ALTERNATING ACTIVE-PHASE ALGORITHM FOR MULTIMATERIAL TOPOLOGY OPTIMIZATION PROBLEMS A 115-LINE MATLAB IMPLEMENTATION

R. TAVAKOLI AND S.M. MOHSENI

Abstract. A new algorithm for the solution of multimaterial topology optimization problems is introduced in the present study. The presented method is based on the splitting of a multiphase topology optimization problem into a series of binary phase topology optimization sub-problems which are solved partially, in a sequential manner, using a traditional binary phase topology optimization solver; internal solver. The coupling between these incomplete solutions is ensured using an outer iteration strategy based on the block coordinate descend method. The presented algorithm provides a general framework to extend the traditional binary phase topology optimization solvers for the solution of multiphase topology optimization problems. The overall algorithmic complexity of the presented algorithm is independent of the number of desired phases, $\mathtt{p}$, and its computational cost is approximately proportional to $\mathtt{p}^2$. The interesting features of the presented algorithm are: generality, simplicity, efficiency, ease of implementation and the inheritance of the convergence properties of its internal optimization solver. The presented algorithm is used to solve multimaterial minimum structural and thermal compliance topology optimization problems based on the classical optimality criteria method. The details of MATLAB implementation are presented and the complete program listings are provided as the supplementary materials. The success and performance of the presented method are demonstrated through several two dimensional numerical examples.

Keywords. Coordinate descent method; MATLAB code; Multiphase topology optimization; Optimality criteria.

Contents

Date: April 15, 2013.

R. Tavakoli (corresponding author: rtavakoli@sharif.ir, faculty member), S.M. Mohseni (undergraduate student), Materials Science and Engineering Department, Sharif University of Technology, Tehran, Iran, P.O. Box 11365-9466, Tel: 982166165209, Fax: 982166005717.

1. Introduction

A topology optimization problem is to determine the unknown material(s) distribution within a pre-defined spatial domain, such that some optimality conditions hold. After the seminal paper by Bendsøe and Kikuchi [1], the structural topology optimization (c.f. [2, 3]) has been received significant attention within the computational mechanics and civil engineering communities. It is currently known as a standard method for design of engineering structures at micro and macro scales under simple and complex service conditions. An interesting question in this context is that whether it is possible to use multiple materials under volume or weight constraint to find a better design in terms of the structure's performance and/or manufacturing cost. This is particularly attractive in the case of multiphysics problems and/or multifunctional structures. However, the majority of researches in this field are limited to binary-phase (materials-void) topology optimization problems. To the best of our knowledge there are a few works focused on the solution of multiphase topology optimization problems. In the remainder of this section the literature of multimaterial topology optimization will be reviewed briefly and then the original contribution of the current study will be outlined.

From a physical point of view, the most important aspect of a multimaterials topology optimization problem is the representation of the material properties tensors as a function of local volume fractions and physical properties of individual phases. It is commonly performed using material interpolation [4], homogenization [3, 5] or averaged Hashin-Shtrikman (HS) upper-lower bounds [3] approaches.

Using the homogenization method, design of three-phase composites with extremal thermal expansion, piezoelectricity and bulk modulus are considered in [6, 7], [8] and [9] respectively (cf. [10]). These design problems have been formulated as some three-phase topology optimization problems. In these works authors however did not attend to numerical solution of resulted systems of optimality conditions and passed it to a general purpose optimization black-box. In addition to these works, there are some works in which the authors paid specific attention to introduce special numerical methods to solve multiphase topology optimization problems. Prior to reviewing these works, it is worth to mention that difficulties aroused from the numerical solution of multiphase topology optimization problems. These difficulties are mainly related to the mathematical structure of design space. In the case of two-phase topology optimization, the design variable is a field variable that includes the local volume fraction of one of the contributing phases. The corresponding design space is commonly a sufficiently regular field with local lower and upper bounds, and a global integral constraint. This structure makes it possible to keep

the solution strictly feasible with respect to optimization constraints; for instance using the optimality criteria [3] or gradient projection [11] methods. However, in the case of multiphase topology optimization problems not only there are multiple field variables (related to volume fractions of different phases), their corresponding lower and upper bounds and multiple global integral constraints but also there are pointwise constraints on summation of volume fraction fields (the incompressibility constraint). The later constraints significantly increases the computational cost of the corresponding numerical solution.

To decrease the complexity of design domain, a phase-field approach coupled to a special materials interpolation method has been used in [12, 13] to solve three-phase topology optimization problems. In this method one field variable has been used to determine the volume fraction of different phases. Although it still suffers from the existence of multiple global constraints, the mentioned local constraints (computationally expensive constraints) are naturally removed using mono-field design variable. However because of using mono-field materials interpolation method, it permits only the existence of at most two phases within each computational cell simultaneously, i.e., triple junctions are mandatory filtered by this approach.

The variational multilevel sets approach (cf. [14, 15]) has been adapted in [16–18] to solve multimaterials topology optimization problems. Later, the piecewise-constant variational level set method (cf. [17]) has been used in [19, 20] to improve the computational cost of former approach. Some of drawbacks of level-set based methods are: the mesh dependency of convergence speed (due to CFL stability condition) and dependency of final topology to initial design. Moreover, the global volume constraints are only satisfied at final solutions.

Using multimaterial phase-field approach based on Cahn-Hilliard equation, a general method to solve multiphase structural topology optimization problems have been introduced in [21, 22]. The intrinsic volume preserving property is the most important benefit of this method, i.e., the iterations will be kept strictly feasible with respect to design domain without any further effort. The slow convergence of the phase field method is however the main difficulty of this approach. For instance over 10^4 iterations are commonly required to find a suitable topology. This limitation stems in severe time-step restriction due to the stability of the corresponding forth order parabolic PDEs for evolution of phase-field variables. Moreover, due to the existence of an interfacial penalization in Cahn-Hilliard energy functional, i.e., penalizes the total area of interfaces, the resulted topologies are always biased to topologies with minimal interfaces and curvatures. It is worth mentioning that for binary phase composites under periodic boundary conditions rank-N lamination is known to be the optimal microstructure, while due to large interfacial area such solutions are discouraged by phase-field approach. This undesirable effect also appears in variational level set based approaches, as they similarly include an interfacial penalization term due to regularization of level set functions. Another difficulty with phase field approach is existence of many phenomenological constants in the model that should be carefully selected to find desired solutions.

For the sake of completeness it is worth to mention that the extension of evolutionary topology optimization methods [23] to manage multiple materials is another candidate to solve multiphase topology optimization problems. In [24] a bi-directional evolutionary topology optimization method has been suggested to solve multiphase structural topology optimization problems.

In the present work a new multimaterials topology optimization method based on the block coordinate descent approach will be introduced. The main properties of the presented method are: generality, simplicity, efficiency and the ease of implementation. An interesting feature of our approach is that it provides a general framework to convert almost every binary phase topology optimization solver to its multiphase counterpart by a few modifications; without special regard to the nature of related topology optimization problem. Moreover, this work goals to be a tutorial for computer implementation of the presented approach. For this purpose the MATLAB implementation of our approach to solve some model problems will be discussed in details.

2. Statement of multiphase topology optimization problem at abstract level

The statement of a typical multiphase topology optimization problem will be presented in this section. Because the algorithm, presented in the next section, is sufficiently general, the presentation here will be kept at an abstract level, to utilize its application to a wide class of topology optimization problems.

Consider Ω as the design domain. It is assumed that Ω is a fixed nonempty and sufficiently regular subset of $\mathbb{R}^d$ ($d = 1, 2, 3$). In an multimaterial topology optimization problem the goal is to find the optimal distribution of $p \in \mathbb{N}$ ($p \geqslant 2$) number of distinct materials inside Ω. Possibly, the void could be considered as a separate phase in our formulation. Therefore, the binary phase topology optimization problem here implies the classical void-material topology optimization problem. The materials distribution is determined by the local volume fraction fields, α_i ($i = 1, \ldots, p$), corresponding to the contributing phases (note that $\alpha_i = \alpha_i(x)$). It is assumed that $\alpha_i \in \mathcal{V}(\Omega)$, where $\mathcal{V}$ is a sufficiently regular function space (in many practical topology optimization problems L^∞ is the minimal regularity requirement). Obviously we have the following pointwise bound constraints on every α_i,

$$\boldsymbol{l}_i \leqslant \alpha_i \leqslant \boldsymbol{u}_i, \quad i = 1, \ldots, p \tag{1}$$

where $\mathbf{0} \leqslant \boldsymbol{l}_i \leqslant \boldsymbol{u}_i \leqslant \mathbf{1}$ and the inequalities are understood componentwise here. It is worth to mention that to fix the topology of a material at a specific portion of Ω, it is suffice to use the desired fixed value instead of lower and upper bounds in 1. Since no overlap and gap are allowed in the desired design, the summation of volume fractions at every point $x \in \Omega$ should be equal to unity, i.e.,

$$\sum_{i=1}^{p} \alpha_i = 1 \tag{2}$$

where the summation operator is understood componentwise (local) here. In many topology optimization problems we usually have one global volume constraint on the total volume of each material inside Ω, i.e.,

$$\int_\Omega \alpha_i \, dx = \Lambda_i |\Omega|, \quad i = 1, \ldots, p \tag{3}$$

where Λ_i are user defined parameters; obviously $0 \leqslant \Lambda_i \leqslant 1$ and $\sum_{i=1}^{p} \Lambda_i = 1$. These constraints are commonly called as resource constraints. Without loss of generality, we restrict ourselves to equality resource constraints in this study. Considering 2, it is possible to remove volume fraction field of one material and solve

the problem for $p-1$ unknown topology indicator fields. This way is frequent in binary phase topology optimization problems. However, applying this reduction has no benefit in the present study.

For the purpose of convenience we condense all design vectors (scalar fields) into a single vector field denoted by α, i.e., $\alpha = \{\alpha_1, \dots, \alpha_p\}$. Since we solve a finite-dimensional version of original problem in practice, we use superscript h in this study to denote the (finite dimensional) vector corresponding to each topology field. Therefore, after space discretization, the admissible design space, $\mathcal{A}$, could be defined as follows:

$$\mathcal{A}^h := \left\{ \alpha^h \in \{\ \alpha_i^h \in \mathcal{V}^h(\Omega^h)\ \}_{i\in\{1,\dots,p\}} \;\middle|\; \begin{array}{ll} \sum_{i=1}^p \alpha_i^h = 1, & \\ \int_{\Omega^h} \alpha_i^h \, dx = \Lambda_i |\Omega^h|, & i = 1, \dots, p \\ \boldsymbol{l}_i^h \leqslant \alpha_i^h \leqslant \boldsymbol{u}_i^h, & i = 1, \dots, p \end{array} \right\}$$

When $p = 2$, the structure of simplex $\mathcal{A}^h$ is very simple and there are efficient numerical methods to manage these constraints during optimization cycles, for instance using optimality criteria [3] or projected gradient [11] methods. However, when $p > 2$ the structure of $\mathcal{A}^h$ is very complex and to our knowledge there is no (time-linear) algorithm to manage these constraints. This is indeed one of the main difficulties against efficient solution of multiphase topology optimization problems. It is worth to mention that in Cahn-Hilliard method used in [22] these constraints are naturally manipulated without additional computational penalty. The presented approach in this study provides a very simple way to manage these constraints in expense of a computational complexity similar to that of the optimality criteria or the projected gradient approaches. In fact, our algorithm here has a time-linear computational complexity to manage these constraints.

In every topology optimization problem, the local material properties are a local function of volume fractions of contributing phases. Depending on the type of problem, the material properties could be the specific mass density, thermal conductivity, elasticity tensor, etc. Having materials properties as a function of local volume fractions is one of the most important ingredient of every topology optimization problem. There are many different methods to compute local materials properties based on the local volume fractions, for instance: linear mixture approach, SIMP penalized mixture approach (cf. [3, 4]), averaged lower and upper Hashin-Shtrikman bounds (cf. [4, 9, 25]) and homogenization approach (cf. [3, 5]). In the present study it is assumed that the functionality of local materials properties are provided by user. To simplify expositions, this function will be denoted by $\mathcal{M}(\alpha)$ in the present study. Note that we do not care whether there is a single material property or multiple ones and simply use $\mathcal{M}(\alpha)$ to denote the function that interpolates the materials properties from local volume fraction data.

A PDE or a system of PDE constraint(s) is another ingredient of every topology optimization problem. Based on the physics of problem the PDE could be transient, static or hybrid. Without loss of generality we limit ourselves here to static problems. Assume that the solution of PDE constraint is denoted by $U \in \mathcal{U}(\Omega)$, where $\mathcal{U}$ is a sufficiently regular function space. Obviously U is an implicit function of α, i.e., $U(x) = U(\alpha(x))$. Note that U could be a scalar field (e.g. in thermal problems), vector field (e.g. in elasticity problems) or combination of scaler and vector fields (e.g. in multiphysics problems like thermomechanical problems). The partial differential operator corresponding to PDE constraint(s) is denoted by operator $\mathcal{R}(\cdots)$ in the present study. Therefore, the discretized version of PDE

constraint(s) could be expressed as follows:

$$\mathcal{R}^h\big(\mathcal{M}(\alpha^h), U^h(\alpha^h)\big) = 0 \quad \texttt{in} \quad \Omega^h \tag{4}$$

where $U^h \in \mathcal{U}^h(\Omega^h)$ and the corresponding boundary conditions are encapsulated into the discritized version of PDE operator.

