Struct Multidisc Optim (2011) 44:629–649
DOI 10.1007/s00158-011-0649-4

RESEARCH PAPER

Topology design with negative masks using gradient search

Anupam Saxena

Received: 15 November 2010 / Revised: 8 March 2011 / Accepted: 20 March 2011 / Published online: 7 May 2011

Abstract Previous implementations of the Material Mask Overlay Scheme (MMOS) (Saxena, ASME J Mech Des 130(8):1–9, 2009, 2010; Jain and Saxena, ASME J Mech Des 132(6):1–10, 2010) use stochastic search to achieve well connected, perfectly binary topologies but are computationally expensive. Here, a gradient search is employed to lay the negative masks over the design region. A continuous material assignment model is presented and studied. This investigation is motivated by the following goals: (a) reduction in the number of evaluations thereby making the search computationally efficient compared to the previous implementations, (b) obtaining as close to well-connected binary topologies as possible, and (c) influence of various parameters on the quality of solutions and existence of gray cells.

Keywords Mask overlay method · Structural optimization · Gradient search · Binary solutions

1 Prior work

To date, many topology design algorithms using continuous approximation of the design region have been proposed and exemplified using both, stiff structures and compliant mechanisms. Among the early ones is by Bendsøe and Kikuchi (1988) who proposed the Homogenization approach. The Solid Isotropic Material with Penalization (or SIMP) approach was presented thereafter (Bendsøe 1995) that modeled cell densities as design variables. The elastic modulus E of each cell (or finite element) was approximated as $E = \rho^n E_0$ where E_0 represented the modulus of the desired material. $\rho = 0$ implied *absence* of a cell from the topology while $\rho = 1$ signified its *presence*. Parameter $n(\geq 3)$ was used to expedite the cell's material state to approach one of the two aforementioned binary values. PEAK (Yin and Ananthasuresh 2001) and SIGMOID (Saxena and Saxena 2007) were other material models used to encourage binary topologies in such *cell based* material layout approaches.

A few *node based* topology design algorithms were also proposed (e.g., Guest et al. 2004) wherein densities were associated with the nodes (and not the cells). The density at a node was projected using the linear projection and regularized Heaveside functions onto the cells. Many methods employed rectangular finite elements that exhibit well known connectivity singularities like the checkerboards, point flexures, layering and islanding. Filtering was used with cell based approaches to smear out the design and/or the associated gradients (e.g., Sigmund 2007). Mapping of material densities onto the wavelet based design space (Yoon et al. 2004) was another alternative to suppress checkerboards and de facto hinges. Wavelets help express material density in terms of basis functions which are linked to the length scales and are not directly coupled with the finite element mesh (Poulsen 2003). With the node based approaches, it was possible to impose a minimum length scale on the design explicitly (Guest et al. 2004) to prevent single node connections. Formation of local, disconnected

Electronic supplementary material The online version of this article (doi:10.1007/s00158-011-0649-4) contains supplementary material, which is available to authorized users.

A. Saxena (✉)
Indian Institute of Technology, Kanpur, 208016, India
e-mail: anupams@iitk.ac.in, saxena@igm.rwth-aachen.de

A. Saxena
Institute for Kinematics and Machine Dynamics, IGM, RWTH, 52062
Aachen, Germany

layers or islands was avoided with refined finite element meshes (Rahmatallah and Swan 2004).

To naturally eliminate the connectivity singularities, use of honeycomb parameterization was proposed (Saxena and Saxena 2003, 2007). Honeycomb representation offers only edge connectivity between cells and thus ensures finite stiffness at all junctions. Various finite element approximations were employed. E.g., a hexagonal cell was approximated using two four noded elements in three possible ways (Saxena and Saxena 2003, 2007), and using six triangular elements (Langelaar 2007). Intact, six noded Wachspress finite elements (Talischi et al. 2009) were also used to obtain the continuum response. In (Saxena 2010), isoparametric Wachspress hexagonal cells were used to accommodate irregular hexagons that appear when the notches at the contour boundaries are moderated.

Methods that do not directly associate the design with the finite element information (e.g., elements or nodes) exist as well. These use contours to impose the design on the mesh. The level set method is a general purpose approach employed to track and propagate an interface over time (Sethian and Wiegmann 2000; Wang et al. 2005; Luo et al. 2008). In topology optimization, it uses an implicit embedding function $LSM(\mathbf{x}, t)$ to evolve its contours $z = 0$ which represent the continuum boundaries. Here, $\mathbf{x} \equiv (x, y, z)$ is a Euclidean point and t represents time. Within iteration, finite element equations pertaining to the physical problem, and finite difference equations associated with the Hamilton-Jacobi level set relation are solved. The velocity field, computed using the finite element analysis, is coupled with the embedding function to drive its isophotes ($z = 0$ contours) in the subsequent time step. Some other methods that can yield binary material layouts using stochastic techniques are the sub-division scheme (Hull and Canfield 2006) and those by Chapman (Jakiela et al. 2000).

This work aims at reducing the number of function evaluations with the Material Mask Overlay Method while retaining single material (0–1) topologies as much as possible. Use of a gradient based search with a continuous material model is a natural choice. The following section briefs how MMOS determines the continuum topology. Then, continuous material assignment and sensitivity analysis are presented. The algorithm is demonstrated with standard formulations in structural optimization and many synthesis examples. Discussion and closure are drawn last.

2 Material mask overlay method: overview

Figure 1 briefly explains the working of the Material Mask Overlay Method which is based on well-known photolithographic scheme (used in MEMS or IC chip fabrication). A set of M masks, each acting as a 'material sink' is employed. A mask of any size or shape lying over some region of the domain is modeled to absorb the material beneath itself. All other regions, not exposed to the masks, remain filled. The continuum topology then gets determined by a set of unexposed cells, that is, those which are not encapsulated within any mask. Properties of the desired material are assigned to such cells. Figure 1 illustrates the use of 'negative' circular masks though the masks can be of any other shape (e.g., see Saxena 2010).

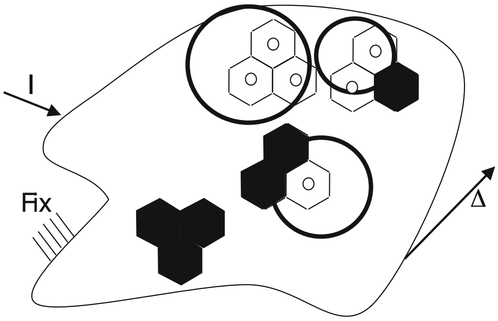


Fig. 1 Material Mask Overlay Strategy illustrated with negative masks. Masks of circular shape (*thick circles*) are used for material removal. *Hexagonal cells* whose centroids (*small circles*) are inside a mask are empty (*white*) while those outside the masks are all filled (*black*) with the desired material. Key: *Fix* fixed boundary, *I* input port, *triangle* expected output deformation(in case of monolithic compliant continua)

Previous implementations of MMOS employed circular (Saxena 2009; Jain and Saxena 2010), elliptic and rectangular masks (Saxena 2010). Circular masks were found to be adequate in determining the material layout (Saxena 2010). Elliptical (that modestly represent simple closed curves) and rectangular (those representing closed polygons) masks do not bring in notable merits. Instead, two variables per mask get additionally introduced in the search space. Both, genetic algorithm (Saxena 2009) and simple mutation based hill climber (Jain and Saxena 2010; Saxena 2010) stochastic algorithm were used in single and a sequence of sub-searches. Additionally, adaptive schemes were used (Saxena 2010) wherein the number of masks (and hence the number of decision variables) was judiciously varied as the search progresses. While MMOS yields well connected binary solutions, the associated stochastic search is computationally expensive. Previous improvements in MMOS were successful in lowering the number of function evaluations from many thousands (∼100,000 or more) to a few thousand. Even then, the computational efficiency of the gradient based approaches is not matched. It is possible to develop analytical material models with MMOS so that a gradient search can be employed. One such possibility is explored below.

3 Material mask overlay method and gradient search

3.1 Material model

The material assignment approximation used herein employs the logistic function $f(\alpha, t)$ given by

$$f(\alpha, t) = \frac{1}{1 + e^{-\alpha t}} \tag{1}$$

where α is a parameter. A sigmoid function (Saxena and Saxena 2007) or any other continuous regularization of the Heaveside step function is applicable. Figure 2 depicts the behavior of $f(\alpha, t)$ as α is varied. For large α, $f(\alpha, t) \approx 0$ or 1 for $t \in \{-\infty, \infty\} - \{-\delta, \delta\}$ where δ depends on α. With increasing α, δ decreases which makes the interval $\{-\delta, \delta\}$ smaller. If $\alpha \to \infty$, $f(\alpha, t) \to H(t)$ which is the Heaveside function defined as $H(t) = 1$ for $t > 0$, and $H(t) = 0$ otherwise. The density ρ_j of the jth cell can be associated with a circular mask M_K identified by (x_K, y_K, R_K) where (x_K, y_K) are its center coordinates and R_K is its radius. This relation can be expressed as

$$\rho_j \propto \frac{1}{1 + e^{-\alpha(d_{jK} - R_K)}} \tag{2}$$

Here $d_{jK} = \sqrt{(x_K - x_j)^2 + (y_K - y_j)^2}$ is the distance between the centroid (x_j, y_j) of the jth cell and the center of M_K. For large α, when $d_{jK} < R_K$, $\rho_j \approx 0$ while when $d_{jK} > R_K$, $\rho_j \approx 1$.This is consistent with the notion of negative masks introduced in (Saxena 2009, 2010; Jain and Saxena 2010). As required, cells enclosed within a mask attain the void state while those outside the mask retain the desired material. If $d_{jK} = R_K$, $\rho_j = 1/2$ (irrespective of how large α is) implying that a cell whose centroid lies very near or on the mask boundary is neither fully void ($\rho_j = 0$) nor fully solid ($\rho_j = 1$). Such cells get assigned fictitious material states which are not intended in topology optimization. To model the cumulative effect of all M masks on the material state of the jth cell, individual terms in (2) can all be multiplied. Thus, the cell density is now expressed as

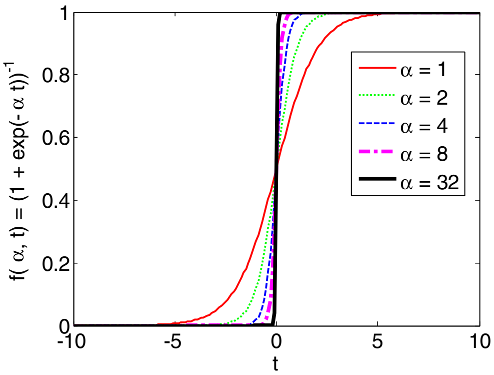


Fig. 2 The logistic function used inthe material model. Higher values of α encourage 0–1 material assignment

$$\rho_j = \prod_{k=1}^{Ma} \left\{ \frac{1}{1 + e^{-\alpha(d_{jk} - R_k)}} \right\} + \varepsilon \tag{3}$$

where Ma is the number of masks. Equation (3) suggests that if a cell's centroid is not encapsulated within any mask, the cell is filled with material. Otherwise, the material is absorbed by the mask(s) laying over that cell. A small positive ε is added to ensure that the overall stiffness matrix stays non-singular throughout the small deformation analysis.

3.2 Gradient computation

The position and size of each circular mask over the domain determine the material layout governed by the objective and a set of constraints used. To determine these variables using a gradient search algorithm, design sensitivities (function/constraint gradients) are computed below. Let Φ represent a function or constraint and let ψ_K represent the variables x_K, y_K or R_K for the Kth mask with respect to which the derivatives are needed. Using chain rule

$$\frac{\partial \Phi}{\partial \psi_K} = \sum_{j=1}^{Nc} \left[\frac{\partial \Phi}{\partial \rho_j} \right] \left[\frac{\partial \rho_j}{\partial \psi_K} \right] \tag{4}$$

where Nc is the number of cells. $\frac{\partial \Phi}{\partial \rho_j}$ is usually available for a design problem after the finite element analysis (see Section 4.1). To compute $\frac{\partial \rho_j}{\partial \psi_K}$, let C be given by

$$C = \prod_{\substack{p=1 \\ p \neq K}}^{Ma} \left\{ \frac{1}{1 + e^{-\alpha(d_{jp} - R_p)}} \right\} \tag{5}$$

The contribution of the Kth mask on the material density ρ_j is given by $f_K(\alpha, d, R)$ where

$$f_K(\alpha, d, R) = \frac{1}{1 + e^{-\alpha(d_{jK} - R_K)}} \tag{6}$$

Then

$$\frac{\partial f_K(\alpha, d, R)}{\partial \psi_K} = \frac{\alpha e^{-\alpha(d_{jK} - R_K)}}{\left[1 + e^{-\alpha(d_{jK} - R_K)}\right]^2} \frac{\partial (d_{jK} - R_K)}{\partial \psi_K} \tag{7}$$

If $\psi_K \equiv x_K$,

$$\frac{\partial (d_{jK} - R_K)}{\partial \psi_K} = \frac{(x_K - x_j)}{d_{jK}} \tag{8}$$

If $\psi_K \equiv y_K$,

$$\frac{\partial\left(d_{jK}-R_K\right)}{\partial\psi_K}=\frac{\left(y_K-y_j\right)}{d_{jK}} \tag{9}$$

and if $\psi_K \equiv R_K$, then

$$\frac{\partial\left(d_{jK}-R_K\right)}{\partial\psi_K}=-1 \tag{10}$$

Finally, $\frac{\partial\rho_j}{\partial\psi_K}=C\frac{\partial f_K(\alpha,d,R)}{\partial\psi_K}$ which requires (7–10). $\frac{\partial\rho_j}{\partial\psi_K}$ can be simplified using (3), (5) and (7) as

$$\frac{\partial\rho_j}{\partial\psi_K}=\rho_j\frac{\alpha e^{-\alpha\left(d_{jK}-R_K\right)}}{\left[1+e^{-\alpha\left(d_{jK}-R_K\right)}\right]}\frac{\partial\left(d_{jK}-R_K\right)}{\partial\psi_K} \tag{11}$$

