systems, man, and cybernetics, vol 2. IEEE, pp 1556–1561

Tsotskas C, Kipouros T, Savill M (2015) Fast multi-objective optimisation of a micro-fluidic device by using graphics accelerators. Proc Comput Sci 51:2237–2246

Xie YM, Steven GP (1993) A simple evolutionary procedure for structural optimization. Comput Struct 49:885–896

Yang XY, Xie YM, Steven GP, Querin OM (1999) Bidirectional evolutionary method for stiffness optimization. AIAA J 37:1483–1488

Zadeh LA (1963) Optimality and non-scalar-valued performance criteria. IEEE Trans Automat Contr 8:59–60

Zhou A, Qu BO, Li H, Zhoa SZ, Suganthan PN, Zhang Q (2011) Multiobjective evolutionary algorithms: a survey of the state of the art. Swarm Evol Comput 1:32–49

Zuo Z, Xie YM, Huang X (2010) An improved bi-directional evolutionary topology optimization method for frequencies. Int J Struct Stab Dynam 10:55–75

Zuo Z, Xie YM, Huang X (2012) Evolutionary topology optimization of structures with multiple displacements and frequency constraints. Adv Struct Eng 15:385–398