Definition of the objective functional is the remaining part of the problem statement here. In general the objective functional, $\mathcal{J}(\cdots)$, is an integral over Ω which depends on α and U. The discretized form of objective functional is expressed in the following form in this study:

$$\min_{\alpha^h} \mathcal{J}^h\big(\alpha^h, U^h(\alpha^h)\big) \tag{5}$$

Considering above terminology, every multimaterial topology optimization problem could be expressed in the following abstract from:

$$\min_{\alpha^h \in \mathcal{A}^h} \mathcal{J}^h\big(\alpha^h, U^h(\alpha^h)\big) \quad \texttt{subject to}: \quad \mathcal{R}^h\big(\mathcal{M}(\alpha^h), U^h(\alpha^h)\big) = 0 \quad \texttt{in} \quad \Omega^h \tag{6}$$

3. Statement of alternating active phase algorithm at abstract level

The statement of alternating active phase algorithm will be presented in this section. Currently there are many scientific and commercial codes to solve binary-phase topology optimization problems based on different methods. However, extension of these codes to solve multiphase topology optimization problems is not a straightforward procedure. In addition, by such an extension, algorithms may loss efficiency and robustness. The motivation for development of alternating active phase algorithm is to provide a general framework to convert binary phase topology optimization codes to multiphase ones by minimal efforts and modifications. Moreover, keeping the efficiency and robustness of original algorithms are another factors considered in the present work. Because our algorithm is sufficiently general, the presentation here will be kept at an abstract level, to make it possible to apply this algorithm to different classes of topology optimization problems.

The alternating active phase algorithm includes an outer iteration in which $p(p-1)/2$ binary phase topology optimization subproblems are solved partially (i.e., incomplete solution), using an appropriate binary phase topology optimization solver. This procedure could be performed either independently (in parallel) or sequentially; similar to Jacobi and Gauss-Seidel iterations, respectively. During the solution of every subproblem, the topologies of $p-2$ phases are fixed to the last known values and those of two remained phases (active phases) will be varied. If active phases are denoted by subscripts "a" and "b", α^h_{ab} is used to denote the design vector in which α^h_a and α^h_b could be varied and α^h_i is fixed for $i \neq \{a, b\}$.

Therefore, an appropriate binary phase topology optimization solver is required in the alternating active phase algorithm. This solver is called as the internal optimization solver, henceforth. Roughly speaking, it is easy to make the internal optimization solver by modification of binary phase topology optimization algorithm. The first required modification is to replace the binary phase materials properties operator $\mathcal{M}(\cdot)$ with its corresponding multiphase counterpart, $\mathcal{M}(\alpha^h_{ab})$.

In the binary phase topology optimization algorithms, the admissible design space, $\mathcal{A}^h_{\texttt{bi}}$, has a simple structure. If the local volume fraction of stiff phase is

denoted by field vector w^h, $\mathcal{A}^h_{\texttt{bi}}$ could be expressed as follows:

$$\mathcal{A}^h_{\texttt{bi}} := \left\{ \;\; w^h \in \mathcal{V}^h(\Omega^h) \;\;\; \middle| \;\;\; \boldsymbol{l}^h \leqslant w^h \leqslant \boldsymbol{u}^h, \quad \int_{\Omega^h} w^h \; dx = \Lambda \; |\Omega^h| \;\; \right\}$$

In practice the lower and upper bounds $\boldsymbol{l}^h$ and $\boldsymbol{u}^h$ are set near $\mathbf{0}^h$ and $\mathbf{1}^h$ respectively. Because the topologies of $p-2$ phases are fixed during every binary phase topology optimization step in our algorithm, the second required modification is to modify the structure of $\mathcal{A}^h_{\texttt{bi}}$. Assuming the topologies of $p-2$ phases are fixed inside Ω^h, the remained volume fraction field, denoted by $\boldsymbol{r}^h_{ab}$, which could be varied during every binary phase topology optimization sub-problem is computed as follows:

$$\boldsymbol{r}^h_{ab} = \mathbf{1}^h - \sum_{\substack{i=1 \\ i \neq \{a,b\}}}^{p} \alpha^h_i \tag{7}$$

Since $\sum_{i=1}^{p} \alpha_i = 1$ within each computational cell, it is only required to take the volume fraction of a as the design variable of the binary phase topology optimization sub-problem. After solving this sub-problem, the volume fraction field of phase b (background phase) could be computed by the following relation:

$$\alpha^h_b = \boldsymbol{r}^h_{ab} - \alpha^h_a \tag{8}$$

Considering 7, the corresponding (temporary) upper bound to phase a, $\boldsymbol{u}^h_{a,\texttt{temp}}$, during the binary phase topology optimization sub-problem should be computed as follows (note that lower bound does not need any modification):

$$\boldsymbol{u}^h_{a,\texttt{temp}} = \min \; (\boldsymbol{u}^h_a, \; \boldsymbol{r}^h_{ab}) \tag{9}$$

Theretofore, the admissible design space corresponding to the binary phase topology optimization sub-problem in which the topologies of phases a and b will be varied, $\mathcal{A}^h_{ab}$, could be expressed as follows (phase b is assumed to be the background phase):

$$\mathcal{A}^h_{ab} := \left\{ \;\; \alpha^h_a \in \mathcal{V}^h(\Omega^h) \;\;\; \middle| \;\;\; \boldsymbol{l}^h_a \leqslant \alpha^h_a \leqslant \boldsymbol{u}^h_{a,\texttt{temp}}, \quad \int_{\Omega^h} \alpha^h_a \; dx = \Lambda_a \; |\Omega^h| \;\; \right\}$$

Therefore the internal optimization solver, denoted by operator $\mathcal{S}^h(\alpha^h_{ab}, \mathcal{M}, U^h, \mathcal{R}^h)$, should solve the following binary topology optimization problem:

$$\min_{\alpha^h_{ab} \in \mathcal{A}^h_{ab}} \mathcal{J}^h\big(\alpha^h_{ab}, U^h(\alpha^h_{ab})\big) \quad \texttt{s.t.:} \quad \mathcal{R}^h\big(\mathcal{M}(\alpha^h_{ab}), U^h(\alpha^h_{ab})\big) = 0 \quad \texttt{in} \quad \Omega^h \tag{10}$$

Because of nonlinear nature of topology optimization problems, the internal optimization solver uses an iterative procedure. Therefore, it needs a convergence and/or termination criteria. The infinity norm, $\|\cdot\|_\infty$, of change in design vector during two consecutive optimization cycles is commonly used as the termination criterion. In the other word, when the maximum of local change in the topology is smaller than a user defined threshold, the iterations are discarded, and the last design vector is reported as an approximate optimal solution. Moreover, one may define an upper bound on the number of optimization cycles (posing limitation on the computational cost) as a stopping criterion. Input parameters "`tol_in`" and "`iter_max_in`" are defined as the stopping criteria for internal optimization algorithm in the present study.

To avoid checkerboard instability and/or length scale control (cf. [26]), a filtering (smoothing) mechanism is commonly used in binary phase topology optimization problems. The filter parameters have a significant effect on the convergence rate

and resulted topologies of the internal optimization solver. The selection of filter parameters is more serious in numerical solution of multiphase topology optimization problems. According to our numerical experience (will be mentioned later), dynamic change in filter parameters leads to a more efficient and robust optimization algorithm. The filter parameters are denoted by "`filter_params`" in the present study.

Prior to the presentation of alternating active phase algorithm, we will made some elementary assumptions on the properties of internal optimization solver.

Assumption 1. *The internal optimization solver is globally convergent*[1].

Assumption 2. *The internal optimization solver is interior with respect to PDE and design constraints. In the other word, iterations of the internal optimization solver will be strictly feasible with respect to all constraints.*

As it is formerly mentioned, in the case of two phase topology optimization problems, the structure of admissible design domain is very simple such that it is possible to keep optimization iterations strictly feasible with respect to $\mathcal{A}^h$, in expense of a negligible computational cost. Moreover, in most of topology optimization algorithms the corresponding PDE(s) is solved at every stage of optimization. Therefore, 2 is not a restrictive assumption.

In practical engineering design problems, we may not essentially look for a local solution to the mathematical model corresponding to the physical problem, but some improvements on an initial design could be of interest, or even sufficient. In this case, having a monotone interior algorithm could be suffice. Note that an interior algorithm here does not imply an interior-point algorithm. For instance, the classical optimality criteria algorithm possesses requirements of assumption 2. The optimality criteria algorithm however is not globally convergent, except in some special cases. Therefore, we may relax assumption 1 to the following counterpart:

Assumption 3. *The internal optimization solver is monotone with respect to the objective functional. In the other word, the objective functional decreases monotonically as iterations of the internal optimization algorithm proceed.*

Assuming phases are arranged from stiffer to softer as phase counter increases from 1 to p, the Gauss-Seidel version of alternating active phase algorithm is presented in algorithm 1. The Jacobi version of alternating active phase algorithm could be expressed by some simple modifications in algorithm 1. The required modifications are mentioned in algorithm 2 (to save the space, similar parts are removed).

[1]An algorithm is globally convergent, if it converges to a local solution starting from an arbitrary initial guess.

Algorithm 1: Alternating active phase algorithm (Gauss-Seidel version)

Problem definition:

given p, Ω, $\mathcal{M}$, $\mathcal{R}$, $\mathcal{J}$, $\{\Lambda_1, \dots, \Lambda_p\}$, $\{\boldsymbol{l}_1, \dots, \boldsymbol{l}_p\}$, $\{\boldsymbol{u}_1, \dots, \boldsymbol{u}_p\}$

Solution parameters:

given h, `tol_in`, `iter_max_in`, `filter_params`, `tol_out`, `iter_max_out`

Discretization:

compute Ω^h, $\mathcal{R}^h$, $\mathcal{J}^h$, $\{\boldsymbol{l}_1^h, \dots, \boldsymbol{l}_p^h\}$, $\{\boldsymbol{u}_1^h, \dots, \boldsymbol{u}_p^h\}$

Initialization:

$\{\alpha_1^h, \dots, \alpha_p^h\} = \{\Lambda_1, \dots, \Lambda_1\}$, `iter_out` $= 0$

Outer iterations:

repeat
  $\alpha^{h,\mathtt{old}} = \alpha^h$
  for `a` $= 1$ **to** `p` **do**
    for `b` $=$ `a` $+ 1$ **to** `p` **do**
      $\alpha_{ab}^h = \alpha^h$
      `params_in` $=$ {`tol_in`, `iter_max_in`, `filter_params`}
      $\alpha^h =$ solution of $\mathcal{S}^h(\alpha_{ab}^h, \mathcal{M}, U^h, \mathcal{R}^h)$ with `params_in`
    end
  end
  update `filter_params`
  `iter_out` $=$ `iter_out` $+ 1$
  `change_out` $= \|\alpha^h - \alpha^{h,old}\|_\infty$
until (`change_out` $<$ `tol_out`) `and` (`iter_out` $<$ `iter_max_out`) ;

Algorithm 2: Alternating active phase algorithm (Jacobi version)

$\cdots$

repeat
  $\alpha^{h,\mathtt{old}} = \alpha^h$, $\zeta^h = \mathbf{0}$
  for `a` $= 1$ **to** `p` **do**
    for `b` $=$ `a` $+ 1$ **to** `p` **do**
      `params_in` $= \cdots$
      $\zeta^h = \zeta^h +$ solution of $\mathcal{S}^h(\alpha_{ab}^h, \mathcal{M}, U^h, \mathcal{R}^h)$ with `params_in`
    end
  end
  $\alpha^h = \zeta^h / (p(p-1)/2)$
  $\cdots$
until $(\cdots)$;

4. Comment on the convergence theory of alternating active phase algorithm

In this section we briefly comment on the convergence theory of the alternating active phase algorithm. As it is mentioned in the previous section, having an interior and monotone topology optimization algorithm is suffice for many practical topology optimization applications. The alternating active phase algorithm inherits this property from its internal optimization solver. Therefore the following corollary is evident:

Corollary 4.1. *Alternating active phase algorithm is monotonic and interior under assumptions 2 and 3.*

To establish the global convergence theory of alternating active phase algorithm is a more delicate issue. We believe that such a theory should be developed based on the detailed properties of the internal optimization solver.

It is worth to mention that the alternating active phase algorithm is in close connection to the cyclic coordinate descent method (c.f. [section 5.4.3 of 27] and [section 8.9 of 28]) which is an ancient optimization algorithm. More precisely the alternating active phase algorithm is similar to block-wise cyclic coordinate descent methods. Although cyclic coordinate descent methods are very old and inefficient in general, they are revisited recently and significant contribution has been done during the past two decades. It is mainly because of their promising efficiency to solve special classes of large scale real-world problems. There are many theoretical works on the convergence of cyclic coordinate descent methods which could be used to establish the convergence theory of the alternating active phase algorithm, under some specific conditions. Because our goal is not to do a theoretical study in this work, we end this section refereing interested readers to [27–33].

5. Model problems

5.1. **Minimum compliance topology optimization problem.** The volume constrained (binary phase) minimum compliance topology optimization problem is one of the most common model problem in the literature of topology optimization. In this model problem, the unknown materials distribution is determined such that the stiffness of structure is maximized. To avoid the trivial solution, a global volume constraint is applied on the total volume occupied by the stiff phase in the final design. In this section we recall multi-phase version of this model problem. Our test problem here is similar to that of [22].