3.3 Discontinuity of derivatives

The derivatives in (11) are discontinuous for $\psi_K \equiv x_K$ and y_K when the center of the Kth mask tends to coincide with the centroid of the jth cell (i.e., when $d_{jK}\to 0$). Let (x_K, y_K) approach (x_j, y_j) along a line $y-y_j=m(x-x_j)$. Let the mask radius R_K be constant. Then

$$\begin{aligned}
&\lim_{(x_K,y_K)\to(x_j,y_j)}\frac{\partial\rho_j}{\partial\psi_K}\\
&=\lim_{(x_K,y_K)\to(x_j,y_j)}\rho_j\frac{\alpha e^{-\alpha\left(d_{jK}-R_K\right)}}{\left[1+e^{-\alpha\left(d_{jK}-R_K\right)}\right]}\frac{\partial\left(d_{jK}-R_K\right)}{\partial\psi_K}\\
&=\lim_{(x_K,y_K)\to(x_j,y_j)}C\frac{\alpha e^{-\alpha\left(d_{jK}-R_K\right)}}{\left[1+e^{-\alpha\left(d_{jK}-R_K\right)}\right]^2}\frac{\partial\left(d_{jK}-R_K\right)}{\partial\psi_K}\\
&=C\frac{\alpha e^{\alpha(R_K)}}{\left[1+e^{\alpha(R_K)}\right]^2}\lim_{(x_K,y_K)\to(x_j,y_j)}\frac{\partial\left(d_{jK}-R_K\right)}{\partial\psi_K}\\
&=C\frac{\alpha e^{\alpha(R_K)}}{\left[1+e^{\alpha(R_K)}\right]^2}\lim_{(x_K,y_K)\to(x_j,y_j)}\frac{\left(x_K-x_j\right)}{\left(x_K-x_j\right)\sqrt{1+m^2}}\\
&=C\frac{\alpha e^{\alpha(R_K)}}{\left[1+e^{\alpha(R_K)}\right]^2}\frac{1}{\sqrt{1+m^2}}\ \text{for}\ \psi_K\equiv x_K
\end{aligned} \tag{12}$$

and

$$\begin{aligned}
&\lim_{(x_K,y_K)\to(x_j,y_j)}\frac{\partial\rho_j}{\partial\psi_K}\\
&=C\frac{\alpha e^{\alpha(R_K)}}{\left[1+e^{\alpha(R_K)}\right]^2}\frac{1}{\sqrt{1+\frac{1}{m^2}}}\ \text{for}\ \psi_K\equiv y_K
\end{aligned} \tag{13}$$

For any (x_K, y_K) and (x_j, y_j), if $R_K\to 0$, then

$$\begin{aligned}
&\lim_{R_K\to 0}\frac{\partial\rho_j}{\partial_{\psi_K}}\\
&=\lim_{R_K\to 0}C\frac{\alpha e^{\alpha(d_{jK}-R_K)}}{\left[1+e^{-\alpha(d_{jK}-R_K)}\right]^2}\frac{\partial(d_{jK}-R_K)}{\partial_{\psi_K}}\\
&=C\lim_{R_K\to 0}\frac{\alpha e^{-\alpha(d_{jK}-R_K)}}{\left[1+e^{-\alpha(d_{jK}-R_K)}\right]^2}\frac{\partial(d_{jK}-R_K)}{\partial_{\psi_K}}\\
&=C\frac{\alpha e^{-\alpha(d_{jK})}}{\left[1+e^{-\alpha(d_{jK})}\right]^2}\frac{\partial(d_{jK}-R_K)}{\partial_{\psi_K}}
\end{aligned}$$

which is well behaved if $(x_K,\ y_K)$ is not close to $(x_j,$ $y_j)$. However, the values of the limits in (12), (13) and the one above vary if $(x_K,\ y_K)$ approaches $(x_j,\ y_j)$ along different lines (i.e., those with different slopes m) which is why the derivatives in (11) are discontinuous. Now, let d_{jK} be computed differently such that $d_{jK}=\sqrt{\left(x_j-x_K\right)^2+\left(y_j-y_K\right)^2+p}$ for very small but positive p. This change does not alter any of the equations above. Taking the limits as $(x_K, y_K)\to(x_j, y_j)$ gives

$$\begin{aligned}
&\lim_{(x_K,y_K)\to(x_j,y_j)}\rho_j\frac{\alpha e^{-\alpha(d_{jK}-R_K)}}{\left[1+e^{-\alpha(d_{jK}-R_K)}\right]}\frac{\partial(d_{jK}-R_K)}{\partial_{\psi_k}}\\
&=C\frac{\alpha e^{\alpha(R_K)}}{\left[1+e^{\alpha(R_K)}\right]^2}\lim_{(x_K,y_K)\to(x_j,y_j)}\frac{\partial(d_{jK}-R_K)}{\partial_{\psi_k}}
\end{aligned}$$

For $\psi_K\equiv x_K$

$$\begin{aligned}
&\lim_{(x_K,y_K)\to(x_j,y_j)}\frac{\partial\left(d_{jK}-R_K\right)}{\partial\psi_K}\\
&=\lim_{(x_K,y_K)\to(x_j,y_j)}\frac{\left(x_K-x_j\right)}{\sqrt{\left(x_j-x_K\right)^2\left(1+m^2\right)+p}}=0\ \forall m.
\end{aligned} \tag{14}$$

A similar observation can be made for $\psi_K\equiv \mathrm{y}_K$. For small p, $\sqrt{\left(x_j-x_K\right)^2+\left(y_j-y_K\right)^2+p}\approx\sqrt{\left(x_j-x_K\right)^2+\left(y_j-y_K\right)^2}$ if $(\mathrm{x}_K,\ \mathrm{y}_K)$ and $(\mathrm{x}_j,\ \mathrm{y}_j)$ are quite far apart. Thus, the cell densities may be computed as in (3). However, adding a small positive number to the denominators in (8) and (9) has a similar effect as in (14). Thus

$$\frac{\partial\rho_j}{\partial x_K}=\rho_j\left(\frac{\alpha\exp\left(-\alpha\left\{d_{jK}-R_K\right\}\right)}{1+\exp\left(-\alpha\left\{d_{jK}-R_K\right\}\right)}\right)\left(\frac{x_K-x_j}{d_{jK}+\varepsilon}\right) \tag{15}$$

$$\frac{\partial\rho_j}{\partial y_K}=\rho_j\left(\frac{\alpha\exp\left(-\alpha\left\{d_{jK}-R_K\right\}\right)}{1+\exp\left(-\alpha\left\{d_{jK}-R_K\right\}\right)}\right)\left(\frac{y_K-y_j}{d_{jK}+\varepsilon}\right) \tag{16}$$

$$\frac{\partial \rho_j}{\partial R_K} = -\rho_j \left(\frac{\alpha \exp\left(-\alpha \left\{d_{jK} - R_K\right\}\right.}{1 + \exp\left(-\alpha \left\{d_{jK} - R_K\right\}\right)} \right) \tag{17}$$

3.4 Behavior of the derivatives as $\alpha \to \infty$

All derivatives in (15)–(17) are of the form

$$\frac{\partial \rho_j}{\partial \psi_K} = C_1 \left(\frac{\alpha \exp(-\alpha\{C_2\})}{1 + \exp(-\alpha\{C_2\})} \right)$$

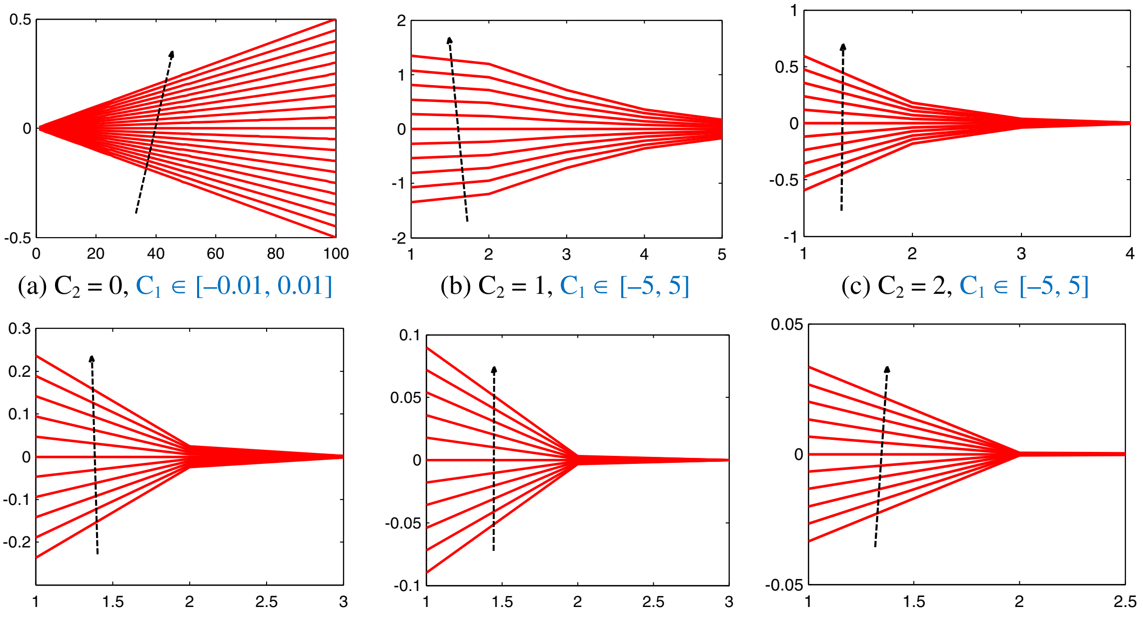
where $C_1 = \rho_j \left(\frac{x_K - x_j}{d_{jK}+\varepsilon}\right)$, $\rho_j \left(\frac{y_K - y_j}{d_{jK}+\varepsilon}\right)$ or $-\rho_j$ and $C_2 = d_{jK} - R_K$. Note that C_1 determines the sign of $\frac{\partial \rho_j}{\partial \psi_K}$. The plot of the above expression vs. α is shown in Fig. 3 for various values of C_1 and C_2. For C_2 less than or equal to 0, the jth cell is inside the Kth mask and its density value is close to the lower bound ε. Further, $\left\|\frac{x_K - x_j}{d_{jK}+\varepsilon}\right\|$ and $\left\|\frac{y_K - y_j}{d_{jK}+\varepsilon}\right\| \leqslant 1$ implying that C_1 is small and cannot be chosen independently. Figure 3a shows the plot of $\frac{\partial \rho_j}{\partial \psi_K}$ for $C_2 = 0$ and $C_1 \in [-0.01, 0.01]$. Except for $C_1 = 0$ the magnitudes of the derivatives are all comparatively larger than zero. Further, the derivatives are proportional to C_1. This implies that all cells within a mask influence its position and size.

If the jth cell is outside the Kth mask (i.e., $C_2 > 0$), C_1 can be chosen independently. Figure 3b–f show these plots for $C_1 \in [-5, 5]$. It is observed that the derivatives converge to zero asymptotically implying that the parameters of the Kth mask are not influenced significantly by a cell which is far from its center. This is expected. The magnitudes of all derivatives converge to zero at a much faster rate for $\alpha > 2$ which can cause numerical instabilities in the algorithm (see discussion).

The proposed material model with negative masks can be compared to the projection schemes in (Guest et al. 2004; Sigmund 2007). In (Guest et al. 2004), fixed solid circular features are used while in (Sigmund 2007), fixed void circular features are employed. Further in (Sigmund 2007), structures are defined in the modified projection method via circular regions with fixed radius. Here, circular masks are free to relocate and vary in size, and their number is independent of the number of cells used in the discretization. The minimum member size is controlled by the size of the hexagonal cell which makes a solution mesh dependent unless the feature size is indirectly imposed via some constraint in the optimization formulation.

4 Examples

We employ the *fmincon* function available with the optimization toolbox in MATLAB^{TM} to illustrate topology optimization using negative masks. *fmincon* is a general purpose algorithm that uses many features and parameters. The examples are solved herein with the medium-scale optimization (line search) option that uses Sequential Quadratic Programming (SQP) as one of the



(a) $C_2 = 0$, $C_1 \in [-0.01, 0.01]$ (b) $C_2 = 1$, $C_1 \in [-5, 5]$ (c) $C_2 = 2$, $C_1 \in [-5, 5]$

(d) $C_2 = 3$, $C_1 \in [-5, 5]$ (e) $C_2 = 4$, $C_1 \in [-5, 5]$ (f) $C_2 = 5$, $C_1 \in [-5, 5]$

Fig. 3 a–f Plots of the derivatives (*vertical axis*) with respect to α (*horizontal axis*) for various values of C_2. In each plot, the *arrow* indicates increasing values of C_1

search algorithms. A compact educational description of SQP is provided in (Epelman 2010). The MATLAB code (main file: mmos_main.m) used is provided in Appendix A with a brief explanation. It is available via the journal website and can also be downloaded from home.iitk.ac.in/~anupams/mmos81.zip. Many classical examples on optimal stiff structures and compliant mechanisms are solved (Fig. 4). Effect of the grid size, number and maximum size of the negative masks, and other parameters are studied. Filtering (e.g., Sigmund 2007; Wang et al. 2010; Bourdin 2001) or projection (e.g., Guest 2009; Xu et al. 2010) methods are avoided to explore the true nature of the formulation. It is shown in (Saxena 2010) that the number of masks actually required for topology determination can differ from the initially chosen number. Here however, masks are not added or deleted within the iterations.