In the case of minimum compliance topology optimization problem, the materials properties operator $\mathcal{M}$ is analog to the elasticity tensor, $\mathbf{C}(x) = \mathbf{C}(\alpha(x))$. Note that $\mathbf{C}$ is a forth order supersymmetric tensor (symmetric in both the right and the left Cartesian index pair, together with symmetry under the interchange of the pairs). Following [22], the SIMP modified version of linear interpolation is used within our materials interpolation operator:

$$\mathbf{C}(\alpha) = \sum_{i=1}^{p} \alpha_i^q \, \mathbf{C}_i \tag{11}$$

where $\mathbf{C}_i$ is the constant stiffness tensor corresponding to phase i-th. For a given materials distribution α, the PDE operator $\mathcal{R}$ could be expressed as follows:

$$\left\{\begin{array}{rcll} \nabla\cdot(\mathbf{C}:\mathcal{D}(\mathbf{u})) &=& \mathbf{f}(x) & \texttt{in}\quad \Omega \\ \mathbf{u}(x) &=& \hat{\mathbf{u}}(x) & \texttt{on}\quad \Gamma_u \\ (\mathbf{C}:\mathcal{D}(\mathbf{u}))\cdot\mathbf{n} &=& 0 & \texttt{on}\quad \Gamma_f \\ (\mathbf{C}:\mathcal{D}(\mathbf{u}))\cdot\mathbf{n} &=& \hat{\mathbf{t}}(x) & \texttt{on}\quad \Gamma_t \end{array}\right. \tag{12}$$

where $\mathbf{u}$ denotes the displacement field which is analog to the state variable U in our formerly mentioned abstract formulation, $\mathbf{f}$ denotes the volumetric body force, $\hat{\mathbf{u}}$ denotes the prescribed displacement on boundaries Γ_u, Γ_f denotes the traction free part of boundaries, $\hat{\mathbf{t}}$ denotes the traction of structure with environment through traction boundaries, Γ_t and $\partial\Omega = \Gamma_u \cup \Gamma_f \cup \Gamma_t$. Moreover, $\mathcal{D}(\mathbf{u}) := \frac{1}{2}(\nabla\mathbf{u} + (\nabla\mathbf{u})^T) = \frac{1}{2}(\partial_i u_j + \partial_j u_i)_{i,j},\ i,j = 1\ldots,d$. Note that the double dot operator, :, denotes the usual contraction over two sets of indices. The objective functional, to be minimized, is defined as follows for this test problem:

$$\mathcal{J}(\alpha,\mathbf{u}) = \frac{1}{2}\int_\Omega (\mathbf{C}(\alpha):\mathcal{D}(\mathbf{u})):\mathcal{D}(\mathbf{u})\ dx \tag{13}$$

To solve this problem, we use a gradient based minimization approach. Therefore, the gradient of the objective functional should be computed. It could be trivially computed using the classical adjoint approach (c.f. [2, 3]).

5.2. **Minimum thermal compliance topology optimization problem.** Minimum thermal compliance problem (c.f. [3]) is the scalar counterpart of the minimum compliance topology optimization problem stated in section 5.1. In this case the materials properties operator should be replaced by thermal conductivity tensor, K:

$$K(\alpha) = \sum_{i=1}^{p} \alpha_i^q\ K_i \tag{14}$$

where K_i is the conductivity tensor corresponding to phase i-th. For a given materials distribution α, the PDE operator $\mathcal{R}$ could be expressed as follows:

$$\left\{\begin{array}{rcll} \nabla\cdot(K(\alpha)\ \nabla u) &=& f(x) & \texttt{in}\quad \Omega \\ u(x) &=& \hat{u}(x) & \texttt{on}\quad \Gamma_u \\ (K\ \nabla u)\cdot\mathbf{n} &=& \hat{q}(x) & \texttt{on}\quad \Gamma_q \end{array}\right. \tag{15}$$

where u denotes the temperature field which is analog to the state variable U in our abstract formulation, f denotes the thermal heat source, $\hat{u}$ denotes the prescribed temperature on boundaries Γ_u, $\hat{q}$ denotes the prescribed heat flux on Γ_q and $\partial\Omega = \Gamma_u \cup \Gamma_q$. The corresponding objective functional is defined as follows:

$$\mathcal{J}(\alpha,u) = \frac{1}{2}\int_\Omega K\ \nabla u\cdot\nabla u\ dx \tag{16}$$

6. MATLAB implementation of model problems

A 99-line MATLAB code has been introduced in [34] to solve minimum compliance topology optimization problems. Later, in [35] the size of Sigmund's 99-line MATLAB code has been reduced to 88 lines and its performance has considerably been improved by Sigmund and his co-workers. In this section, we shall comment on the list of major modifications required to modify the 88-line MATLAB code to

solve multiphase minimum compliance topology optimization problems. Because our goal here is to introduce the utility of the alternating phase framework, we will not care on the performance of our code. It is tried to keep names of parameters and variables in the code consistent to those of presented algorithm. Therefore, to save the space, the parameters and variables will not essentially be defined explicitly in this section (interested readers are recommended to consult [35]). The complete listing of MATLAB code is included in the supplement A of this paper.

The main function, to call multimaterial topology optimization module, has the following structure:

```
function main
[nx,ny,tol_out,tol_f,iter_max_in,iter_max_out,p,q,e,v,rf] = set_parameters ();
 multi_top(nx,ny,tol_out,tol_f,iter_max_in,iter_max_out,p,q,e,v,rf);
end
```

where `set_parameters` function defines the optimization parameters and mesh resolution, and function `multitop` does the multimaterial topology optimization procedure. `e` and `v` include the elasticity and volume fraction parameters of all phases respectively (analog to `E0` and `volfrac` in 88-line MATLAB code). Moreover, `rf` and `tol_f` denote the filter radius and filter adjusting tolerance; will be described later. In fact `filter_params` $\equiv$ {`rf`, `tol_f`}. A typical form of `set_parameters` function is as follows (these parameters are related to cantilever beam #1 test case described in section 7.1):

```
function [nx,ny,tol,tolf,im_in,im_out,p,q,e,v,rf] = set_parameters()
 nx = 96; ny = 48; tol = 0.001; tolf = 0.05; im_in = 2; im_out = 200;
 p = 3; q = 3; e = [2 1 1e-9]'; v = [0.4 0.2 0.4]'; rf = 8;
end
```

Following Algorithm 1, the MATLAB implementation of `multi_top` function could be expressed as follows:

```
function multi_top(nx,ny,tol_out,tol_f,iter_max_in,iter_max_out,p,q,e,v,rf)
 alpha = zeros(nx*ny,p);
 for i = 1:p
  alpha(:,i) = v(i);
 end
 % MAKE FILTER
 [H,Hs] = make_filter (nx,ny,rf);
 change_out = 2*tol_out; iter_out = 0;
 while (iter_out < iter_max_out) && (change_out > tol_out)
  alpha_old = alpha;
  for a = 1:p
   for b = a+1:p
    [obj,alpha] = bi_top(a,b,nx,ny,p,v,e,q,alpha,H,Hs,iter_max_in);
   end
  end
  iter_out = iter_out + 1;
  change_out = norm(alpha(:)-alpha_old(:),inf);
  fprintf('Iter:%5i Obj.:%11.4f change:%10.8f\n',iter_out,obj,change_out);
  % UPDATE FILTER
  if (change_out < tol_f) && (rf>3)
   tol_f = 0.99*tol_f; rf = 0.99*rf; [H,Hs] = make_filter (nx,ny,rf);
```

```
  end
  % SCREEN OUT TEMPORAL TOPOLOGY
  I = make_bitmap (p,nx,ny,alpha);
  image(I), axis image off, drawnow;
 end
end
```

In the above code function `make_filter` does the sensitivity filtering to avoid the topological instability during the optimization procedure. It is identical to that of 88-line MATLAB code. It is based on a single path of discrete Laplacian smoother with compact support of radius `rf`. It is worth to mention that other filters (e.g. PDE filter based on Helmholtz equation) presented in [35] led us to similar results, therefore, no attention will be paid here in regard to the selection of filter type.

```
function [H,Hs] = make_filter (nx,ny,rmin)
 ir = ceil(rmin)-1;
 iH = ones(nx*ny*(2*ir+1)^2,1);
 jH = ones(size(iH));
 sH = zeros(size(iH));
 k = 0;
 for i1 = 1:nx
   for j1 = 1:ny
     e1 = (i1-1)*ny+j1;
     for i2 = max(i1-ir,1):min(i1+ir,nx)
       for j2 = max(j1-ir,1):min(j1+ir,ny)
         e2 = (i2-1)*ny+j2; k = k+1; iH(k) = e1; jH(k) = e2;
         sH(k) = max(0,rmin-sqrt((i1-i2)^2+(j1-j2)^2));
       end
     end
   end
 end
 H = sparse(iH,jH,sH); Hs = sum(H,2);
end
```

Our numerical experiments in the present work have shown that the selection of filter radius is a key success of the optimization procedure. According to our experiments, the dynamic modification of this parameter is required to bypass undesirable local minimums. Staring from a sufficiently large `rf` and decreasing it gradually leads to a robust optimization procedure. To exploit this issue in our algorithm, we decrease `rf` (by factor `0.99` in this work) whenever `change_out` $<$ `tol_f`, as implemented in the above code. Moreover, to avoid the topological instability (cf. [26]) a strict lower bound is considered for parameter `rf` in our implementation (`rf` $>$ `3` in this work).

Unlike the binary phase topology optimization, the visualization of resulted topology is not a straightforward task. It is because we have multiple volume fraction fields, should be illustrated in a single plot. To do this, the function `make_bitmap` is designated in this work to generate an appropriate bitmap image from multi-field volume fraction data. The MATLAB implementation of this function is as follows (although its algorithm is general, we restrict our code to five phases in this study):

```
function I = make_bitmap (p,nx,ny,alpha)
```

```
 color = [1 0 0; 0 0 .45; 0 1 0; 0 0 0; 1 1 1];
 I = zeros(nx*ny,3);
 for j = 1:p
  I(:,1:3) = I(:,1:3) + alpha(:,j)*color(j,1:3);
 end
 I = imresize(reshape(I,ny,nx,3),10,'bilinear');
end
```

Matrix `color` is a $p \times 3$ matrix include desired `RGB` colors corresponding to desired colors for phase `1 : p`; the row i-th of `color` includes the desired color of phase i-th.

Function `bi_top` is the modified binary phase topology optimization function, i.e., modified version of `top88` in [35].

```
function [o,alpha] = bi_top(a,b,nx,ny,p,v,e,q,alpha_old,H,Hs,iter_max_in)
 alpha = alpha_old; iter_in = 0; nu = 0.3;
 %% PREPARE FINITE ELEMENT ANALYSIS
 A11 = [12 3 -6 -3; 3 12 3 0; -6 3 12 -3; -3 0 -3 12];
 A12 = [-6 -3 0 3; -3 -6 -3 -6; 0 -3 -6 3; 3 -6 3 -6];
 B11 = [-4 3 -2 9; 3 -4 -9 4; -2 -9 -4 -3; 9 4 -3 -4];
 B12 = [ 2 -3 4 -9; -3 2 9 -2; 4 9 2 3; -9 -2 3 2];
 KE = 1/(1-nu^2)/24*([A11 A12;A12' A11]+nu*[B11 B12;B12' B11]);
 nodenrs = reshape(1:(1+nx)*(1+ny),1+ny,1+nx);
 edofVec = reshape(2*nodenrs(1:end-1,1:end-1)+1,nx*ny,1);
 edofMat = repmat(edofVec,1,8)+repmat([0 1 2*ny+[2 3 0 1] -2 -1],nx*ny,1);
 iK = reshape(kron(edofMat,ones(8,1))',64*nx*ny,1);
 jK = reshape(kron(edofMat,ones(1,8))',64*nx*ny,1);
 %% DEFINE LOADS AND SUPPORTS (CANTILEVER BEAM)
 F = sparse(2*(ny+1)*(nx+1),1,-1,2*(ny+1)*(nx+1),1);
 fixeddofs = [1:2*ny+1];
 U = zeros(2*(ny+1)*(nx+1),1);
 alldofs = [1:2*(ny+1)*(nx+1)];
 freedofs = setdiff(alldofs,fixeddofs);
 %% INNER ITERATIONS
 while iter_in < iter_max_in
  iter_in = iter_in + 1;
  %% FE-ANALYSIS
  E = e(1)*alpha(:,1).^q;
  for phase = 2:p
   E = E + e(phase)*alpha(:,phase).^q;
  end
  sK = reshape(KE(:)*E(:)',64*nx*ny,1);
  K = sparse(iK,jK,sK); K = (K+K')/2;
  U(freedofs) = K(freedofs,freedofs)\F(freedofs);
  %% OBJECTIVE FUNCTION AND SENSITIVITY ANALYSIS
  ce = sum((U(edofMat)*KE).*U(edofMat),2);
  o = sum(sum(E.*ce));
  dc = -(q*(e(a)-e(b))*alpha(:,a).^(q-1)).*ce;
  %% FILTERING OF SENSITIVITIES
  dc = H*(alpha(:,a).*dc)./Hs./max(1e-3,alpha(:,a)); dc = min(dc,0);
  %% UPDATE LOWER AND UPPER BOUNDS OF DESIGN VARIABLES
  move = 0.2;
  r = ones(nx*ny,1);
  for k = 1:p
   if (k ~= a) && (k ~= b)
    r = r - alpha(:,k);
   end
```

```
  end
  l = max(0,alpha(:,a)-move);
  u = min(r,alpha(:,a)+move);
  %% OPTIMALITY CRITERIA UPDATE OF DESIGN VARIABLES
  l1 = 0; l2 = 1e9;
  while (l2-l1)/(l1+l2) > 1e-3
   lmid = 0.5*(l2+l1);
   alpha_a = max(l,min(u,alpha(:,a).*sqrt(-dc./lmid)));
   if sum(alpha_a) > nx*ny*v(a); l1 = lmid; else l2 = lmid; end
  end
  alpha(:,a) = alpha_a;
  alpha(:,b) = r-alpha_a;
 end
end
```

Comparing function `bi_top` and `top88`, the list of required modifications in `top88` are evident.

Therefore, by adding about 30 lines of code and doing a little modifications on `top88` code, we can solve multiphase topology optimization problem based on [35] approach. In the same manner, it should not be difficult to apply the alternating active phase framework to other binary phase topology optimization problems.

To illustrate the utility of above mentioned modification to similar problem with different physics, we did the implementation of our algorithm for solution of minimum thermal compliance topology optimization problems. For this purpose, 91-line MATLAB code presented in section 5.1.6 of [3] for minimum thermal compliance topology optimization problem was firstly modified according to 88-line MATLAB code of [35] recipes (to improve the computational performance). Then, the alternating active phase framework has been applied to it. The complete listing of multiphase version of minimum thermal compliance topology optimization solver is included in supplement B of this paper.

7. Numerical results and discussion

The success and performance of the alternating active phase algorithm will be studied in this section by means of numerical experiments. A personal computer with an AMD 2.4 GHz and 2.5 GB DDR2 RAM is used as the computational resource in this study.

7.1. **Minimum compliance topology optimization.**

The numerical results corresponding to the model problem mentioned in section 5.1 are presented in this section. For this purpose, three minimum compliance topology optimization problems which are frequently used in the literature are considered here. They are cantilever beam, bridge structure and MBB beam; as shown in figure 1. The load `F` which is shown in figure 1 is equal to unity in all of our test cases here. To compare our results to available literature, the geometries and boundary conditions of these test cases are selected similar to those of [22]. To save the computational cost, the symmetry of computational domain and loads is considered in bridge structure and MBB beam test cases (note that the presented results are however related to full domain in these cases).

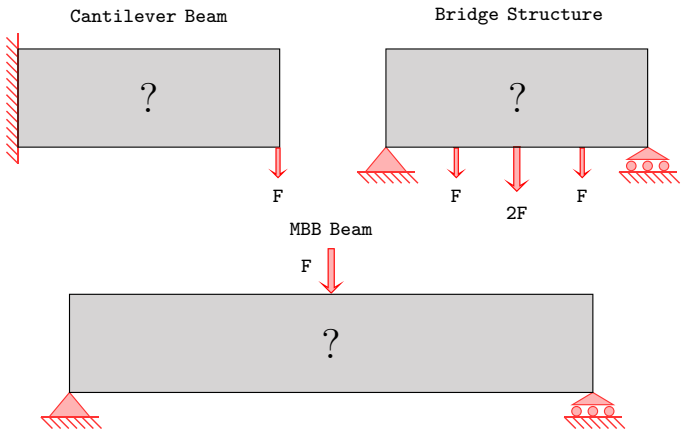


FIGURE 1. Geometries and boundary conditions corresponding to minimum compliance model problems (unspecified boundary condition denotes the traction free boundary condition).

The optimization control parameters, to be set in `set_parameters` function in all of test cases in this section are as follows:

```
function [...] = set_parameters ()
 ...
 tol_out = 0.001; tol_f = .05; iter_max_in = 2; iter_max_out = 200; q = 3; rf = 8;
 ...
end
```

Note that we did not pay any specific attention to adjust these parameters in our numerical experiments here. Note that the careful selection of these parameters could improve the computational performance and outcome of our algorithm. The other design optimization parameters corresponding to these test problems are listed in table 1 (these parameters are adapted from [22]).

TABLE 1. Design parameters of minimum compliance topology optimization test cases.

Test Problem	nx	ny	p	e	v
Cantilever Beam #1	96	48	3	$[2\ \ 1\ \ 1e-9]'$	$[0.4\ \ 0.2\ \ 0.4]'$
Cantilever Beam #2	96	48	4	$[4\ \ 2\ \ 1\ \ 1e-9]'$	$[0.2\ \ 0.1\ \ 0.1\ \ 0.6]'$
Bridge Structure #1	96	96	3	$[2\ \ 1\ \ 1e-9]'$	$[0.35\ \ 0.25\ \ 0.4]'$
Bridge Structure #2	96	96	3	$[2\ \ 1\ \ 1e-9]'$	$[0.4\ \ 0.2\ \ 0.4]'$
Bridge Structure #3	96	96	4	$[9\ \ 3\ \ 1\ \ 1e-9]'$	$[0.2\ \ 0.1\ \ 0.1\ \ 0.6]'$
MBB − Beam #1	96	48	3	$[2\ \ 1\ \ 1e-9]'$	$[0.4\ \ 0.2\ \ 0.4]'$
MBB − Beam #2	96	48	4	$[4\ \ 2\ \ 1\ \ 1e-9]'$	$[0.2\ \ 0.15\ \ 0.15\ \ 0.5]'$
MBB − Beam #3	96	48	4	$[9\ \ 3\ \ 1\ \ 1e-9]'$	$[0.16\ \ 0.08\ \ 0.08\ \ 0.68]'$

The MATLAB implementation of boundary conditions corresponding to cantilever beam case is identical to lines 77 and 78 of the MATLAB code presented in section 6. For the other two cases, they should be replaced by:

```
...
%% DEFINE LOADS AND SUPPORTS (HALF OF MBB-BEAM)
F = sparse(2,1,-1,2*(ny+1)*(nx+1),1);
fixeddofs = union([1:2:2*(ny+1)],[2*(nx+1)*(ny+1)]);
...
```

```
...
%% DEFINE LOADS AND SUPPORTS (HALF OF BRIDGE STRUCTURE)
F = sparse([2*(ny+1) 2*(ny+1)*(nx/2+1)],[1 1],[-2 -1] ,2*(ny+1)*(nx+1),1);
fixeddofs = union([1:2:2*(ny+1)],[2*(nx+1)*(ny+1)]);
...
```

The topological evolution during optimization cycles corresponding to the aforementioned test problems, are shown in figures 2 to 9. Each plot includes 15 frames which are respectively related to (outer) iterations: 5, 10, 15, 20, 30, 40, 50, 60, 80, 100, 120, 140, 160, 180, 200. The coloring scheme is adapted from [22], i.e., the hardness of phases is decreased based on the following order: red, blue, green and black. The history of objective function during the optimization iterations (outer iterations) is plotted in figure 10. Figure 11 compares our results to those of [22]. The consumed CPU time corresponding to these experiments are listed in table 2.

TABLE 2. Minimum compliance topology optimization test cases: CPU times in second.

Cantilever#1	Cantilever#2	Bridge#1	Bridge#2	Bridge#3	MBB#1	MBB#2	MBB#3
521	1435	1165	1174	1789	641	920	914

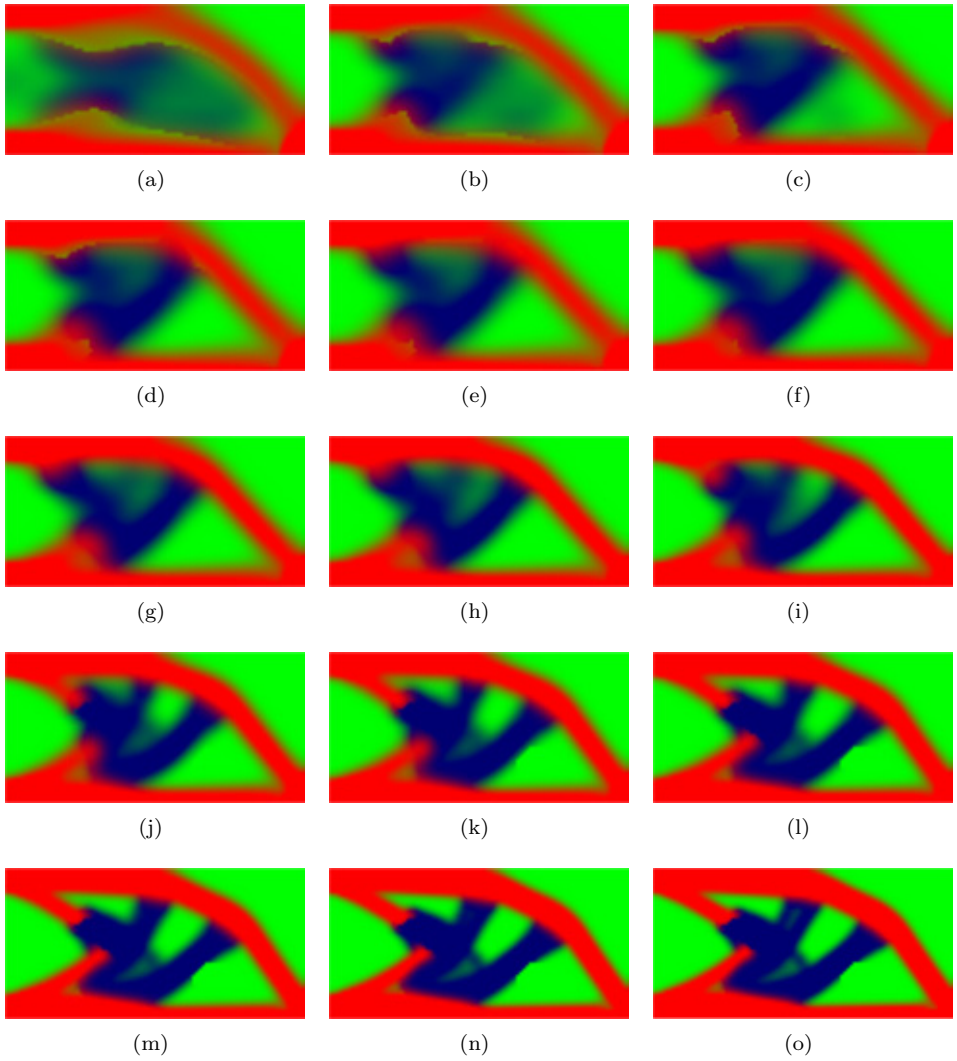


FIGURE 2. Topological changes during the optimization iterations for cantilever beam #1 test problem; a-o are respectively related to iterations 5, 10, 15, 20, 30, 40, 50, 60, 80, 100, 120, 140, 160, 180, 200.

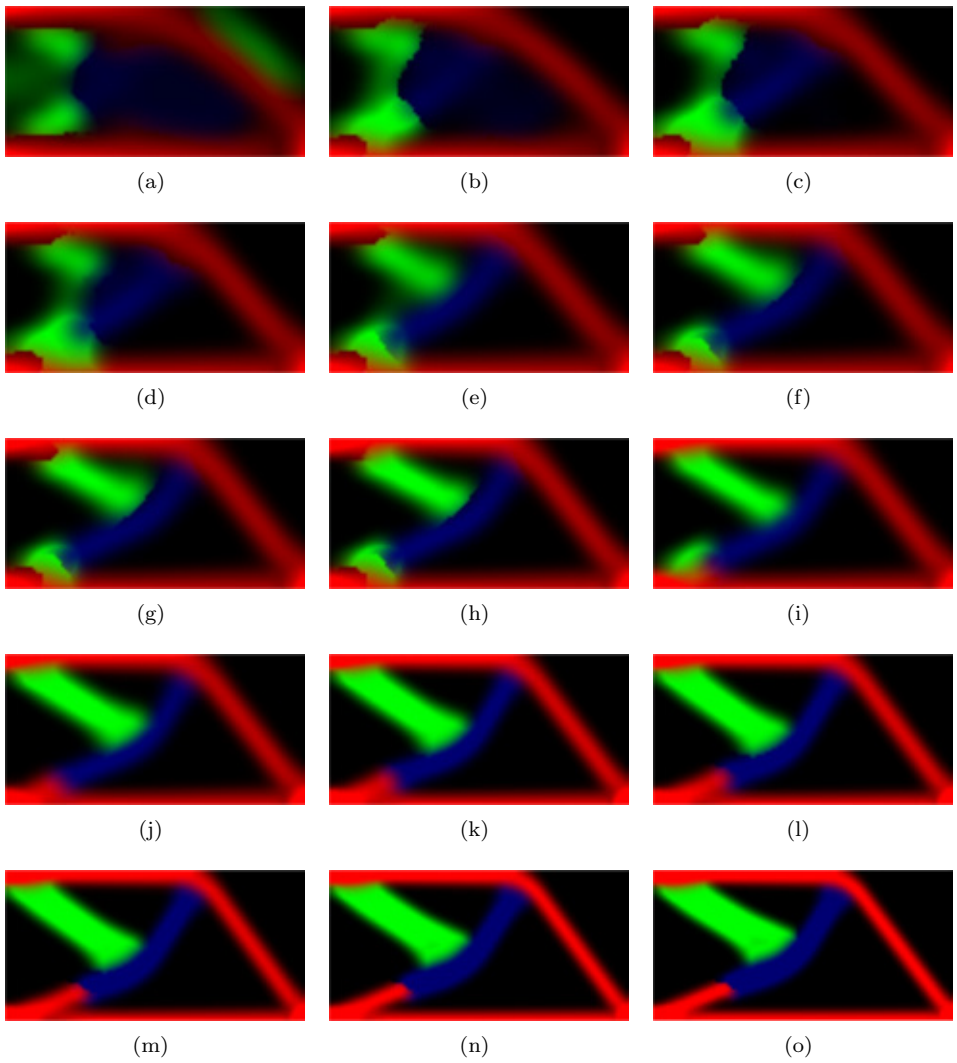


FIGURE 3. Topological changes during the optimization iterations for cantilever beam #2 test problem; a-o are respectively related to iterations 5, 10, 15, 20, 30, 40, 50, 60, 80, 100, 120, 140, 160, 180, 200.