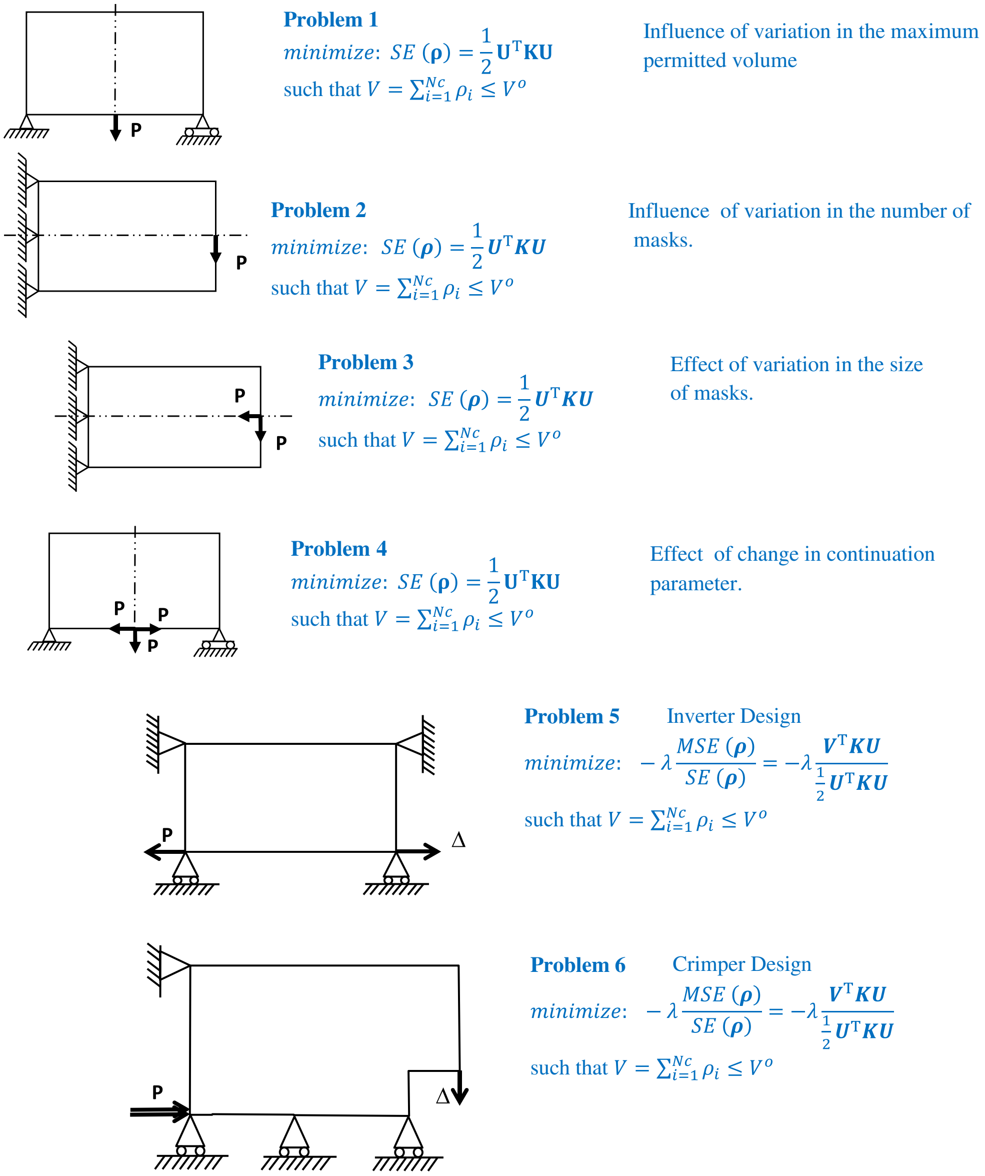
4.1 Small deformation stiff structures and compliant mechanisms

A standard formulation to design stiff structures is to minimize the mean compliance (or strain energy, *SE*) under a resource constraint. Thus

$$\begin{aligned} &minimize: SE(\boldsymbol{\rho}) = \frac{1}{2}\mathbf{U}^T\mathbf{K}\mathbf{U} \\ &\text{such that } V = \textstyle\sum_{i=1}^{Nc} \rho_i \leq V^o \end{aligned} \tag{F1}$$

where $\boldsymbol{\rho} = [\rho_1, \rho_2, \ldots, \rho_N]$ is a set of cell densities, V represents the volume fraction and V^o denotes its upper bound.

Fig. 4 Various classical problems in stiff structure (*Problems 1*–4) and compliant mechanism (*Problems 5*–6) design. Key: *Bold arrows* show the applied loads **P** and/or the direction of the desired deformation Δ. Fixed nodes and edges on roller support are also shown



Problem 1

$minimize$: $SE\,(\boldsymbol{\rho}) = \frac{1}{2}\mathbf{U}^{\mathrm{T}}\mathbf{KU}$

such that $V = \sum_{i=1}^{Nc} \rho_i \leq V^o$

Influence of variation in the maximum permitted volume

Problem 2

$minimize$: $SE\,(\boldsymbol{\rho}) = \frac{1}{2}\boldsymbol{U}^{\mathrm{T}}\boldsymbol{KU}$

such that $V = \sum_{i=1}^{Nc} \rho_i \leq V^o$

Influence of variation in the number of masks.

Problem 3

$minimize$: $SE\,(\boldsymbol{\rho}) = \frac{1}{2}\boldsymbol{U}^{\mathrm{T}}\boldsymbol{KU}$

such that $V = \sum_{i=1}^{Nc} \rho_i \leq V^o$

Effect of variation in the size of masks.

Problem 4

$minimize$: $SE\,(\boldsymbol{\rho}) = \frac{1}{2}\mathbf{U}^{\mathrm{T}}\mathbf{KU}$

such that $V = \sum_{i=1}^{Nc} \rho_i \leq V^o$

Effect of change in continuation parameter.

Problem 5 Inverter Design

$minimize$: $-\lambda\frac{MSE\,(\boldsymbol{\rho})}{SE\,(\boldsymbol{\rho})} = -\lambda\frac{\boldsymbol{V}^{\mathrm{T}}\boldsymbol{KU}}{\frac{1}{2}\boldsymbol{U}^{\mathrm{T}}\boldsymbol{KU}}$

such that $V = \sum_{i=1}^{Nc} \rho_i \leq V^o$

Problem 6 Crimper Design

$minimize$: $-\lambda\frac{MSE\,(\boldsymbol{\rho})}{SE\,(\boldsymbol{\rho})} = -\lambda\frac{\boldsymbol{V}^{\mathrm{T}}\boldsymbol{KU}}{\frac{1}{2}\boldsymbol{U}^{\mathrm{T}}\boldsymbol{KU}}$

such that $V = \sum_{i=1}^{Nc} \rho_i \leq V^o$

$\mathbf{U}$ is the overall displacement response of an intermediate design computed using linear finite element analysis and $\mathbf{K}$ is the global stiffness matrix. Vector $\mathbf{F} = \mathbf{KU}$ comprises of a set of input static loads. Gradients of the strain energy with respect to the cell densities can be determined as

$$\frac{\partial SE(\rho)}{\partial \rho_i} = -\frac{1}{2}(\mathbf{u}_i)^{\mathrm{T}} \frac{\partial k_i}{\partial \rho_i}\mathbf{u}_i \tag{18}$$

where $\mathbf{u}_i$ are the nodal displacements (12 components of $\mathbf{U}$) of the ith hexagonal cell and $\mathbf{k}_i$ is its stiffness matrix. If $\mathbf{k}_i = \rho_i \mathbf{k}^{\mathrm{o}}$ where $\mathbf{k}^{\mathrm{o}}$ is the stiffness of the solid cell, then $\frac{\partial k_i}{\partial \rho_i} = \mathbf{k}^{\mathrm{o}}$. Let ψ_j represent one of the three design variables of the jth mask. Then, using chain rule, $\frac{\partial SE(\boldsymbol{\rho})}{\partial \psi_j}$ can be expressed as

$$\frac{\partial SE(\boldsymbol{\rho})}{\partial \psi_j} = \sum\nolimits_{i=1}^{Nc}\left(\frac{\partial SE(\boldsymbol{\rho})}{\partial \rho_i}\frac{\partial \rho_i}{\partial \psi_j}\right) \tag{19}$$

where $\frac{\partial \rho_j}{\partial \psi_K}$ are available from (15)–(17).

To design an optimal compliant topology, we use the flexibility-stiffness formulation (Saxena and Ananthasuresh 2000). This multi-criteria objective allows maximizing the desired output deformation Δ (see Fig. 1) and minimizing the internal energy stored within the continuum to make it capable of sustaining the external loads. An output spring along the direction of the desired deformation ensures force transfer or non-zero mechanical advantage. In standard form, the flexibility-stiffness objective is written as

$$\begin{aligned} &minimize: -\frac{MSE(\boldsymbol{\rho})}{SE(\boldsymbol{\rho})} = -\lambda\frac{\mathbf{V}^{\mathrm{T}}\mathbf{KU}}{\frac{1}{2}\mathbf{U}^{\mathrm{T}}\mathbf{KU}} \\ &such\ that\ V = \sum\nolimits_{i=1}^{Nc}\rho_i \le V^o \end{aligned} \tag{F2}$$

Here, $\mathbf{V}$ is the displacement response obtained by applying a unit dummy force along the direction of the desired deformation (Yin and Ananthasuresh 2003). That is, if $\mathbf{F}_d$ is a force vector that contains the dummy load, then $\mathbf{F}_d = \mathbf{KV}$. *MSE* or mutual strain energy is the same as the desired output deformation. λ is the scaling constant required to avoid early convergence to a local minimum (see discussion). Other formulations like the maximization of the mechanical advantage or geometric advantage are similar to (F2) but with different scale factors. Derivatives of the strain energy with respect to mask parameters can be obtained using (18) and (19). Gradients of the output displacement can be computed as

$$\frac{\partial MSE(\boldsymbol{\rho})}{\partial \rho_i} = -(\mathbf{v}_i)^{\mathrm{T}}\frac{\partial \mathbf{k}_i}{\partial \rho_i}\mathbf{u}_i \tag{20}$$

where $\mathbf{v}_i$ are the nodal components of $\mathbf{V}$ for the ith cell. Like in (19), $\frac{\partial MSE(\boldsymbol{\rho})}{\partial \psi_j}$ can also be evaluated using the chain rule. That is

$$\frac{\partial MSE(\boldsymbol{\rho})}{\partial \psi_j} = \sum\nolimits_{i=1}^{Nc}\left(\frac{\partial MSE(\boldsymbol{\rho})}{\partial \rho_i}\frac{\partial \rho_i}{\partial \psi_j}\right) \tag{21}$$

Now, for the gradients of the objective in (F2), i.e., for $\frac{\partial}{\partial \psi_j}\left(-\frac{MSE(\boldsymbol{\rho})}{SE(\boldsymbol{\rho})}\right)$, we have

$$\begin{aligned} &\frac{\partial}{\partial \psi_j}\left(-\frac{MSE(\boldsymbol{\rho})}{SE(\boldsymbol{\rho})}\right) \\ &= -\left\{\frac{\frac{\partial}{\partial \psi_j}MSE(\boldsymbol{\rho})}{SE(\boldsymbol{\rho})} - \frac{MSE(\boldsymbol{\rho})\frac{\partial}{\partial \psi_j}SE(\boldsymbol{\rho})}{[SE(\boldsymbol{\rho})]^2}\right\} \end{aligned} \tag{22}$$

which uses (15)–(21).

For all examples below, $Ma = M \times N$ user-specified masks are placed uniformly over the domain. M is the number of masks along the horizontal direction while N is the same along the vertical direction. Masks are allowed to move outside the design region. Masks can be positioned within the area $(L + 2R_{\max}) \times (W + 2R_{\max})$ where L and $W$are the length and width of the region and $R_{\max}$ is the maximum permitted user-specified radius of the mask.

Convergence for optimization is governed by the parameters *max_iter*, *max_eval*, *tol_fun* and *tol_var*. *max_iter* is the maximum number of iterations permitted, *max_eval* corresponds to the upper bound on the number of function evaluations, and *tol_fun* and *tol_var* are the minimum tolerances on the function and variable values. For all examples, *tol_fun* = *tol_var* = 10^{-10} is used. Different values for the permitted number of function evaluations and iterations are used in different cases.

The focus is to obtain close to binary solutions which can be achieved if the exponent is large. However, for large α, most derivatives in (15)–(17) will be close to zero which may lead to numerical problems (see Section 3.4). Thus, a continuation approach is used wherein α is initially chosen small and is gradually increased with the number of function evaluations. It is expected that with increasing α, the sequence of intermediate designs will converge to a 0–1 solution. Further, the exponents used to compute the objective and resource constraint are decoupled. That is, two exponents α_{f} (for objective and its gradients) and α_{v} (for volume constraint and its derivatives) are used. In the examples presented, α_{f} is gradually increased with iterations while α_{v} is held constant at a low value (for all examples presented, $\alpha_{\mathrm{v}} = 1$) to prevent early convergence. This therefore permits the upper bound V^{o} to control the continuum volume only indirectly which is studied through Problem 1 (Fig. 4).