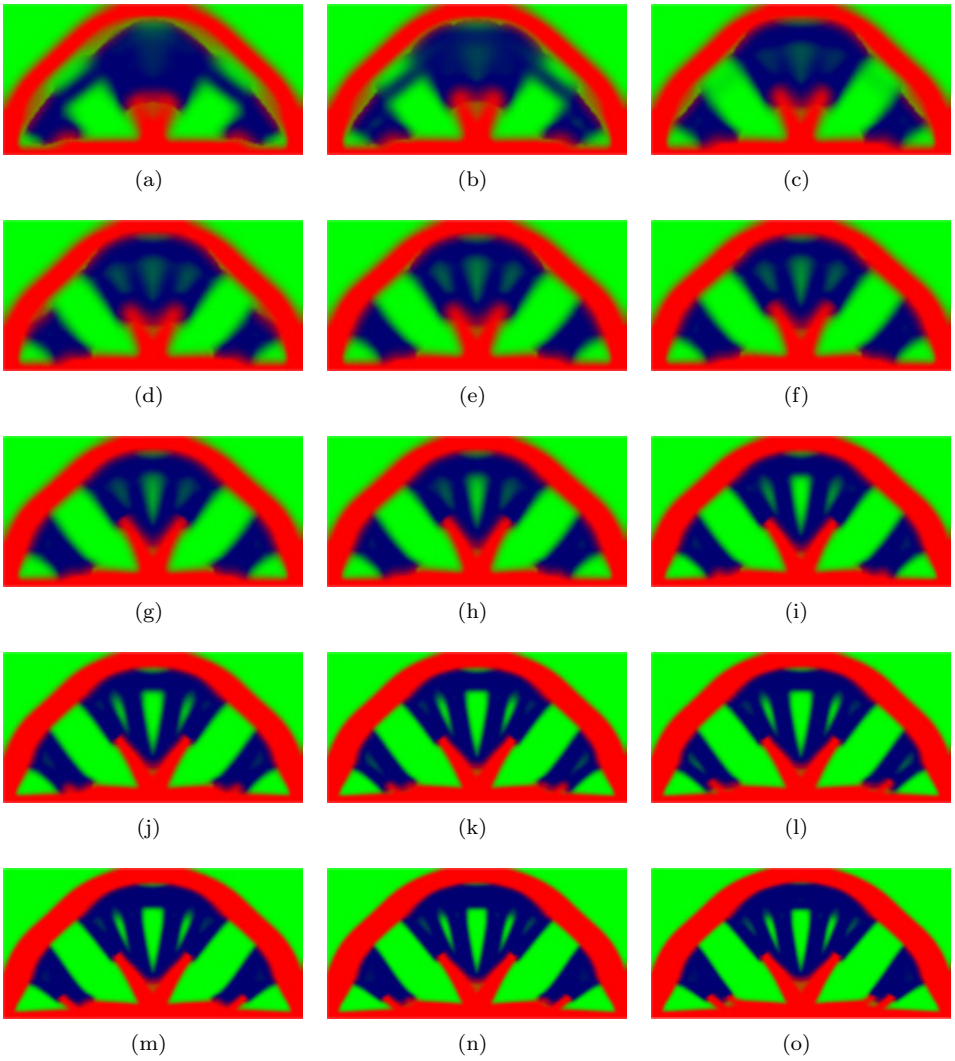


FIGURE 4. Topological changes during the optimization iterations for bridge structure #1 test problem; a-o are respectively related to iterations 5, 10, 15, 20, 30, 40, 50, 60, 80, 100, 120, 140, 160, 180, 200.

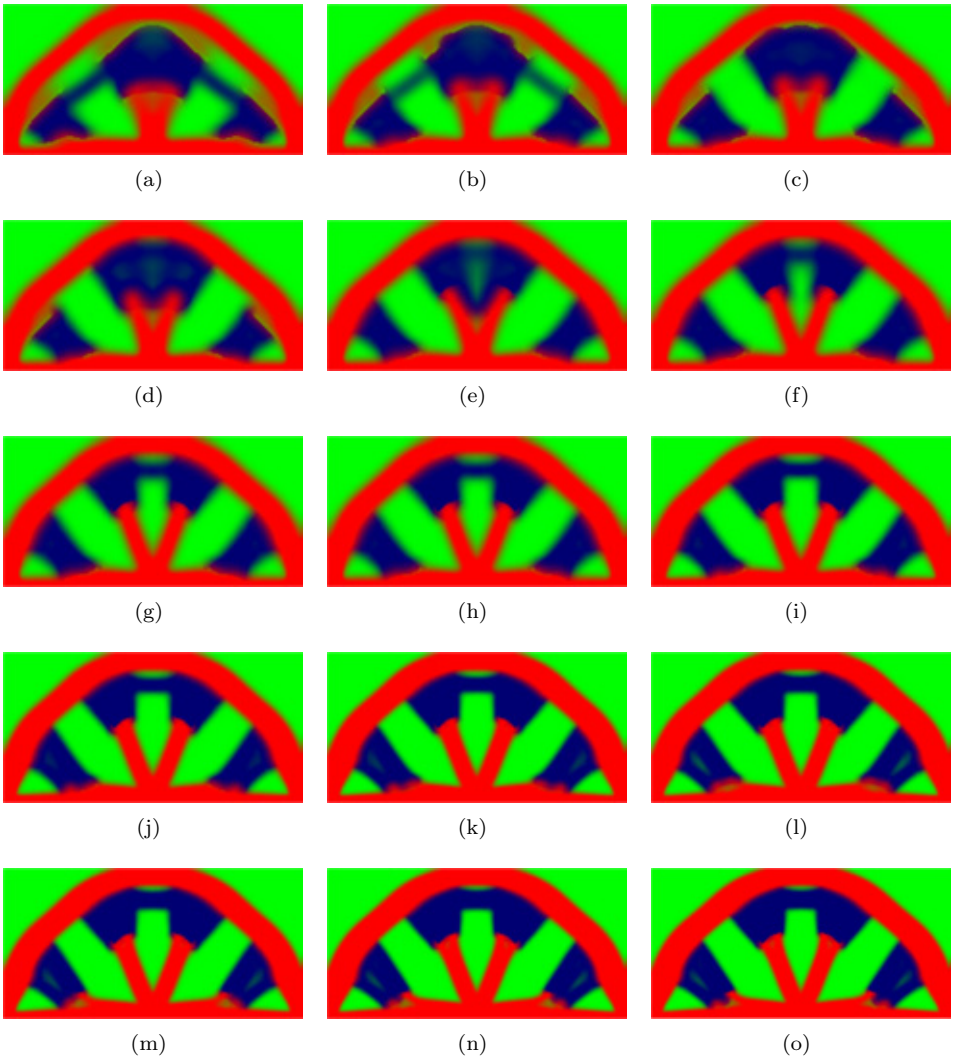


FIGURE 5. Topological changes during the optimization iterations for bridge structure #2 test problem; a-o are respectively related to iterations 5, 10, 15, 20, 30, 40, 50, 60, 80, 100, 120, 140, 160, 180, 200.

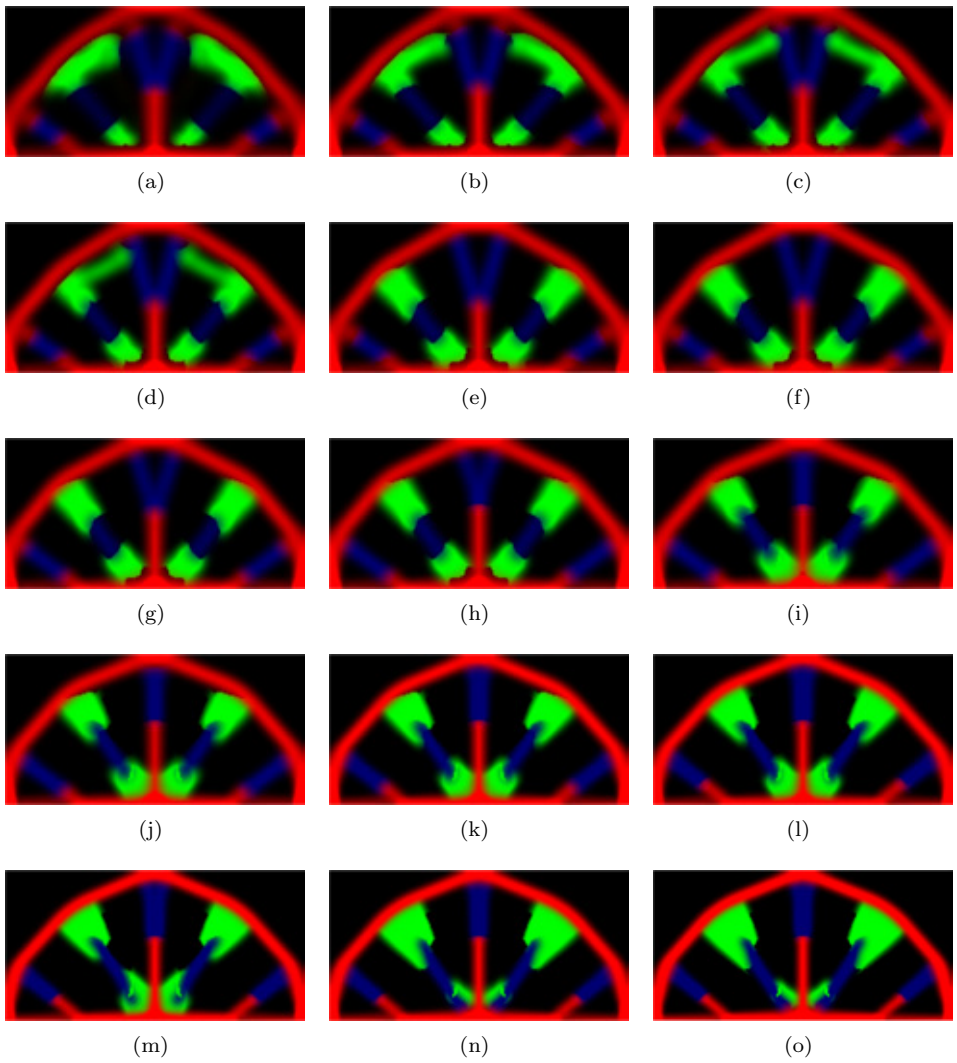


FIGURE 6. Topological changes during the optimization iterations for bridge structure #3 test problem; a-o are respectively related to iterations 5, 10, 15, 20, 30, 40, 50, 60, 80, 100, 120, 140, 160, 180, 200.

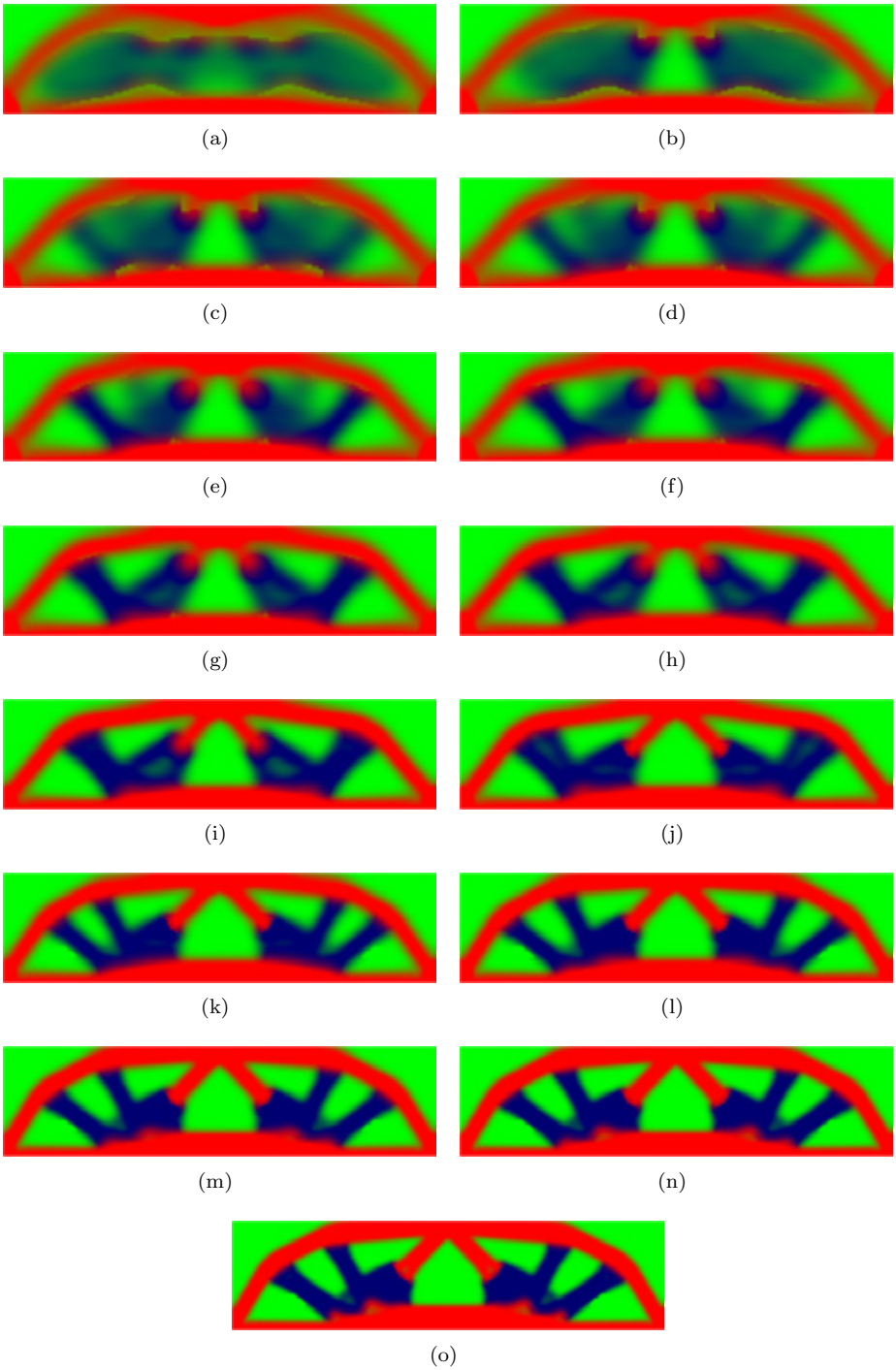


FIGURE 7. Topological changes during the optimization iterations for MBB beam #1 test problem; a-o are respectively related to iterations 5, 10, 15, 20, 30, 40, 50, 60, 80, 100, 120, 140, 160, 180, 200.