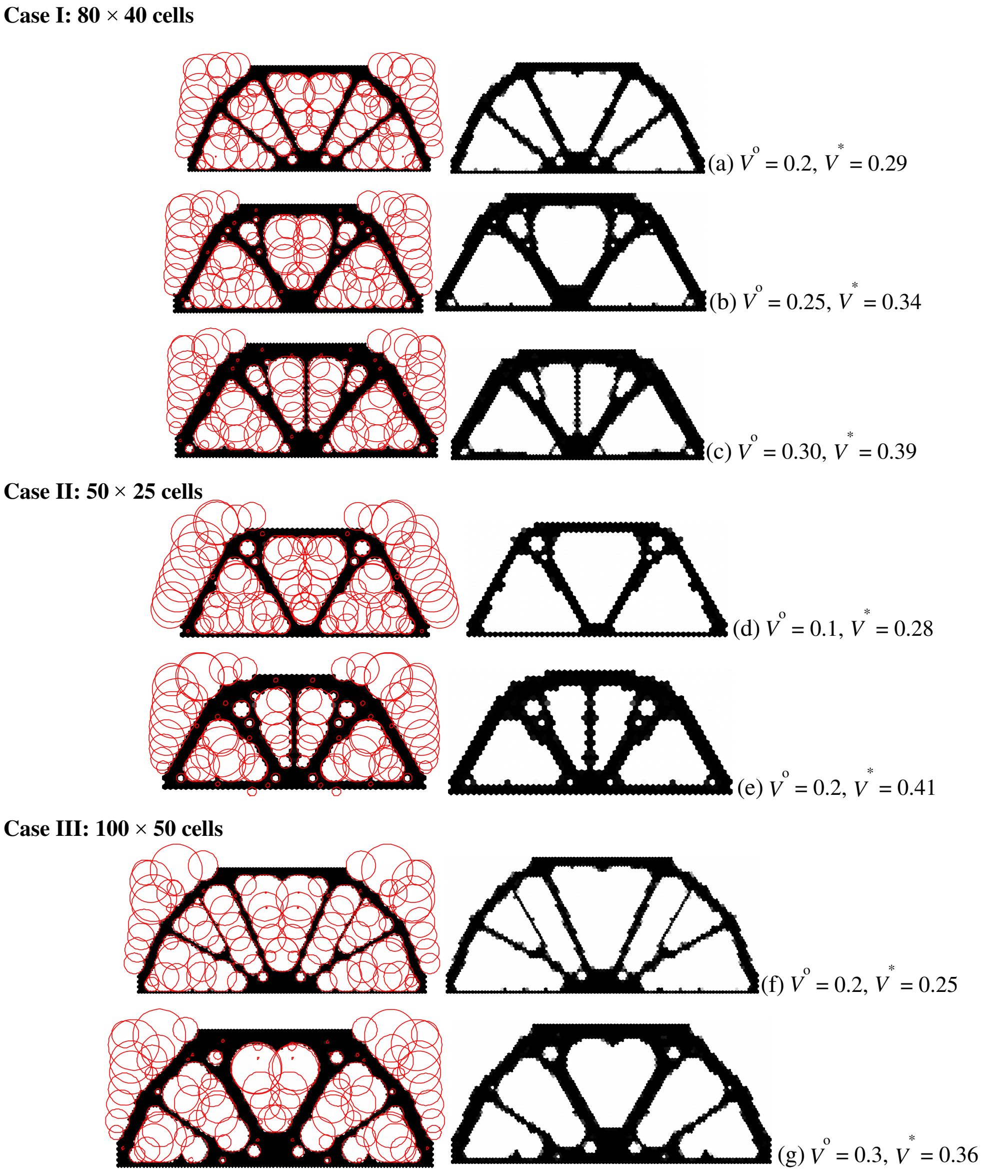
4.2 Results

4.2.1 Upper bound on the volume fraction

Problem 1 is solved with the elastic modulus E as 1,000 N mm^{-2}, thickness t as 1 mm, α_{f} (initial) = $\alpha_{\mathrm{v}} = 1$, and $R_{\max} = W/3$. The problem is studied with different mesh sizes while maintaining the L/W ratio. Case I corresponds to a mesh with 80 × 40 cells, case II to 50 × 25 cells, and case III to 100 × 50 cells. Load **P** is taken as 20 N for all cases. For the examples, $\alpha_{\mathrm{v}} = 1$ while α_{f} is gradually increased as $\alpha_{\mathrm{f}} = f_{\mathrm{cont}}\,\alpha_{\mathrm{f}}$ after every function evaluation where f_{cont} is the continuation parameter. f_{cont} is chosen as 1.014. The solutions are generated for *max_iter* $= 100$ and *max_eval* $= 400$. The upper bound V^{o} on the continuum volume is varied for different cases as shown in Fig. 5. The final volume fractions V^* computed using the cell densities of the solutions differ from the respective initial fractions V^{o}. This difference is large for coarse meshes. V^* gets comparable to V^{o} as the number of cells is increased.

4.2.2 Number of masks

Problem 2 is solved to investigate the effect of change in the number of user-specified masks on the solutions. The elastic modulus and thickness are the same as in Problem 1. The upper bound on the volume fraction is 0.2. All examples are solved for the mesh with 80 × 41 cells. The maximum mask radius is set as $R_{\max} = W/3 = 41/3$. As before, α_{f} (initial) $= \alpha_{\mathrm{v}} = 1$ and α_{f} is gradually increased after every function evaluation while retaining the value of α_{v}. The continuation parameter f_{cont} is 1.014. Two loads of $\mathbf{P} = -10$ N each are applied just above and below the horizontal line of symmetry (nodes do not exist on this line). The problem is solved for three cases. The number of masks are varied as



Case I: 80 × 40 cells
(a) $V^{\mathrm{o}} = 0.2$, $V^* = 0.29$
(b) $V^{\mathrm{o}} = 0.25$, $V^* = 0.34$
(c) $V^{\mathrm{o}} = 0.30$, $V^* = 0.39$
Case II: 50 × 25 cells
(d) $V^{\mathrm{o}} = 0.1$, $V^* = 0.28$
(e) $V^{\mathrm{o}} = 0.2$, $V^* = 0.41$
Case III: 100 × 50 cells
(f) $V^{\mathrm{o}} = 0.2$, $V^* = 0.25$
(g) $V^{\mathrm{o}} = 0.3$, $V^* = 0.36$

Fig. 5 **a–g** Topologies for Problem 1 obtained using three mesh sizes. Solutions are shown with (*left*) and without (*right*) masks. The upper bound V^{o} on the volume fraction is varied as shown for the respective cases. V^* is the volume fraction computed with the final cell densities

$M \times N = 15 \times 10$, 10×10 and 20×12 respectively. It is observed in problem 1 that when α_f is increased, many (but not all) cells are close to their binary states. Here therefore, α_f is allowed to increase further by setting *max_iter* to 150. Alternatively, α_f can also be increased by choosing larger f_cont which is explored in Problem 4. Maximum permitted function evaluations are 400. The solutions are depicted in Fig. 6.

4.2.3 Permitted upper bound on the mask size

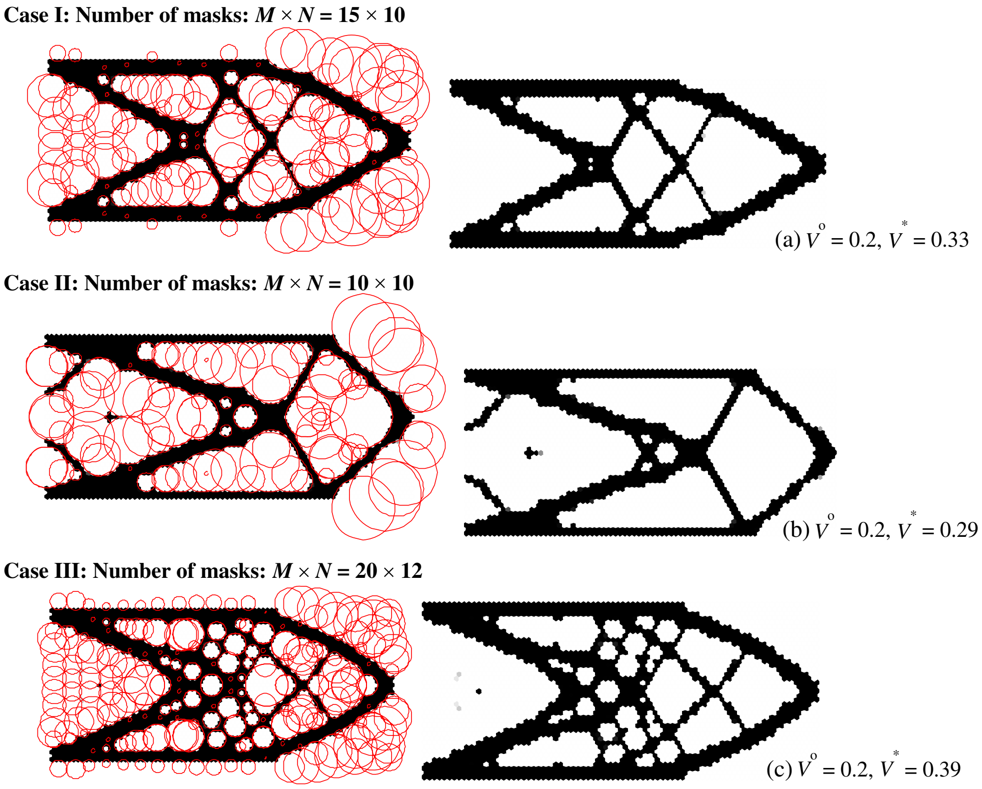
Problem 3 involves minimization of the mean compliance (strain energy) under multiple loads. Two loads of equal magnitudes act at the middle right edge (Fig. 4). Each load **P** is 40 N. For multiple load problems, the strain energy is computed individually and their sum is minimized. Correspondingly, their gradients are also added. The mesh used for Problem 3 comprises 60×31 cells. Other parameters like the modulus and thickness are the same as the first two problems. The continuation factor is changed to 1.01 and the maximum number of permitted iterations is lowered to 120. Four cases are solved with $M \times N = 15 \times 10$ masks with the upper bound on the volume fraction as 0.1. In these, R_max is varied as $W/3$, $W/2$ and W and $W/6$ respectively where $W = 31$, the number of cells along the vertical direction. Figure 7 depicts the resultant topologies.

4.2.4 Continuation parameter

The final stiffness minimization problem is solved for the multi-load case again shown as Problem 4 in Fig. 4. Here, the effect of convergence to the Heaveside function is studied. The parameter choices are different from the first three problems. The elastic modulus and continuum thickness are changed to 100 N mm^{-2} and 5 mm respectively. Number of masks along the horizontal and vertical directions are 14 each. The problem is solved with a fixed mesh size of 50 cells $\times$ 25 cells. The upper bound V^o on the volume fraction is changed to 0.05. Maximum number of iterations is set to 120. For four individual cases, the final values of α_f are desired as 10, 20, 30 and 60 after 200 evaluations. Accordingly, the continuation parameter f_cont is computed as $\exp(\frac{1}{200}\log[\alpha_\mathrm{f}]) = 1.0116, 1.0151, 1.0172$ and 1.0207 and the permitted number of design evaluations *max_eval* is set to 200. The designs are shown in Fig. 8.

4.2.5 Compliant inverter and crimper

We next synthesize the top symmetric half of a compliant inverter (Problem 5 in Fig. 4) for two different mesh sizes using the formulation in (F2). In the first case, the mesh is chosen to comprise 40×20 cells. The magnitude


Case I: Number of masks: $M \times N = 15 \times 10$

(a) $V^\mathrm{o} = 0.2$, $V^* = 0.33$

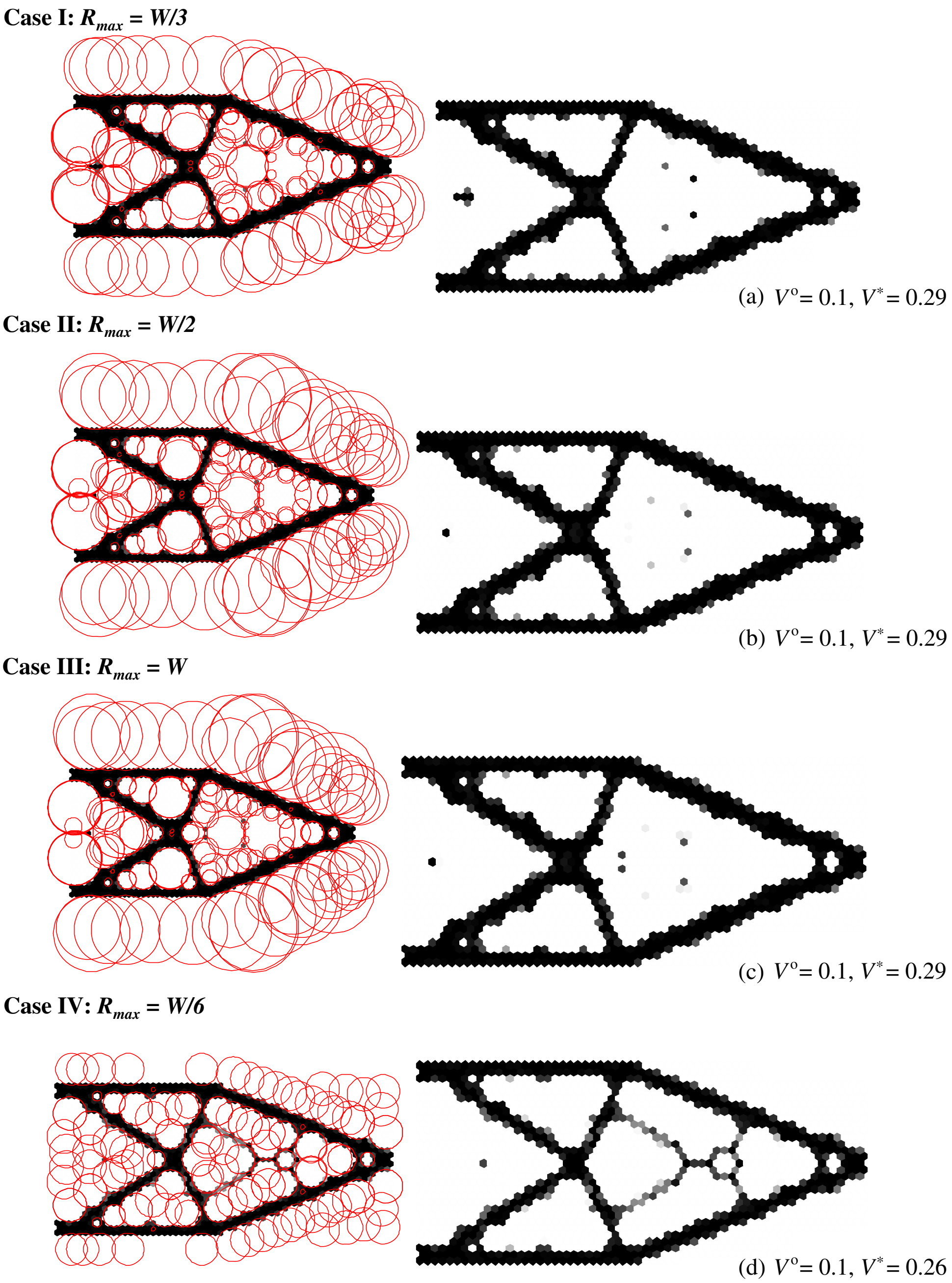
Case II: Number of masks: $M \times N = 10 \times 10$

(b) $V^\mathrm{o} = 0.2$, $V^* = 0.29$

Case III: Number of masks: $M \times N = 20 \times 12$

(c) $V^\mathrm{o} = 0.2$, $V^* = 0.39$

Fig. 6 a–c Solutions to Problem 2. Three cases are considered wherein the number of masks is varied in each