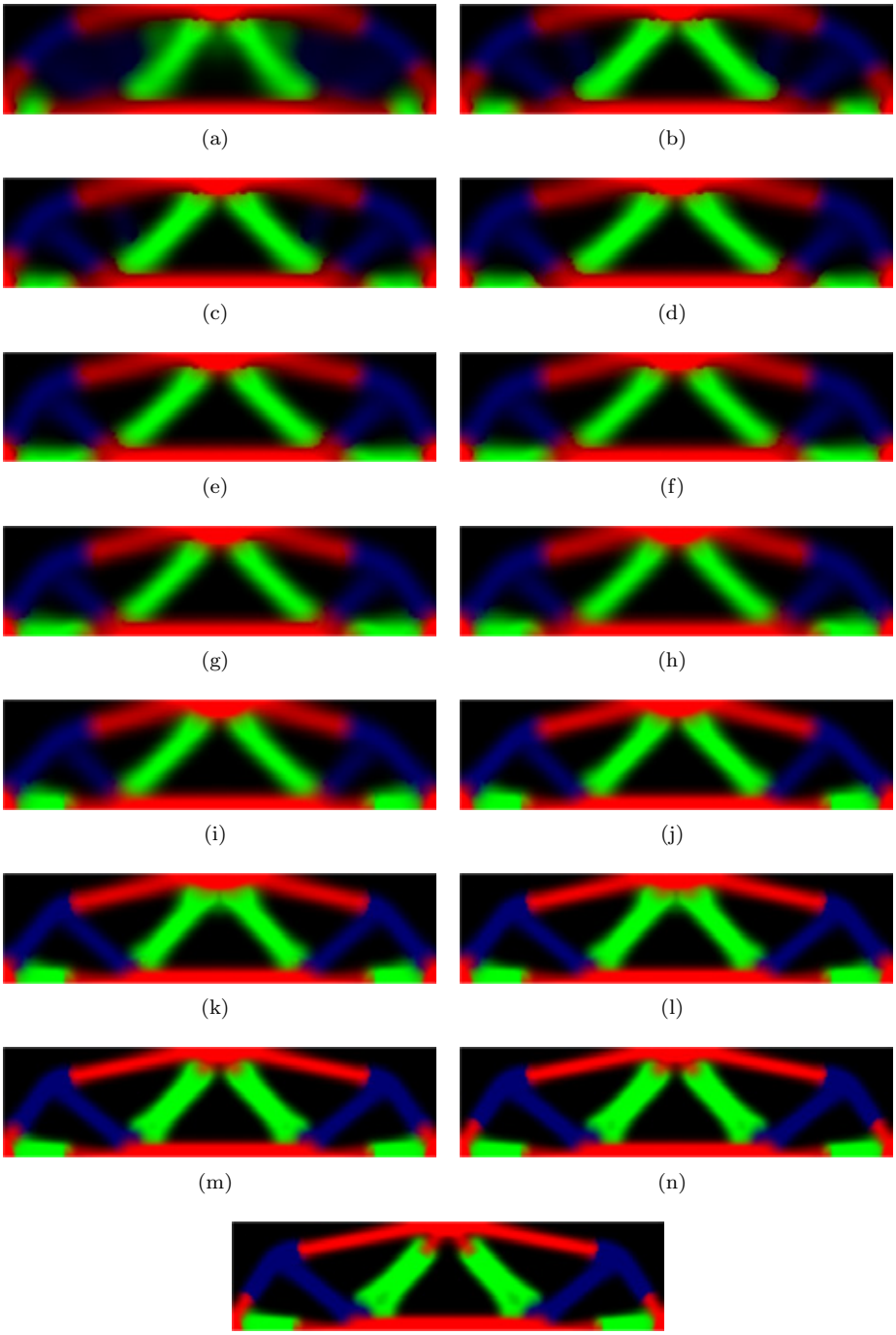


FIGURE 8. Topological changes during the optimization iterations for MBB beam #2 test problem; a-o are respectively related to iterations 5, 10, 15, 20, 30, 40, 50, 60, 80, 100, 120, 140, 160, 180, 200.

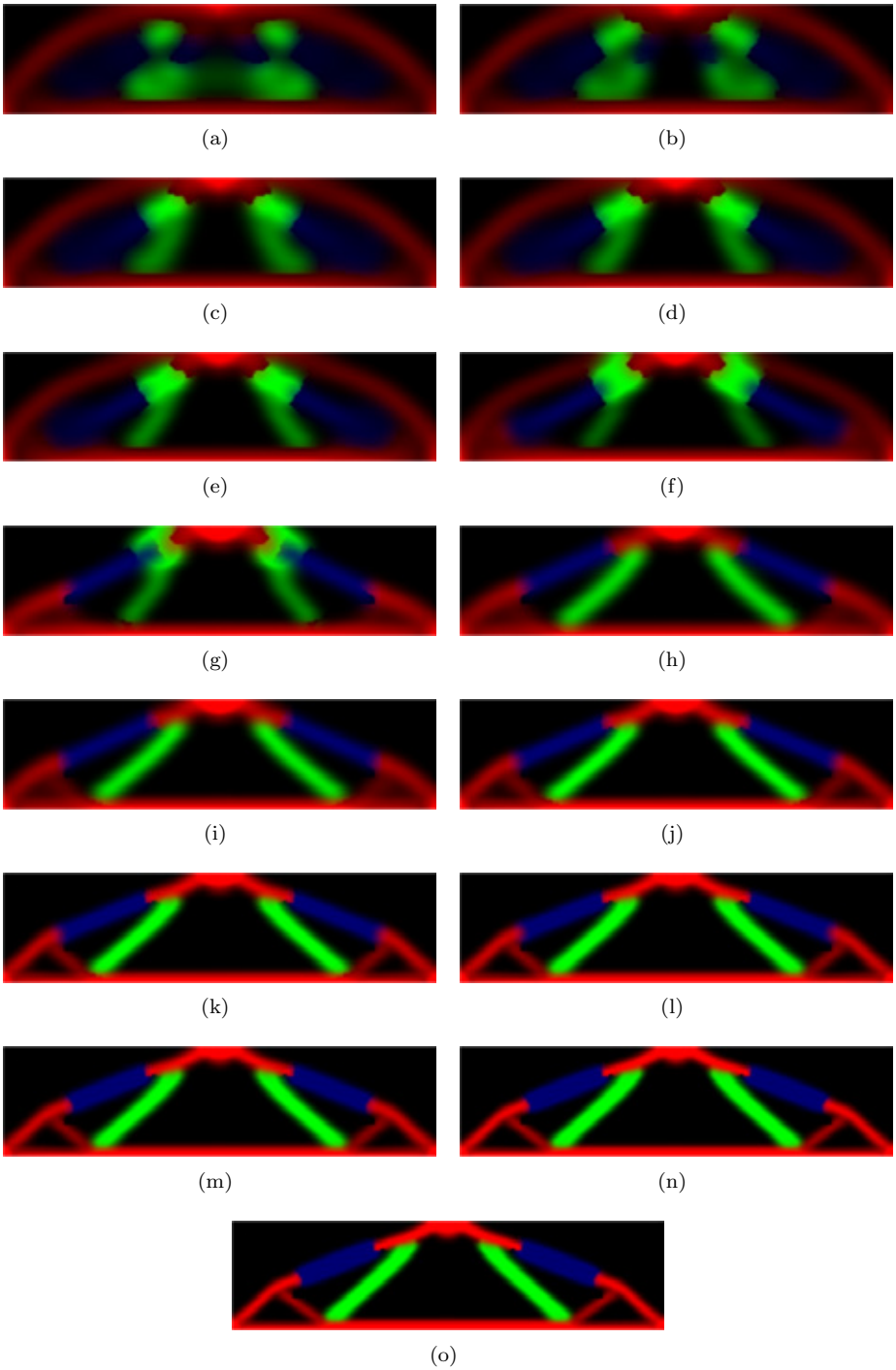
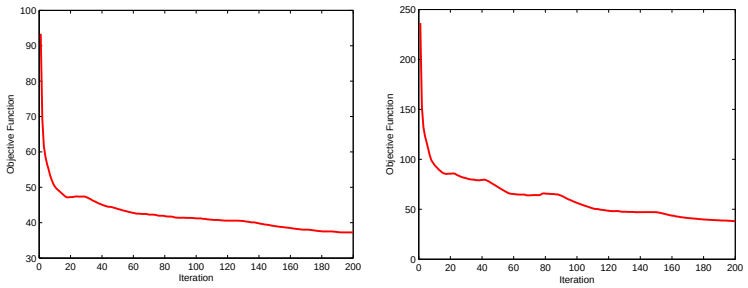
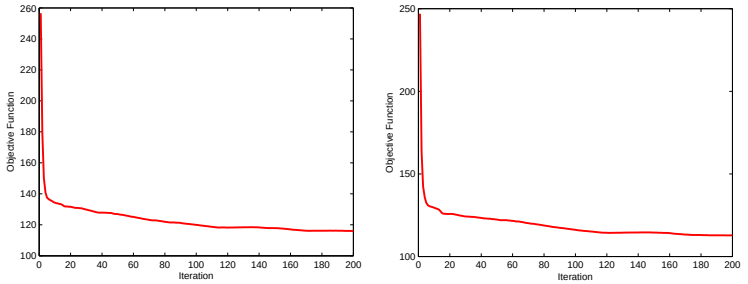


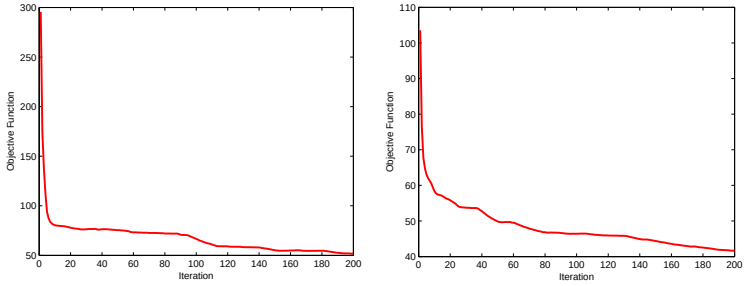
FIGURE 9. Topological changes during the optimization iterations for MBB beam #3 test problem; a-o are respectively related to iterations 5, 10, 15, 20, 30, 40, 50, 60, 80, 100, 120, 140, 160, 180, 200.



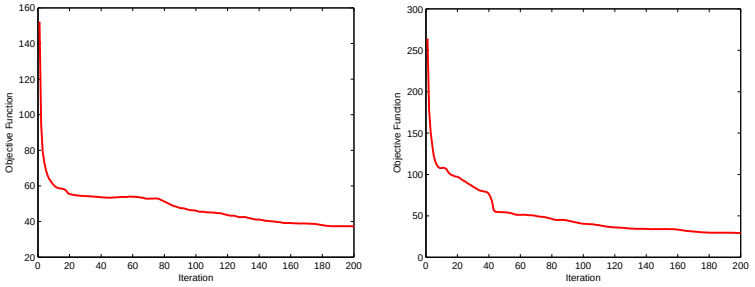
(a) cantilever beam #1 (b) cantilever beam #2



(c) bridge structure #1 (d) bridge structure #2



(e) bridge structure #3 (f) MBB-beam #1



(g) MBB-beam #2 (h) MBB-beam #3

FIGURE 10. The history of objective functional for minimum compliance test problems in this study.

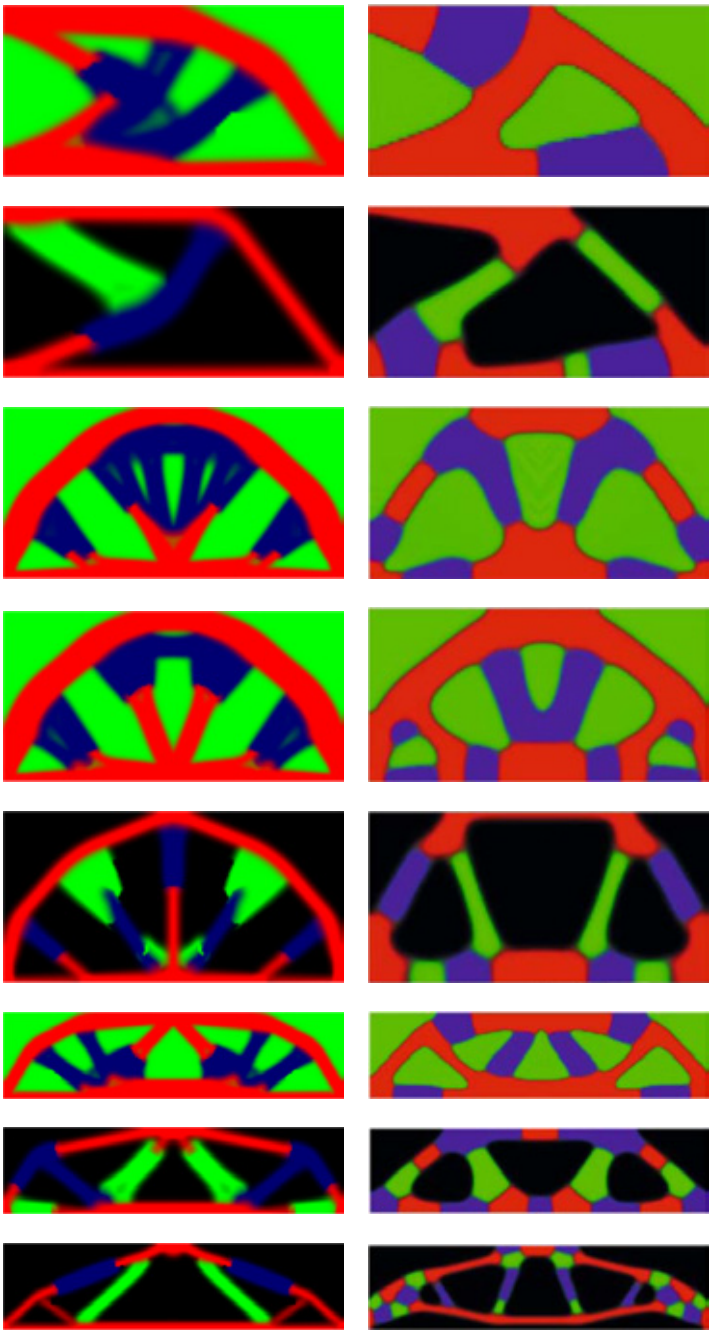


FIGURE 11. Final topologies, current study (left) vs. results of [22] (right); from top to down: cantilever beam #1, #2, bridge structure #1, #2, #3, MBB beam #1, #2, #3.