Case I: $R_{max} = W/3$

(a) $V^{o} = 0.1$, $V^{*} = 0.29$

Case II: $R_{max} = W/2$

(b) $V^{o} = 0.1$, $V^{*} = 0.29$

Case III: $R_{max} = W$

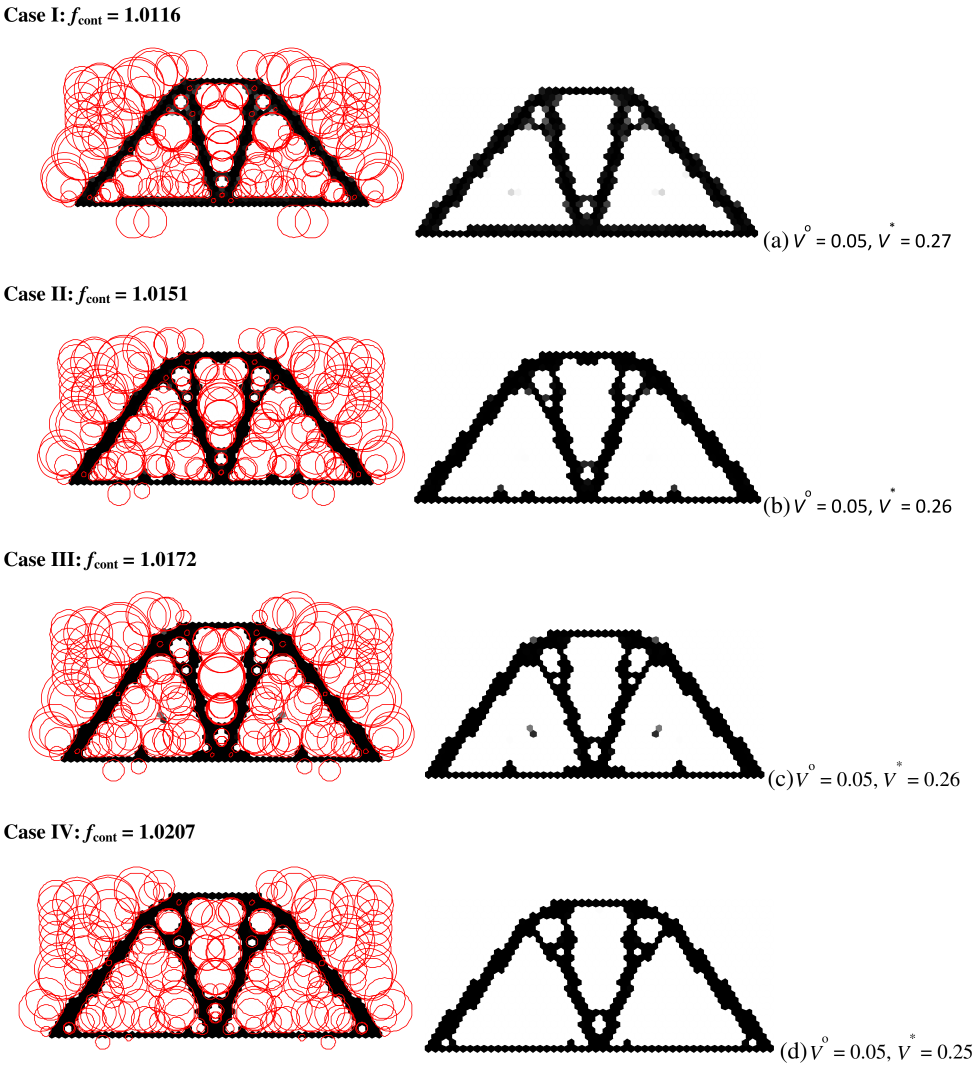
(c) $V^{o} = 0.1$, $V^{*} = 0.29$

Case IV: $R_{max} = W/6$

(d) $V^{o} = 0.1$, $V^{*} = 0.26$

Fig. 7 a–d Results for Problem 3. In the four cases considered, the maximum permitted mask size is varied

of load **P** is 40 N at the left bottom corner. The right bottom corner is where the output displacement is maximized along the direction shown. An output spring of 2 N mm^{-1} along the direction of desired displacement simulates the reaction force. For this problem, the modulus and thickness are 1,000 N mm^{-2} and 1 mm, the number of masks along the horizontal and vertical directions are 10 each, α_f (initial) $= \alpha_v = 1$, and upper bound on the volume fraction is 0.2. First, the example is solved with no continuation on the exponent α_f. That is, f_{cont} is set to 1. The scaling constant λ in (F2) is set to 10,000. Smaller values (e.g., 1, 10, 100 and 1,000) of λ were not adequate to obtain the positive desired displacement for this example. The permitted numbers of iterations and function evaluations are set to 100 and 400 respectively. The resultant inverter topology (without and with masks) is shown in Fig. 9a. As expected, there exist many gray cells. Next, when continuation is performed with $f_{cont} = 1.008$ (α_f is close to 24.2 after 400 evaluations), close to 0–1 solution is obtained (Fig. 9b) albeit a few gray cells are still present in the solution. Some disconnected



Case I: $f_{\mathrm{cont}} = 1.0116$

(a) $V^{\mathrm{o}} = 0.05$, $V^{*} = 0.27$

Case II: $f_{\mathrm{cont}} = 1.0151$

(b) $V^{\mathrm{o}} = 0.05$, $V^{*} = 0.26$

Case III: $f_{\mathrm{cont}} = 1.0172$

(c) $V^{\mathrm{o}} = 0.05$, $V^{*} = 0.26$

Case IV: $f_{\mathrm{cont}} = 1.0207$

(d) $V^{\mathrm{o}} = 0.05$, $V^{*} = 0.25$

Fig. 8 **a**–**d** Topologies for Problem 4. The continuation parameter is increased to obtain close to binary designs

regions also appear which can be considered as vestigial and can be ignored (see discussion). Figure 9c shows a compliant displacement inverter synthesized using 80 × 40 cell mesh with V^{o} reduced to 0.15 and λ increased to 50,000.

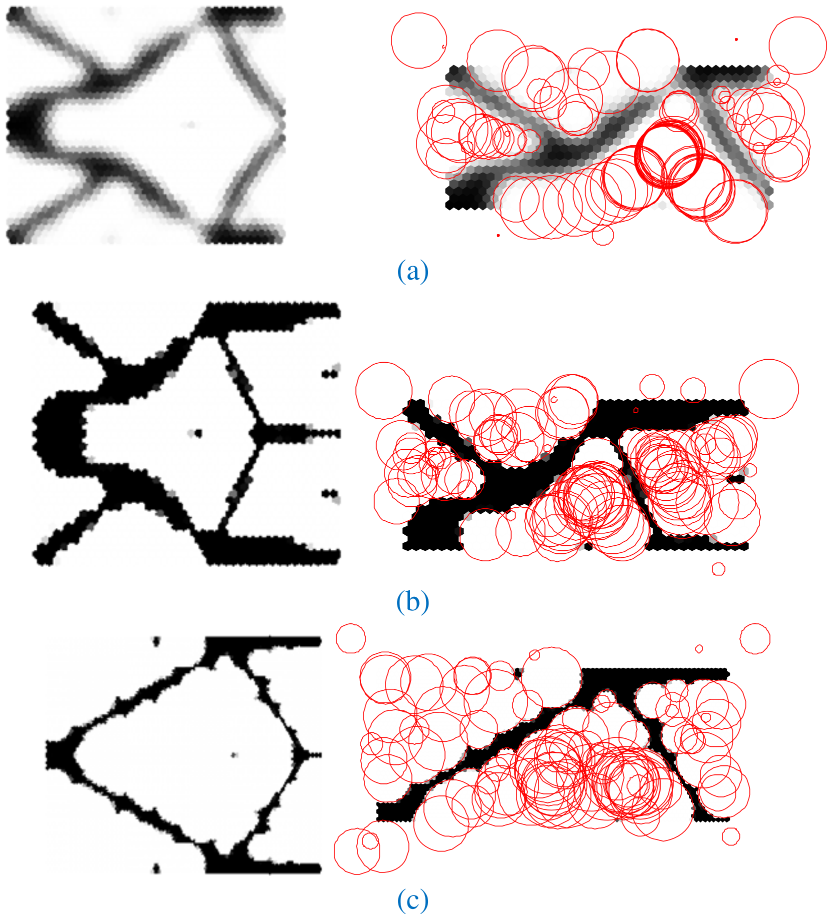
Finally, top symmetric halves of compliant crimpers are designed (Problem 6, Fig. 4) using the formulation in (F2) with three different meshes. The following parameters are used. Elastic modulus is 100 N mm^{-2}, thickness is 2 mm, number of masks along the horizontal and vertical are 15 and 10 respectively, the load **P** applied is 50 N, and the scaling constant λ in (F2) is 10,000. Maximum number of iterations and function evaluations are set to 100 and 200 respectively. Figure 10 shows full solutions for different mesh sizes, volume fractions and continuation parameters used. As observed, some gray cells do get retained in the topologies.

5 Discussion

This paper models topology optimization via negative masks to obtain continua using gradient search. No filtering (e.g., Sigmund 2007; Wang et al. 2010; Bourdin 2001) schemes are employed to explore the true nature of the model and to observe how different parameters in the formulation influence the solutions. Problems 1–4 pertain to four standard and different stiffness maximization examples.

5.1 Interpretation of solutions

In all solutions, checkerboards are not observed which is attributed to the honeycomb geometry used. In problem 1, optimal beam solutions are sought for different bounds on


(a)

(b)

(c)

Fig. 9 Compliant inverter topologies generated using **a** 40 × 20 cells without continuation, **b** 40 × 20 cells with continuation and **c** 80 × 40 cells with continuation. The *column on the left* shows full solutions while that on the *right* shows top symmetric halves with masks

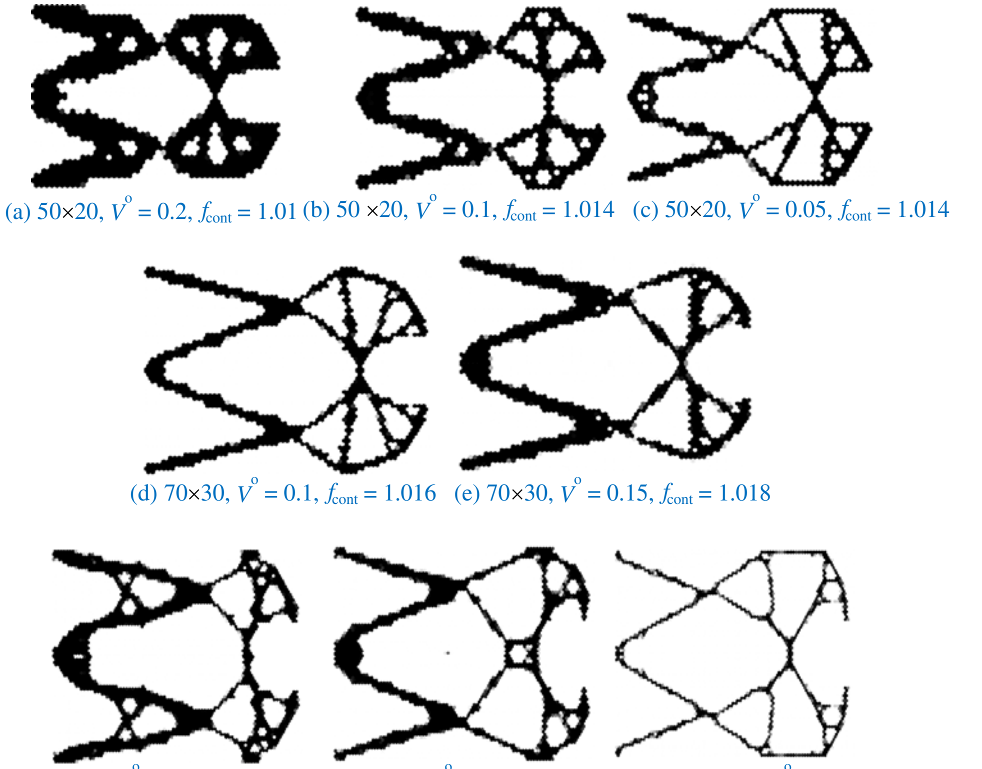
the volume fraction. It is observed that for coarse meshes, the discrepancy between the upper bound V^o on the volume fraction and the actual volume fraction V^* is more. Note that the continuum volume is computed using $\alpha_v = 1$ and not the final exponent α_f used to compute the strain energy to avoid undesired early convergence. Thus, this discrepancy is expected. A similar observation is made for solutions (a)–(c) in Fig. 10. The difference between the volume fractions decreases if finer grids are used for domain representation (Fig. 5a, c, e, f, g).

When comparing solutions (a), (e) and (f) in Fig. 5, it is observed that the three topologies are different. For the 80 × 40 cell mesh, the final volume fraction V^* is about 0.29. For the 50 × 25 cell and 100 × 50 cell meshes, the final volume fractions are respectively 0.41 and 0.25. Thus, even when these solutions are generated using the same initial upper bound on the volume fraction, they exist on different final constraint boundaries (those corresponding to α_f). This alludes to two aspects. (a) The upper bound V^o offers only indirect control on the continuum volume which becomes better with mesh refinement. (b) The final solutions lie on different constraint boundaries and are therefore different.

There is some similarity between solutions (c) and (e) in Fig. 5 with final volume fractions as 0.39 and 0.41. Solutions (a) and (g) are similar as well though the difference between their final volume fractions is more. However, solutions (a) and (d) with $V^* = 0.29$ and 0.28 respectively are not similar. Likewise, solutions (b) and (g) with final volume fractions as 0.34 and 0.36 are different in appearance. These observations allude to the third aspect that features in the topologies can change with the mesh size (note that relatively coarse meshes are used here). Solution (f) in Fig. 5 is a typical example which shows two additional diagonal bars, each of thickness comparable to the cell-size in the mesh. As mentioned earlier, such solutions are expected since no explicit filtering methods are used. Permitting slightly larger volume fraction (e.g., in solution g) does help eliminate these bars. Also, the member thicknesses improve.

Topologies in Fig. 6 suggest that solutions can vary with the user-specified number of masks. Here as well, the final solutions lie on different constraint boundaries even though all are generated using the same upper bound V^o. Thus, all reasons mentioned above hold true. In (Saxena 2010), via adaptively changing the number of masks, existence of

Fig. 10 **a**–**h** Compliant Crimpers designed using formulation (F2)



(a) 50×20, $V^{\text{o}} = 0.2$, $f_{\text{cont}} = 1.01$ (b) 50 ×20, $V^{\text{o}} = 0.1$, $f_{\text{cont}} = 1.014$ (c) 50×20, $V^{\text{o}} = 0.05$, $f_{\text{cont}} = 1.014$

(d) 70×30, $V^{\text{o}} = 0.1$, $f_{\text{cont}} = 1.016$ (e) 70×30, $V^{\text{o}} = 0.15$, $f_{\text{cont}} = 1.018$

(f) 90×40, $V^{\text{o}} = 0.2$, $f_{\text{cont}} = 1.01$ (g) 90×40, $V^{\text{o}} = 0.15$, $f_{\text{cont}} = 1.014$ (h) 90×40, $V^{\text{o}} = 0.05$, $f_{\text{cont}} = 1.012$

their optimal number for a continuum topology optimization problem is suggested. The observation herein is related to some extent with that in (Saxena 2010). In other words, if an optimal continuum is synthesized with a strict final resource constraint and if the number of masks is allowed to vary, the actual number of masks required will usually be different from those initially specified. The results in Fig. 7 suggest that the solutions can change with the variation in the upper bound on the mask radius. Those in Figs. 6 and 8 illustrate that close to binary solutions are possible if the continuation of the exponent α_{f} is handled carefully. Some disconnected islands are observed in Figs. 6, 7, 8, 9 which are addressed later in this section. All topologies have serrated boundaries, which are expected with coarse meshes due to the honeycomb geometry. Also, no explicit boundary smoothing step as in (Saxena 2010) is implemented. When observing compliant topologies, both long and short connections of thickness comparable to the cell size are observed. Also, gray cells exist in such solutions.

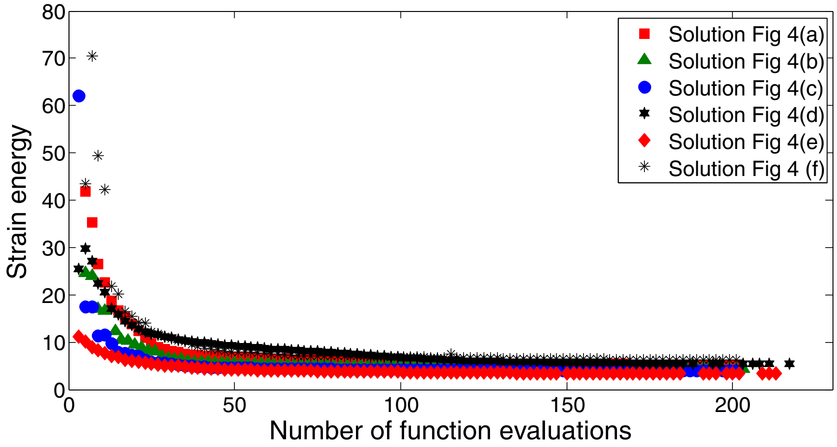
5.2 Efficiency: stochastic vs. gradient based search

Amongst the stochastic approaches that determine topologies by overlaying negative masks, the adaptive approach in (Saxena 2010) is the most efficient requiring about 2,000 function evaluations. Other approaches (e.g. Saxena 2009; Jain and Saxena 2010) that use genetic algorithm or successive searches require much more. To exemplify efficiency of the proposed approach, the convergence plots for the first six solutions in Fig. 5 are shown in Fig. 11. The plots suggest that for most cases, the strain energy is minimized significantly much earlier than 200 function evaluations. After 100 evaluations, strain energy is lowered gradually while the exponent α_{f} still increases so that close to binary solutions are obtained near the end of the search. This is about a factor of 10 reduction compared to the stochastic approach in (Saxena 2010). However, the topologies obtained therein are well connected and perfectly binary, the boundaries are smoother with moderated notches, and the number of masks is altered per iteration so that their adequate number is determined. In comparison, here, the solutions are close to binary with some gray (fictitious) cells still present and the number of masks remains the same as those specified by the user. But, there is a significant decrease in the number of evaluations.

5.3 Isolated islands

Some solutions in Figs. 6–9 contain regions that do not form a part of the respective main topologies. Investigation on whether such regions can be considered vestigial and can be removed is performed. The strain energy density (SED)

Fig. 11 Convergence histories for the first six topologies in Fig. 5

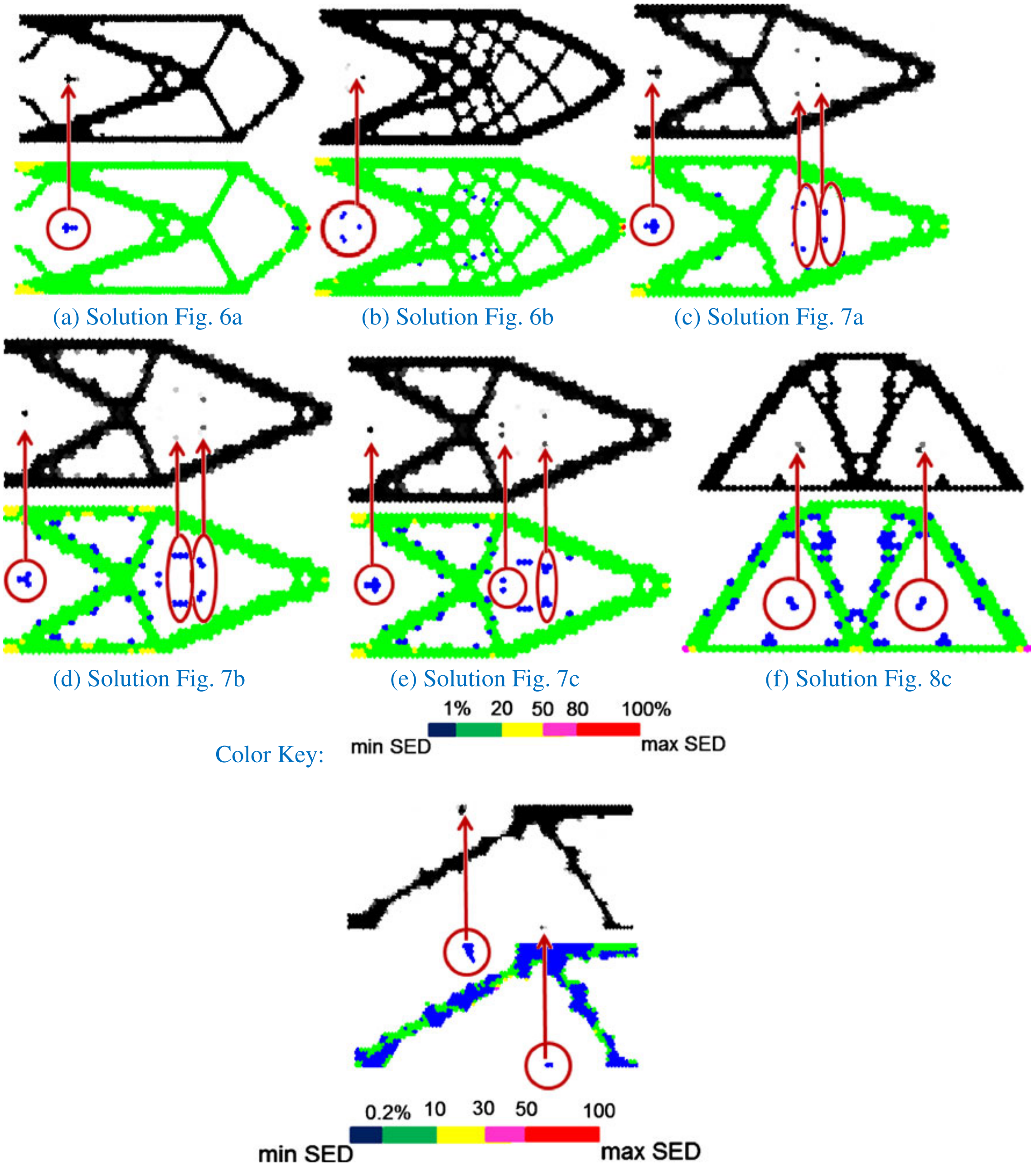


contours of these solutions are plotted in Fig. 12. For each case, regions in the SED contours are identified by circles with arrows pointing towards the corresponding dark disconnected hexagonal cells in the topologies. It is observed that such regions store 1% or less of the maximum strain energy density implying that they can be considered nonfunctioning and can be discarded. Note that these exist very close to the perimeters of a few masks and such regions attain the material densities close to 1 not because of their participation in the main continuum. Rather, they seem to be the artifacts of the high exponent value α_{f}. In fact, if the isolated cells are part of the main continuum, the objective (at least for stiff structures) can be improved depending on where in the topology the cells get placed. Since there are only a few islands in a solution, it is reckoned that the improvement will be marginal.

5.4 Numerical intricacies

The method presented here relies on proper choice of initial parameters and initial guess so that small deformation stiff structures and compliant continua can be designed. Care should be taken especially when choosing the initial values of the exponents. In the author's experience when solving symmetric stiff structure problems (Problems 1–4), specifying α_{f} (initial) and $\alpha_{\mathrm{v}} > 1$ has often led to the loss of symmetry in density and gradient values. A possible reason could be that numerical errors get introduced due to the near zero values of $\frac{\partial \rho_j}{\partial \psi_K}$ and hence the objective and constraints (Fig. 3). At any time during optimization, the exponent should not be overly large. This is again due to $\left|\frac{\partial \rho_j}{\partial \psi_K}\right|$ being very close to zero making it difficult for the optimization algorithm to proceed further. The scale factor λ in (F2) is a parameter on which the design of a compliant continuum can significantly depend on. At times, an undesired local minimum may result requiring the initial guess (initial position and sizes of the masks) to be altered. Based on experience, the author recommends the following values/ranges for different parameters. Significant improvements in the proposed algorithm are still possible which will be addressed in future endeavors. Interested readers and practitioners are invited to experiment with and modify/improve the MATLAB program (provided as supplementary material and in Appendix A; the interactive interface is accessed via mmos_main.m).The code is inspired by the previous educational articles on topology optimization that employ SIMP and level set methods (e.g., Sigmund 2001; Challis 2010; Andreassen et al. 2011).

1. Exponents $\alpha_{\mathrm{f}}/\alpha_v$ (*alp* in the code): A low initial value ($\alpha_{\mathrm{f}} = \alpha_{\mathrm{v}} = 1$ strictly recommended, or less) should be used. The author has not experimented much with $\alpha_{\mathrm{f}} = \alpha_{\mathrm{v}} < 1$ (e.g., 0.9). For a symmetric problem, if asymmetric mask movement/sizing is observed, optimization can be re-started with lower initial α_{f} and α_{v}. The continuation parameter f_{cont} may also be lowered. Alternatively, the problem can be solved by considering its symmetric half.
2. Continuation parameter f_{cont} (*fact* in the code): A value very close to but greater than 1 should be used. These values are mentioned with various examples solved in the paper. The parameter can be estimated as $f_{\mathrm{cont}} = \exp\left(\frac{1}{\max\_eval}\log\left[\alpha_f^{\mathrm{final}}\right]\right)$ where *max_eval* is the number of evaluations permitted and $\alpha_f^{\mathrm{final}}$ is the final value of the exponent. The above relation assumes that the initial α_{f} is 1. The author suggests that the number of evaluations allowed should be large enough so that increase in the exponent value is gradual. $\alpha_f^{\mathrm{final}}$ will depend on the optimization problem at hand, but use of a very high value is not recommended (e.g., see solutions to Problem 4).



(a) Solution Fig. 6a

(b) Solution Fig. 6b

(c) Solution Fig. 7a

(d) Solution Fig. 7b

(e) Solution Fig. 7c

(f) Solution Fig. 8c

(g) Solution Fig. 9(c)

Fig. 12 a–g Contour plots of the strain energy density (SED) for some solutions in Figs. 6, 7, 8, 9. The SEDs corresponding to the disconnected regions are marked with *circles*/*ellipses*

3. R_{max} (*max_R* in the code): Per Fig. 7, values of *max_R* in the range $[1/6W, W]$ where W is related to the minimum dimension of the rectangular region, has posed no difficulty in yielding solutions. This can however not be generalized.
4. V^o (*volf* in the code): For this parameter, a range of values between [0.05, 0.3] has been employed in the examples. As mentioned before, this parameter controls the continuum volume only indirectly which improves with mesh refinement. For a coarse mesh, one may try with V^o as low as 0.05 and expect the final volume fraction to be close to 0.3. For fine meshes, V^o will usually approximate the final volume fraction better. If desirable solutions are not obtainable for a low V^o, larger values may be tried.

5. λ (lines 28 and 29 in *analysis_mmos_eff.m*): The value of this parameter can significantly influence the solution of a compliant mechanism. λ is used to scale the magnitude of the output displacement with respect to the strain energy. For the problems solved herein, λ is set to 10,000 or 50,000. However, choice of $\lambda > 0$ is problem dependent.
6. *epsilon*: A low positive value (e.g., 10^{-3} or 10^{-4}) is usually employed as the lower bound for cell densities.
7. *adj_x* and *adj_y*: The author has set these values as zero for the examples. They are used to respectively control the horizontal and vertical positions of the entire group of masks in the initial guess.

6 Closure

Topology optimization with negative masks is implemented in this paper with gradient search. It was previously shown using the honeycomb parameterization that well connected perfectly binary solutions could be attained with the stochastic search, but the approach required a large number of design evaluations. The computational effort required herein is about 1/10th compared to its predecessor schemes that use the random mutation based hill climber procedure. Close to binary solutions are possible if the increase in the exponent of the logistic function is gradual. Some fictitious cells can appear at continuum boundaries and at sites where the local critical feature size is comparable to the characteristic size of the mesh. It is conjectured that a hybrid method combining both, gradient and stochastic searches will help obtain well connected, perfectly binary topologies efficiently. This, in particular, will be useful when designing large deformation continua (e.g., Bruns and Tortorelli 2001; Pedersen et al. 2001; Saxena and Ananthasuresh 2001; Rai et al. 2007, 2009; Mankame and Ananthasuresh 2007; Reddy et al. 2010) where computation of the search directions will be possible for solutions with convergent nonlinear displacement analyses. For a non-convergent one, stochastic mutation steps will help alter that intermediate design to a neighboring convergent one which can be used as an initial guess for the subsequent gradient search. Extension of the Material Mask Overlay Method to three-dimensional continuum design is straight forward by replacing the circular masks with the spherical ones and hexagonal cells with tetrakaidecahedrons.

Acknowledgments The author acknowledges the support from the Alexander von Humboldt Stiftung, Germany and the Institute of Kinematics and Machine Dynamics, IGM, 52062 RWTH Aachen, Germany. He also acknowledges the anonymous reviewer for his valuable comments that have helped improve this manuscript.