Discussion. The plots in figures 2 to 9 show the success of the presented multi-phase topology optimization framework. Considering figure 10, the convergence of the presented code is similar to that of original bi-material topology optimization code (note that optimality criteria approach is not a strictly monotone algorithm). This observation illustrates that the presented algorithm inherits the convergence properties of its mother algorithm. Considering the reported CPU times and the number of outer iterations, it appears that the computational cost of the presented algorithm for p phase is about $p(p-1)/2$ times of the computational cost of the corresponding binary phase topology optimization algorithm. Therefore, the computational cost is proportional to p^2. Because the same sets of controlling parameters are used in this section for all cases, the algorithm's parameters are not strongly depend on the type of problem. This observation increases the utility of the presented approach. Comparison of our results with [22] shows that the global outline of final topologies are similar, however, there are significant differences in details of resulted topologies. Comparing the structural performance and restructurability of resulted structures appears to be an interesting issue.

7.2. **Minimum thermal compliance topology optimization.** The numerical results corresponding to the model problem mentioned in section 5.2 is presented in this section. For this purpose, two minimum thermal compliance topology optimization problems are considered here; see figure 12. The first problem is adapted from [36] and the second one is adapted from section 5.1.6 of [3]. The design optimization parameters corresponding to thermal test problems are listed in table 3. The symmetries of problems are exploited in our simulation to reduce the computational cost. Note that the values of parameters which are not determined here are taken equal to those of section 7.1 (to show the ability of framework to manage different kind of problems using the same set of controlling parameters).

The history of objective function during the optimization iterations (outer iterations) is plotted in figure 13 for test cases1 #1, #2 and #3. Except for the case 2 #2 (in which the optimization is terminated at `iter_out` = `673`), the optimization is stopped after `iter_max_out` other iterations in all cases. The variation of filter radius (`rf`) during optimization cycles is plotted in figure 14 for cases 2 #2, #3. The final topology corresponding to the above mentioned test problems, are shown in figure 15. The colors are selected such that the conductivity decreases based on the following order: red, blue, green, black and white. The consumed CPU time corresponding to these experiments are listed in table 4.

Discussion. Similar to the previous section, the results of this section confirm the convergence and success of the presented algorithm. Considering the results of case 2 #1, the algorithm did not converge to stopping criteria based on the minimal change in the resulted topology during outer cycles. Checking the topological changes at the later iterations in this case shows the oscillatory behavior in the objective function values (in fact algorithm does not proceed toward a better outcome by additional iterations). This observation suggests to consider a more robust convergence criteria to improve the computational performance. According to Figure 14, the filter radius decreases monotonically during the optimization iterations. According to our numerical experiment, using a better strategy for updating filter radius could improve the convergence speed and also the termination criteria.

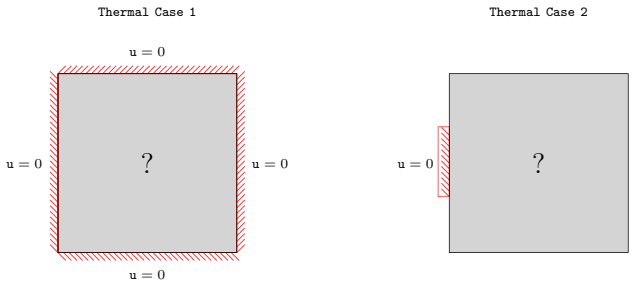


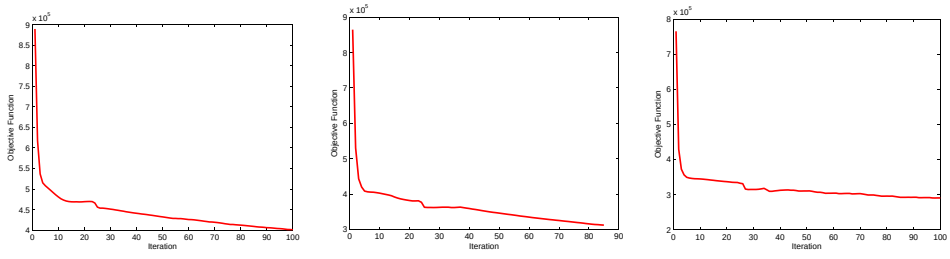
FIGURE 12. Geometries and boundary conditions corresponding to minimum thermal compliance model problems .

TABLE 3. Design parameters of minimum thermal compliance topology optimization test cases.

Problem	nx	ny	p	e	v	iter_max_out
Case1 #1	50	50	3	[4 2 1]′	[0.35 0.35 0.3]′	100
Case1 #2	50	50	4	[6 4 2 1]′	[0.25 0.25 0.25 0.25]′	100
Case1 #3	50	50	5	[8 6 4 2 1]′	[0.2 0.2 0.2 0.2 0.2]′	100
Case2 #1	100	50	3	[1.e4 1.e3 1]′	[0.25 0.25 0.5]′	1000
Case2 #2	100	50	4	[1.e4 5.e3 1.e3 1]′	[0.2 0.15 0.15 0.5]′	1000
Case2 #3	100	50	5	[1.e3 4.e2 2.e2 1.e2 1]′	[0.2 0.1 0.1 0.1 0.5]′	400

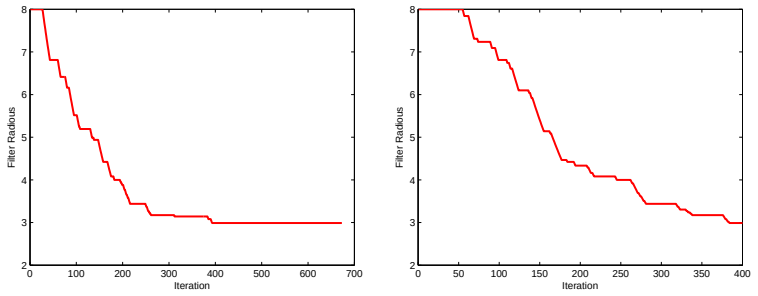
TABLE 4. Minimum thermal compliance topology optimization test cases: CPU times in second.

Case1#1	Case1#2	Case1#3	Case2#1	Case2#2	Case2#3
38	61	87	730	731	781



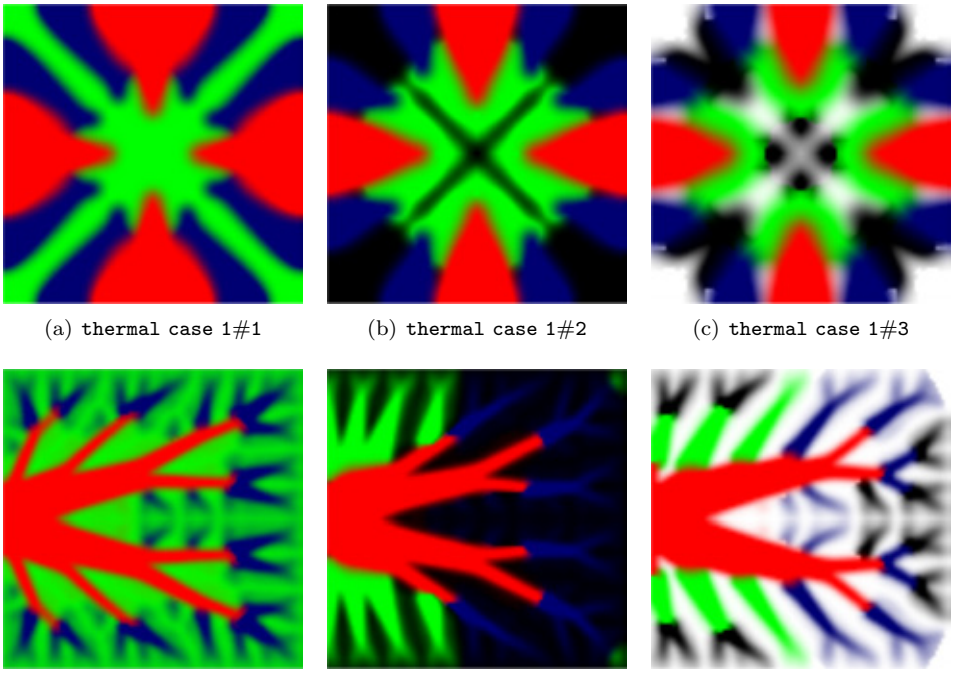
(a) thermal case 1#1 (b) thermal case 1#2 (c) thermal case 1#3

FIGURE 13. The history of objective functional for thermal problem 1 in this study.



(a) `thermal case 2#2` (b) `thermal case 2#3`

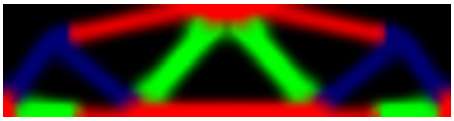
FIGURE 14. The variation of filter radius during minimum thermal compliance topology optimization cycles for cases 2#2 and 2#3.



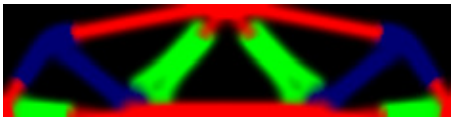
(a) `thermal case 1#1` (b) `thermal case 1#2` (c) `thermal case 1#3`

(d) `thermal case 2#1` (e) `thermal case 2#2` (f) `thermal case 2#3`

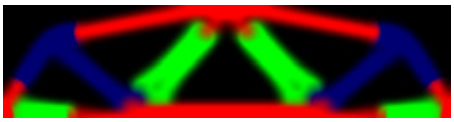
FIGURE 15. The final topology corresponding to the minimum thermal compliance test cases.



(a) 48×24



(b) 96×48



(c) 192×96

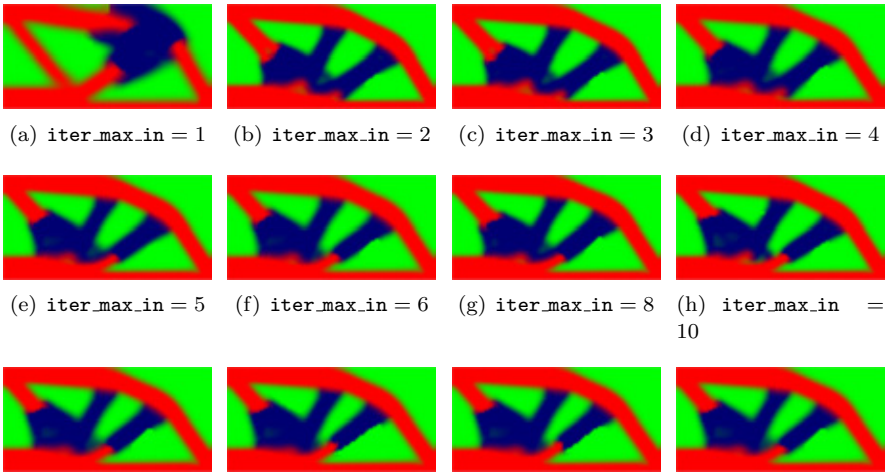
FIGURE 16. Final topology of $\texttt{MBB}-\texttt{Beam}$ #2 test problem on different grid resolution.

7.3. **Grid resolution study.** To study the dependence of numerical results to the mesh size, the solution of $\texttt{MBB}-\texttt{Beam}$ #2 problem is considered here with three different mesh resolutions: 48×24, 96×48 and 192×96. Because the filter parameter `rf` controls the length scale in our algorithm, it is selected based on the grid resolution; $\texttt{rf} = 4, 8, 16$ for these three resolutions respectively. Other parameters are similar to $\texttt{MBB}-\texttt{Beam}$ #2 test case in section 7.1. Results of this experiment are shown in figure 16. Note that for the grid resolution 48×24, the algorithm terminated at iteration 178, because of meeting the convergence criterion.

Discussion. According to figure 16, the final topology is almost insensitive to the resolution of computational grid, using a consistent length-scale control.

7.4. **Sensitivity to the maximum inner iterations (`iter_max_in`).** The effect of maximum number of inner iterations, `iter_max_in`, on the performance and result of the presented algorithm is numerically studied in this section. For this purpose, the solution of `Cantilever Beam` #1 with different `iter_max_in` is considered here ($\texttt{iter\_max\_in} = 1, 2, \ldots, 20$). Figure 17 shows the results of this numerical experiment. The variation of final value of objective function (after 200 outer iterations) by `iter_max_in` is plotted in figure 18.