Appendix A: The matlab code

A.1 The main call

```
1.  function [mask_new] = mmos81(nd,eh,F1,F2,DBC,E,th,Nx,Ny,ep,alp,mR,vf,oc,adx,ady,fact,fig)
2.  % developed by Anupam Saxena for Academic use, 12/3/2010
3.  for i = 1:size(eh,1), ct(i,:) = sum(nd(eh(i,[2:7]),[2,3]),1)/6; end, % hex centroids
4.  [X, Y] = meshgrid([min(ct(:,1))+adx + ([0:Nx-1]/(Nx−1))*(max(ct(:,1)) − min(ct(:,1)))], [min(ct(:,2))+ady+([0:Ny–1]/(Ny-1))* (max(ct(:,2))−min(ct(:,2)))]);
5.  mask = [reshape(X,Nx*Ny,1)′ reshape(Y,Nx*Ny,1)′ 2*ones(1,Nx*Ny)]; % [X Y R] mask variable initialization
6.  low = [(min(ct(:,1))−mR)*ones(1,Nx*Ny) (min(ct(:,2))−mR)*ones(1,Nx*Ny) (0)*ones(1,Nx*Ny)]; % lower bound
7.  upp = [(max(ct(:,1))+mR)*ones(1,Nx*Ny)(max(ct(:,2))+mR)* ones(1,Nx*Ny) (mR)*ones(1,Nx*Ny)]; % upper bound
8.  options = optimset(‘Display’,‘iter’,‘MaxIter’,100, ‘MaxFunEvals’,200,‘TolX’,1e-10,‘TolFun’,1e−10, ‘GradObj’,’on’,‘GradConstr’,‘on’,‘DerivativeCheck’, ‘off’);
9.  save(‘alp1’,‘alp’);
10. mask_new = fmincon(@(xx)analysis_mmos_eff(xx, nd,eh,F1,F2,DBC,E,th,ct,Nx,Ny,alp,ep,oc,fact,fig), mask,[],[],[],[],low,upp,@(xx)vol_cons_mmos(xx,alp, eh,ct,Nx,Ny,ep,vf*size(eh,1),[],[]),options);
11. load alp1, [cell_dens,sens_mask] = sensitivity_mask_mmos(mask_new,eh,ct,alp,Nx,Ny,ep); % final densities
12. plot_masks_and_densities_mmos(nd,eh,cell_dens, mask_new,ct,Nx,Ny,fig); % final solution plot
```

A.2 The analysis function

```
13. function [obj,obj_sens,dv] = analysis_mmos_eff(mk, nd,eh,F1,F2,DBC,E,th,cent,Nx,Ny,alp,ep,oc,fact,fig), % analysis function
14. load alp1.mat, [dens,sens] = sensitivity_mask_mmos (mk,eh,cent,alp,Nx,Ny,ep); % cell densities, derivatives w.r.t mask parameters
15. if (alp < 60), alp = fact*alp; end, % continuation on exponent
16. KE = element_stiffness_mmos(E,th); % element stiffness matrix, same for all cells
17. nn = eh([1:size(eh,1)],[2:7]); % connectivity information
```

```
ldof(:,2*[1:6]−1) = 2*nn(:,[1:6])−1; % local degrees of freedom
ldof(:,2*[1:6]) = 2*nn(:,[1:6]);
IJX = zeros(144*size(eh,1),3); % Triplets for Global Stiffness assembly
ntrip = 0;
for n = 1:size(eh,1), % assembly of the Global Stiffness
forkr=1:12
forkc=1:12
ntrip = ntrip + 1;
IJX(ntrip,:) = [ldof(n,kr) ldof(n,kc) dens(n,1)*KE(kr,kc)];
end
end
end
KG = sparse(IJX(:,1),IJX(:,2),IJX(:,3),2*size(nd,1),2*size(nd,1)); % Global Stiffness in sparse form
FF12 = sparse(2*size(nd,1),4); %[Input Force F, Virtual Force Fd, displacements U, displacements V] in columns
FF12( 2*(F1(:,2) − 1) + F1(:,3),1) = F1(:,4); % input force vector F
FF12( 2*(F2(:,2) − 1) + F2(:,3),2) = F2(:,4); % virtual force vector Fd
FixDOF = 2*(DBC(:,2) −1) + DBC(:,3); % Fixed displacement IDs
KG(2*(F2(:,2)-1)+F2(:,3),2*(F2(:,2)-1)+F2(:,3))= KG(2*(F2(:,2)-1)+F2(:,3),2*(F2(:,2)-1)+F2(:,3))+ diag(F2(:,5)); % output spring
FF12(setdiff([1:2*size(nd,1)],FixDOF),[3:2+oc])= KG(setdiff([1:2*size(nd,1)],FixDOF),setdiff([1:2*size(nd,1)],FixDOF)) \ FF12(setdiff([1:2*size(nd,1)],FixDOF),[1:oc]);
dv(:,1) = −0.5*sum(reshape(FF12(ldof,3),size(eh,1),12)*KE.*reshape(FF12(ldof,3),size(eh,1),12),2); % derivatives of strain energy w.r.t densities
dv(:,2) = −sum(reshape(FF12(ldof,4),size(eh,1),12)*KE.*reshape(FF12(ldof,3),size(eh,1),12),2); % derivatives of the output displacements w.r.t densities
SE_MSE = [0.5*FF12(:,3)′*FF12(:,1) FF12(:,4)′*FF12(:,1)]; % Strain energy and output displacement
obj1 = [SE_MSE(1) −10000*SE_MSE(2)/SE_MSE(1)]; % SE, −Lambda*MSE/SE
obj_sens1=sens′*[dv(:,1) −10000*(dv(:,2)/SE_MSE(1) − SE_MSE(2)*dv(:,1)/SE_MSE(1)ˆ2)]; % obj. derivatives w.r.t mask parameters
obj = full(obj1(oc)); % objective being minimized
obj_sens = obj_sens1(:,oc); % sensitivities of objective being minimized
save(‘alp1’,‘alp’), save(‘mask_prev’,‘mk’), plot_masks_and_densities_mmos(nd,eh,dens,mk,cent,Nx,Ny,fig); pause(1e-6), % intermediate solution plot
```

A.3 Computation of cell densities and their gradients w.r.t mask parameters

```
function [dens,sens] = sensitivity_mask_mmos(mk,eh,ct,alp,Nx,Ny,ep)
for i = 1:size(eh,1), % computation of cell densities
dens(i,1) = ep+prod((1+exp(-alp*(sqrt((ct(i,1)-mk([1:Nx*Ny])).ˆ2+(ct(i,2)-mk([Nx*Ny+1:2*Nx*Ny])).ˆ2 )-mk([2*Nx*Ny+1:3*Nx*Ny]))))).ˆ(-1));
end
dens(find(dens>1)) = 1;
for j = 1:Nx*Ny, % computation of density derivatives w.r.t mask parameters
pp = [mk(j) mk(j+Nx*Ny) mk(j+2*Nx*Ny)];
d1 = sqrt((pp(1)-ct(:,1)).ˆ2 + (pp(2)-ct(:,2)).ˆ2) + 0.001;
sens(:,j) = (alp*dens(:,1).*exp(−alp*(d1-pp(3))).*(pp(1)-ct(:,1))./d1)./(1+exp(−alp*(d1-pp(3)))); % sensitivity w.r.t X_k
sens(:,j+Nx*Ny) = (alp*dens(:,1).*exp(−alp*(d1-pp(3))).*(pp(2)-ct(:,2))./d1)./(1 + exp(-alp*(d1-pp(3)))); % sensitivity w.r.t Y_k
sens(:,j+2*Nx*Ny) = −(alp*dens(:,1).*exp(-alp*(d1-pp(3))))./(1+exp(−alp*(d1-pp(3)))); % sensitivity w.r.t R_k
end
```

A.4 Element stiffness matrix

```
function [K] = element_stiffness_mmos(E,thick), % element stiffness matrix for nu = 0.29
K = E*thick*(1/1000)*[616.43012 92.77147 −168.07333 65.54377 −232.28511 −0.00032 −120.65312 −83.28564 −71.60020 −92.77115 −23.81836 17.74187;
92.77147 509.30685 101.02751 −71.90335 0.00032 −18.03857 −83.28564 −24.48314 −92.77179 −178.72347 −17.74187 −216.15832;
−168.07333 101.02751 455.74522 0.00000 −168.07333 −101.02751 −71.60020 −92.77179 23.60185 −0.00000 −71.60020 92.77179;
65.54377 −71.90335 0.00000 669.99176 −65.54377 −71.90335 −92.77115 −178.72347 −0.00000 −168.73811 92.77115 −178.72347;
−232.28511 0.00032 −168.07333 −65.54377 616.43012 −92.77147 −23.81836 −17.74187 −71.60020 92.77115 −120.65312 83.28564;
−0.00032 −18.03857 −101.02751 −71.90335 −92.77147 509.30685 17.74187 −216.15832 92.77179 −178.72347 83.28564 −24.48314;
−120.65312 −83.28564 −71.60020 −92.77115 −23.81836 17.74187 616.43012 92.77147 −168.07333 65.54377 −232.28511 −0.00032;
−83.28564 −24.48314 −92.77179 −178.72347 −17.74187 −216.15832 92.77147 509.30685 101.02751 −71.90335 0.00032 −18.03857;
−71.60020 −92.77179 23.60185 −0.00000 −71.60020 92.77179 −168.07333 101.02751 455.74522 0.00000 −168.07333 −101.02751;
−92.77115 −178.72347 −0.00000 −168.73811 92.77115 −178.72347 65.54377 −71.90335 0.00000 669.99176 −65.54377 −71.90335;
−23.81836 −17.74187 −71.60020 92.77115 −120.65312 83.28564 −232.28511 0.00032 −168.07333 −65.54377 616.43012 −92.77147;
17.74187 −216.15832 92.77179 −178.72347 83.28564 −24.48314 −0.00032 −18.03857 −101.02751 −71.90335 −92.77147 509.30685;];
```

A.5 Volume constraint and its gradients

```
function [g,d,cons_sens,d1] = vol_cons_mmos(mk,alp,eh,ct,Nx,Ny,ep,vstar,d,d1), % volume constraint and derivatives
[cell_dens,sens_mask] = sensitivity_mask_mmos(mk,eh,ct,alp,Nx,Ny,ep);, % cell densities and derivatives
g = sum(cell_dens)- vstar; % volume constraint
cons_sens = sum(sens_mask,1)′; % derivatives of g
```

A.6 Plot function for intermediate and final solutions

```
function []= plot_masks_and_densities_mmos(nd,eh,cd,mk,ct,Nx,Ny,basefig)
figure(basefig), cla, hold on; axis equal
for i = 1:size(eh,1), % plot densities
fill([nd(eh(i,[2:7]),2)],[nd(eh(i,[2:7]),3)],[(1-cd(i,1)) (1-cd(i,1)) (1-cd(i,1))],‘EdgeColor’,[(1-cd(i,1)) (1-cd(i,1)) (1-cd(i,1))]);
end
for j = 1:Nx*Ny,
plot([mk(j)+mk(2*Nx*Ny+j)*cos(0:pi/24:2*pi)],[mk(Nx*Ny+j)+mk(2*Nx*Ny+j)*sin(0:pi/24:2*pi)],‘r’);
end
```

A.7 Function incorporating multiple load cases

```
function [obj,obj_sens,dv] = analysis_mmos_multiload_eff(mk,nd,eh,F1,F2,DBC,E,th,ct,Nx,Ny,alp,ep,oc,fact,fig)
load alp1.mat[dens,sens] = sensitivity_mask_mmos(mk,eh,ct,alp,Nx,Ny,ep);
if (alp < 60), alp = fact*alp; end, % continuation on exponent
KE = element_stiffness_mmos(E,th); % element stiffness matrix, same for all cells
nn = eh([1:size(eh,1)],[2:7]); % connectivity information
ldof(:,2*[1:6]-1) = 2*nn(:,[1:6])-1; % local degrees of freedom
ldof(:,2*[1:6]) = 2*nn(:,[1:6]);
IJX = zeros(144*size(eh,1),3); % Triplets for Global Stiffness assembly
ntrip = 0;
for n = 1:size(eh,1), % assembly of the Global Stiffness
forkr=1:12
forkc=1:12
ntrip = ntrip + 1;
IJX(ntrip,:) = [ldof(n,kr) ldof(n,kc) dens(n,1)*KE(kr,kc)];
end
end
end
KG = sparse(IJX(:,1),IJX(:,2),IJX(:,3),2*size(nd,1),2*size(nd,1)); % Global Stiffness in sparse form
FF12 = sparse(2*size(nd,1),2*size(F1,1)); % input vector containing multiple loads
for i = 1:size(F1,1), FF12(2*(F1(i,2) − 1) + F1(i,3),i) = F1(i,4); end, % individual forces in different columns
FixDOF = 2*(DBC(:,2) − 1) + DBC(:,3); % Fixed displacement IDs
FF12(setdiff([1:2*size(nd,1)],FixDOF),[1+size(F1,1):2*size(F1,1)])=KG(setdiff([1:2*size(nd,1)],FixDOF),setdiff([1:2*size(nd,1)],FixDOF))\FF12(setdiff([1:2*size(nd,1)],FixDOF),[1:size(F1,1)]);
for i = 1:size(F1,1), dv(:,i) = -0.5*sum(reshape(FF12(ldof,i+size(F1,1)),size(eh,1),12)*KE.*reshape(FF12(ldof,i+size(F1,1)),size(eh,1),12),2); end
for i = 1:size(F1,1), SE(i) = 0.5*FF12(:,i+size(F1,1))′*FF12(:,i); end, % individual strain energies and their derivatives (above)
obj = full(sum(SE)); % objective: net strain energy
obj_sens = sens′*(sum(dv,2)); % derivatives of obj w.r.t mask parameters
save alp1alpobj, save mask_savemk; plot_masks_and_densities_mmos(nd,eh,dens,mk,ct,Nx,Ny,fig); pause(1e-6)
```

Appendix B: Execution and explanation of the code

B.1 Quick execution

(a) Type *mmos_main.m* at the Matlab prompt.

(b) Click on any Load Problem button (top six buttons on the interface). These problems are similar to the respective problems in the paper. To solve Problems III and IV (multiple load problems), *analysis_mmos_eff* in line 10 of *mmos81*.m should be replaced by *analysis_mmos_multiload_eff*. For the rest, *analysis_mmos_eff* should be retained.

(c) Click on the MMOS81 button to observe the solutions evolve.

B.2 Explanation

The Matlab implementation used is composed of 6 primary functions. The main call is:

$$p = mmos81(node, elem\_hexx, F1, F2, DBC, E, t, Nx, Ny, epsi, alp, max\_R, volf, obj\_cho, adj\_x, adj\_y, fact, basefig);$$

All parameters of this function are generated/specified interactively. The function itself is accessed from within the user interface (provided with the supplementary material; execute mmos_main.m to access the interface) though this call can be made separately at the Matlab prompt after the design domain, loading and boundary conditions, i.e., arrays *node, elem_hexx*, $F1$ (*forces*1), $F2$(*forces*2), and *DBC*(*dispbc1*) are generated (steps I–VI; Click on the respective help buttons). A user can independently specify the elastic modulus (E), thickness (t), initial number of masks along the horizontal and vertical (*Nx* and *Ny*), minimum cell density value (*epsi*), initial value of the exponent (*alp*), maximum mask radius (*max_R*; note that the minimum radius is zero), volume fraction (*volf*), *obj* ('1' for minimizing the strain energy in (F1) and '2' to design a compliant mechanism using the flexibility-stiffness formulation in (F2)), initial horizontal and vertical mask adjustments (*adj_x* and *adj_y*; set both as zero if no adjustments in the initial guess is required), the factor (*fact*) by which the exponent (*alp*) is increased per function evaluation (set *fact* as close as possible to 1 but slightly larger, say 1.014; *fact* can also be computed if the final target value of α and the number of iterations/evaluations to be performed are approximately known; see discussion), and finally the figure number (*basefig*) in which the plots are desired. Variable *p*contains the final mask information.

B.3 The main program (lines 1–12)

The centroids *ct*of the hexagonal cells are computed in line 3. In line 4, evenly spaced array of *Nx* × *Ny* points are generated using the *meshgrid* function in Matlab. These form the mask centers in line 5. Each mask is initialized with radius 2 units. This value can be changed if a different initial guess is to be specified. Line 5 also reveals how the design variable vector is formed with first, all *Nx* × *Ny* horizontal coordinates followed by the same number of the vertical coordinates of the masks. The last *Nx* × *Ny* entries correspond to the mask radii. Lines 6 and 7 specify the lower and upper bounds on the mask variables. Masks can be placed anywhere within the rectangle of size (L + *2max_R*) × (W + *2max_R*) where L and W are the dimensions of the rectangular region within which the domain is placed. Note that the domain itself need not be rectangular in shape. Line 8 specifies some options for the *fmincon* function (type optimset at the Matlab prompt for details). Line 9 saves the initial, user specified value of the exponent α to be accessed by the analysis function (*analysis_mmos_eff.m*or *analysis_mmos_multi_load_eff.m*) of the code. A call to the Matlab's optimization routine *fmincon* is made in line 10. The routine requires the objective and constraint information to be given via two separate functions (Appendices B.4 and B.5). The mask positions and sizes are stored in the *mask_new* array. In line 11, the final value of the exponent is retrieved and cell densities are computed. They are plotted in line 12.

B.4 Objective and its gradients (lines 13–44)

The current value of the exponent is retrieved (line 14) based on which the cell densities are computed. The exponent is increased by the factor *fact* in line 15 if α is less than its maximum value (e.g., 60, which can be changed if required). The element stiffness matrix, *KE* is accessed in line 16.The connectivity information from the *elem_hexx array* (*eh*) is accessed in line 17. Local degrees of freedom are computed only once in lines 18–19. For efficient assembly of the stiffness matrix, the notion is to assemble it in triplets I, J and X within a *for* loop and then transfer this information to a sparse matrix at the end of the loop. With the mesh size known, sizes of I, J and X are known beforehand so that they can be initialized before the assembly loop (line 20). Assembly is performed in lines 21–29 and the global stiffness matrix *KG* is stored as a sparse matrix in line 30.In line 31, matrix $FF12$ of size $2Nnode \times 4$ is initialized. The matrix contains four vectors—the first is the input force vector (**F** s.t. $\mathbf{F} = \mathbf{KU}$), the second is the vector containing the dummy load ($\mathbf{F}_d = \mathbf{KV}$) to compute the output displacement, and the third and fourth vectors **U** and **V** are the displacement responses due to the first two load vectors. In lines 32–33, the first two columns of $FF12$ are updated with information from the input load array *forces1* $\equiv F1$ and the output displacement array *forces*2 $\equiv F2$. Fixed degrees of freedom are determined in line 34 via the *dispbc*1 $\equiv DBC$ array. The output spring(s) is (are) added to the global stiffness matrix in line 35. The displacement vectors **U** and **V**, i.e., the third and fourth columns of $FF12$ are solved for in line 36. Note that if only strain energy is minimized ($obj\_cho = 1$), only one displacement vector **U** is computed. Otherwise, for $obj\_cho = 2$ which is the switch to design compliant mechanisms, both **U** and **V** are determined. Gradients of the strain energy and output deformation for each cell are computed in lines 37–38. These lines compute the gradients directly which are stored in the array *dv*. The first column in *dv* stores the sensitivities pertaining to the strain energy while the second column stores the same for the output deformation. For $obj\_cho =$ 1, the second column in *dv* is not needed (the entries are all zeros). Line 39 stores the strain energy and output displacement in vector *SE_MSE* as the first and second entries. The two objectives in (F1) and (F2) are stored in line 40. The scale factor λ in (F2) appears here which should be adjusted if required. In line 41, sensitivities of both objectives with respect to the mask parameters are obtained. Lines 42 and 43 release the function value and its gradients as the first two output variables. In line 44, the current value of the exponent is saved to be accessed and increased (lines 14–15) by *fact* the next time a call is made to this function by *fmincon*. The current mask parameters are also saved. The intermediate solution, positions and sizes of the masks are plotted.

B.5 Partial derivatives of cell densities w.r.t mask parameters (lines 45–56)

Lines 45–56 compute the cell densities and its derivatives with respect to the mask variables. Lines 46–48 compute the cell densities in vectorized form (a *for* loop is absorbed within line 47). Line 49 imposes the upper limit of 1 on the cell densities. The lower limit is already imposed via line 47 through *ep* (*epsilon*). Lines 50 through 56 compute the derivatives of the cell densities with respect to mask parameters. Array *sens* in lines 53–55 stores these derivatives. In the first $Nx \times Ny$ columns, derivatives are stored with respect to the horizontal mask coordinates. The next $Nx \times Ny$ columns contain derivatives with respect to the y mask coordinates, and the last same number of columns store gradients with respect to their radii.

B.6 Element stiffness matrix (lines 57–69)

The element stiffness matrix is computed separately using the unit Wachspress hexagonal finite element. 19 integration points are used with modulus and thickness as 1 each. Poisson's ratio is taken as 0.29. The resultant constant matrix of size 12 × 12 is stored in these lines. The procedure for evaluation of the stiffness matrix is documented in (Saxena 2010; Talischi et al. 2009). User specified values of the elastic modulus (E) and continuum thickness (t) are multiplied to this matrix. Note that K is symmetric which can be used to reduce information storage (and make stiffness assembly more efficient).

B.7 Volume constraint and its gradients (lines 70–73)

The *fmincon.m* routine in Matlab requires the constraints to be specified using a separate function. Both linear and nonlinear constraints can be specified. Linear constraints are not used and so the corresponding values and coefficients are passed in and out as empty matrices d and $d1$ (Correlate the arguments of this function with line 10 of the code). Cell densities and sensitivities are computed again. But here, the initial, user specified value of the exponent is used. That is, the exponent is not updated as in line 15. The volume constraint is registered in line 72 and its gradients are computed in Line 73. Since $\frac{\partial V}{\partial \rho_i}=1 \forall i$, $\frac{\partial V}{\partial \psi_j}$ is the sum over all rows in a column that corresponds to the ψ_jth mask variable.

B.8 Plot of intermediate and final results (74–81)

Intermediate cell densities are plotted via lines 76–78 while the mask positions and their sizes are plotted through lines 79–81.

B.9 Function incorporating multiple load cases

All loads may be input through *forces*1 array (to be edited directly within the MATLAB's workspace), and each line will correspond to a different load case. A separate function *analysis_mmos_multiload_eff.m* Appendix A.7 is available with changes to incorporate computation of strain energy and its derivatives for different load cases. To determine stiff structures for multiple loads (e.g., Problems 3 and 4), line 10 of the code should be modified by replacing *analysis_mmos_eff* with *analysis_mmos_multiload_eff*.

References

Andreassen E, Clausen A, Schevenels M, Lazarov BS, Sigmund O (2011) Efficient topology optimization in MATLAB using 88 lines of code. Struct Multidisc Optim 43:1–16. doi:10.1007/s00158-010-0594-7

Bendsøe MP (1995) Optimization of structural topology, shape and material. Springer, Berlin

Bendsøe D, Kikuchi D (1988) Generating optimal topologies in structural design using a homogenization method. Comput Methods Appl Mech Eng 71(2):197–224

Bourdin B (2001) Filters in topology optimization. Int J Numer Methods Eng 50(9):2143–2158

Bruns TE, Tortorelli DA (2001) Topology optimization of nonlinear elastic structures and compliant mechanisms. Comput Methods Appl Mech Eng 190:3443–3459

Challis VJ (2010) A discrete level-set topology optimization code written in Matlab. Struct Multidisc Optim 41:453–464

Epelman MA (2010) Sequential Quadratic Programming (SQP). Lecture notes on IOE 511/Math 562 at http://www-personal.umich.edu/∼mepelman/teaching/IOE511/511notes.pdf. Accessed 9 Dec 2010

Guest J (2009) Topology optimization with multiple phase projection. Comput Methods Appl Mech Eng 199(1–4):123–135. doi:10.1016/j.cma.2009.09.023

Guest JK, Prévost JH, Belytschko T (2004) Achieving minimum length scale in topology optimization using nodal design variables and projection functions. Int J Numer Methods Eng 61:238–254

Hull P, Canfield S (2006) Optimal synthesis of compliant mechanisms using subdivision and commercial FEA. ASME J Mech Des 128:337–348

Jain C, Saxena A (2010) An improved material-mask overlay strategy for topology optimization of structures and compliant mechanisms. ASME J Mech Des 132(6):1–10. doi:10.1115/1.4001530

Jakiela MJ, Chapman C, Duda J, Adewuya A, Saitou K (2000) Continuum structural topology design with genetic algorithms. Comput Methods Appl Mech Eng 186:339–356

Langelaar M (2007) The use of convex uniform honeycomb tessellations in structural topology optimization. In: Proceedings of the 7th world congress on structural and multidisciplinary optimization. Seoul, South Korea, 21–25 May 2007

Luo JZ, Luo Z, Chen SK, Tong LY, Wang MY (2008) A new level set method for systematic design of hinge-free compliant mechanisms. Comput Methods Appl Mech Eng 198:318–331

Mankame N, Ananthasuresh GK (2007) Synthesis of contact-aided compliant mechanisms for non-smooth path generation. Int J Numer Methods Eng 69(12):2564–2605

Pedersen CBW, Buhl T, Sigmund O (2001) Topology synthesis of large-displacement compliant mechanisms. Int J Numer Methods Eng 50:2683–2705

Poulsen TA (2003) A new scheme for imposing minimum length scale in topology optimization. Int J Numer Methods Eng 57:741–760

Rahmatallah SF, Swan CC (2004) A Q4/Q4 continuum structural topology optimization implementation. Struct Multidisc Optim 27:130–135

Rai AK, Saxena A, Mankame ND (2007) Synthesis of path generating compliant mechanisms using initially curved frame elements. ASME J Mech Des 129:1056–1063

Rai AK, Saxena A, Mankame ND (2009) Unified synthesis of compact planar path-generating linkages with rigid and deformable members. Struct Multidisc Optim 41:863–879. doi:10.1007/s00158-009-0458-1

Reddy BVS, Nagendra, Saxena A (2010) Automated design of contact-aided compliant mechanisms using initially curved frame elements. ASME Design Engineering and Technical Conferences, Montre'al, Canada, 15–18 Aug 2009, #DETC-29172

Saxena A (2009) A material-mask overlay strategy for continuum topology optimization of compliant mechanisms using honeycomb discretization. ASME J Mech Des 130(8):1–9. doi:10.1115/1.2936891

Saxena A (2010) On an adaptive mask overlay topology synthesis method. ASME Design Engineering and Technical Conferences, Montre'al, Canada, 15–18 Aug 2009, DETC2010-29113

Saxena A, Ananthasuresh GK (2000) On an optimality property of compliant topologies. Struct Multidisc Optim 19: 36–49

Saxena A, Ananthasuresh GK (2001) Topology synthesis of compliant mechanisms for nonlinear force-deflection and curved path specification. Trans ASME-R J Mech Des 123(1):33–42

Saxena R, Saxena A (2003) On honeycomb parameterization for topology optimization of compliant mechanisms. ASME Design Engineering Technical Conferences, Design Automation Conference, Chicago, IL, 2–6 Sept 2003, paper #. DETC2002/DAC-48806

Saxena R, Saxena A (2007) On honeycomb representation and SIGMOID material assignment in optimal topology synthesis of compliant mechanisms. Finite Elem Anal Des 43(14): 1082–1098

Sethian JA, Wiegmann A (2000) Structural boundary via level set and immersed interface methods. J Comput Phys 163(2):489–528

Sigmund O (2001) A 99 line topology optimization code written in Matlab. Struct Multidisc Optim 21:120–127

Sigmund O (2007) Morphology-based black and white filters for topology optimization. Struct Multidisc Optim 33:401–424

Talischi C, Paulino GH, Le CH (2009) Honeycomb Wachspress finite elements for structural topology optimization. Struct Multidisc Optim 37(6):569–583

Wang MY, Chen SK, Wang XM, Mei YL (2005) Design of multimaterial compliant mechanisms using level set methods. ASME J Mech Des 127:941–956

Wang F, Lazarov BS, Sigmund O (2010) On projection methods, convergence and robust formulations in topology optimitation. Struct Multidisc Optim. doi:10.1007/s00158-010-0602-y

Xu S, Cai Y, Cheng G (2010) Volume preserving nonlinear density filter based on heaviside funtions. Struct Multidisc Optim 41:495–505

Yin L, Ananthasuresh GK (2001) Topology optimization of compliant mechanisms with multiple materials using a peak function material interpolation scheme. Struct Multidisc Optim 23:49–62

Yin L, Ananthasuresh GK (2003) Design of distributed compliant mechanisms. Mech Based Des Struct Mach 31(2):151–179

Yoon G, Kim Y, Bendsoe M, Sigmund O (2004) Hinge – free topology optimization with embedded translation – invariant differentiable wavelet shrinkage. Struct Multidisc Optim 27:139–150