Discussion. According to figure 17, the final topology is almost insensitive to the variation of `iter_max_in` for $\texttt{iter\_max\_in} \geqslant 2$. Considering figure 18, increasing the value of `iter_max_in` to 20 not only does not improve the convergence of outer iterations but also slightly decreases the overall convergence rate. It is while the computational cost increases linearly with `iter_max_in`. Therefore, there is an optimal value for `iter_max_in` parameter. According to our numerical experiments in this study $\texttt{iter\_max\_in} = 2$ is almost an optimal choice.



(a) `iter_max_in` $= 1$ (b) `iter_max_in` $= 2$ (c) `iter_max_in` $= 3$ (d) `iter_max_in` $= 4$

(e) `iter_max_in` $= 5$ (f) `iter_max_in` $= 6$ (g) `iter_max_in` $= 8$ (h) `iter_max_in` $= 10$

(i) `iter_max_in` $= 12$ (j) `iter_max_in` $= 14$ (k) `iter_max_in` $= 16$ (l) `iter_max_in` $= 20$

FIGURE 17. Effect of maximum number of inner iterations on the final topology initial for `Cantilever Beam #1`.

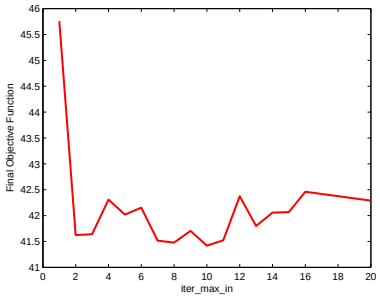
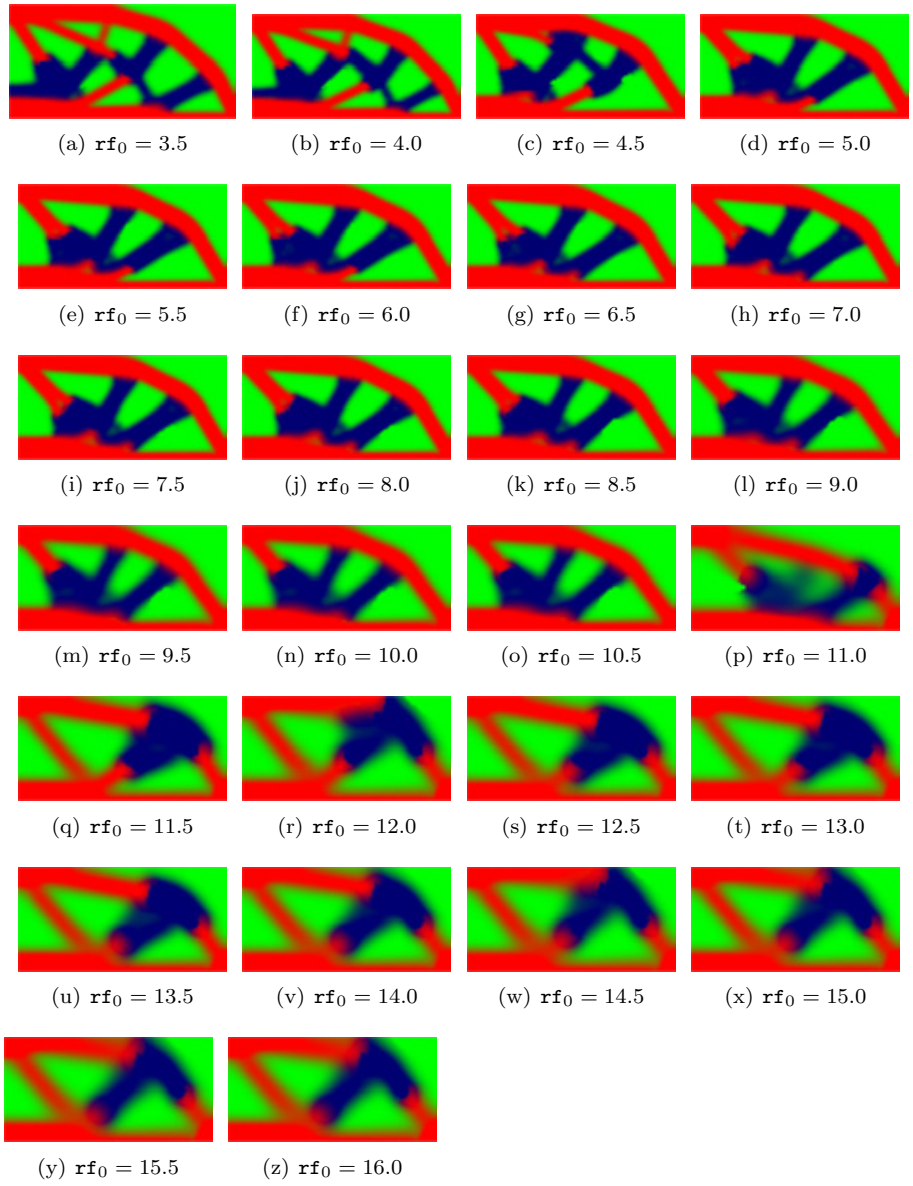


FIGURE 18. The variation of final value of objective function (after 200 outer iterations) as a function of maximum number of inner iterations, `iter_max_in`.

7.5. **Sensitivity to the filter radius.** To study the effect of initial filter radius on the final topology, we consider `Cantilever Beam #1` with different initial filter radius, $\texttt{rf}_0$, ranging from 3.5 to 16.0 with step size 0.5. Figure 19 shows the results of this experiment (in all cases the optimization is run for 200 outer iterations).

Discussion. Figure 19, shows that the outline of final topology varies significantly by the variation of initial filter radius. Note that the initial filter radius does not directly correlate to the minimum length scale in the structure, because $\texttt{rf}_0$ dynamically varies during the optimization. This observation implies that we could produce different (local) solutions using different sets of controlling parameters.



(a) $\texttt{rf}_0 = 3.5$ (b) $\texttt{rf}_0 = 4.0$ (c) $\texttt{rf}_0 = 4.5$ (d) $\texttt{rf}_0 = 5.0$

(e) $\texttt{rf}_0 = 5.5$ (f) $\texttt{rf}_0 = 6.0$ (g) $\texttt{rf}_0 = 6.5$ (h) $\texttt{rf}_0 = 7.0$

(i) $\texttt{rf}_0 = 7.5$ (j) $\texttt{rf}_0 = 8.0$ (k) $\texttt{rf}_0 = 8.5$ (l) $\texttt{rf}_0 = 9.0$

(m) $\texttt{rf}_0 = 9.5$ (n) $\texttt{rf}_0 = 10.0$ (o) $\texttt{rf}_0 = 10.5$ (p) $\texttt{rf}_0 = 11.0$

(q) $\texttt{rf}_0 = 11.5$ (r) $\texttt{rf}_0 = 12.0$ (s) $\texttt{rf}_0 = 12.5$ (t) $\texttt{rf}_0 = 13.0$

(u) $\texttt{rf}_0 = 13.5$ (v) $\texttt{rf}_0 = 14.0$ (w) $\texttt{rf}_0 = 14.5$ (x) $\texttt{rf}_0 = 15.0$

(y) $\texttt{rf}_0 = 15.5$ (z) $\texttt{rf}_0 = 16.0$

FIGURE 19. Effect of initial filter radius on results for `Cantilever Beam #1`.

8. Summary

Combining the classical binary phase topology optimization algorithm and the (Gauss-Seidel-like) block coordinate descent algorithm, a general framework is introduced to solve multimaterials topology optimization problems. The algorithm is easy to implement and almost every binary phase topology optimization solver could be easily extended to its multiphase counterpart using the presented framework; such that the resulted solver inherits the convergence properties of its original solver. The overall algorithmic complexity and coding of the presented approach is independent of the number of contributing phases. Using the presented framework, the classical optimality criteria based binary phase minimum compliance topology optimization MATLAB code is extended to solve multiphase thermal and structural topology optimization problems. The success and efficiency of the presented algorithm is illustrated numerically by solution of several test problems.

References

[1] M.P. Bendsøe and N. Kikuchi. Generating optimal topologies in structural design using a homogenization method. *Comput Meth Appl Mech Engng*, 71(2):197–224, 1988.

[2] M.P. Bendsøe. *Optimization of structural topology, shape, and material*. Springer Verlag, 1995.

[3] M.P. Bendsøe and O. Sigmund. *Topology Optimization: theory, methods and applications*. Springer, 2004.

[4] M.P. Bendsøe and O. Sigmund. Material interpolation schemes in topology optimization. *Arch Appl Mech*, 69(9):635–654, 1999.

[5] G. Allaire. Shape optimization by the homogenization method. *New York, Springer-Verlag, 2002.*, 2002.

[6] O. Sigmund and S. Torquato. Composites with extremal thermal expansion coefficients. *Appl Phys Lett*, 69(21):3203–3205, 1996.

[7] O. Sigmund and S. Torquato. Design of materials with extreme thermal expansion using a three-phase topology optimization method. *J Mech Phys Solids*, 45(6):1037–1067, 1997.

[8] O. Sigmund and S. Torquato. Design of smart composite materials using topology optimization. *Smart Materials and Structures*, 8:365, 1999.

[9] L.V. Gibiansky and O. Sigmund. Multiphase composites with extremal bulk modulus. *J Mech Phys Solids*, 48(3):461–498, 2000.

[10] O. Sigmund. Recent developments in extremal material design. *in Trends in Computational Mechanics, W.A. Wall, K.-U. Bletzinger and K. Schweizerhof (eds)*, pages 228–232, 2001.

[11] R. Tavakoli and H. Zhang. A nonmonotone spectral projected gradient method for large-scale topology optimization problems. *Numerical Algebra, Control and Optimization*, 2(2):395–412, 2012.

[12] B. Bourdin and A. Chambolle. Design-dependent loads in topology optimization. *ESAIM COCV*, 9:19–48, 2003.

[13] M.Y. Wang and S. Zhou. Synthesis of shape and topology of multi-material structures with a phase-field method. *J Comput-Aided Mater Des*, 11(2):117–138, 2004.

[14] H.K. Zhao, T. Chan, B. Merriman, and S. Osher. A variational level set approach to multiphase motion. *J Comput Phys*, 127(1):179–195, 1996.

[15] L.A. Vese and T.F. Chan. A multiphase level set framework for image segmentation using the mumford and shah model. *Int J Comput Vis*, 50(3):271–293, 2002.

[16] M. Yulin and W. Xiaoming. A level set method for structural topology optimization and its applications. *Adv Eng Softw*, 35(7):415–441, 2004.

[17] M.Y. Wang and X. Wang. color level sets: a multi-phase method for structural topology optimization with multiple materials. *Comput Meth Appl Mech Engng*, 193(6):469–496, 2004.
[18] M.Y. Wang and X. Wang. A level-set based variational method for design and optimization of heterogeneous objects. *Computer-Aided Design*, 37(3):321–337, 2005.
[19] P. Wei and M.Y. Wang. Piecewise constant level set method for structural topology optimization. *Int J Numer Methods Eng*, 78(4):379–402, 2009.
[20] Z. Luo, L. Tong, J. Luo, P. Wei, and M.Y. Wang. Design of piezoelectric actuators using a multiphase level set method of piecewise constants. *J Comput Phys*, 228(7):2643–2659, 2009.
[21] S. Zhou and M.Y. Wang. 3d multi-material structural topology optimization with the generalized cahn-hilliard equations. *CMES: Comput Model in Eng Sci*, 16(2):83–102, 2006.
[22] S. Zhou and M.Y. Wang. Multimaterial structural topology optimization with a generalized cahn–hilliard model of multiphase transition. *Struct Multidisc Optim*, 33(2):89–111, 2007.
[23] X. Huang, M. Xie, et al. *Evolutionary topology optimization of continuum structures: methods and applications*. Wiley, 2010.
[24] X. Huang and YM Xie. Bi-directional evolutionary topology optimization of continuum structures with one or multiple materials. *Comput Mech*, 43(3):393–401, 2009.
[25] Z. Hashin and S. Shtrikman. A variational approach to the theory of the elastic behaviour of multiphase materials. *J Mech Phys Solids*, 11(2):127–140, 1963.
[26] O. Sigmund and J. Petersson. Numerical instabilities in topology optimization: a survey on procedures dealing with checkerboards, mesh-dependencies and local minima. *Struct Multidisc Optim*, 16(1):68–75, 1998.
[27] W.I. Zangwill. *Nonlinear programming: a unified approach*. Prentice-Hall Englewood Cliffs, NJ, 1969.
[28] D.G. Luenberger and Y. Ye. *Linear and nonlinear programming, third edition*. Springer Verlag, 2008.
[29] JC Bezdek, RJ Hathaway, RE Howard, CA Wilson, and MP Windham. Local convergence analysis of a grouped variable version of coordinate descent. *J Optim Theory Appl*, 54(3):471–477, 1987.
[30] P. Tseng. Convergence of a block coordinate descent method for nondifferentiable minimization. *J Optim Theory Appl*, 109(3):475–494, 2001.
[31] C.J. Lin, S. Lucidi, L. Palagi, A. Risi, and M. Sciandrone. Decomposition algorithm model for singly linearly-constrained problems subject to lower and upper bounds. *J Optim Theory Appl*, 141(1):107–126, 2009.
[32] P. Tseng and S. Yun. A coordinate gradient descent method for linearly constrained smooth optimization and support vector machines training. *Comput Optim Appl*, 47(2):179–206, 2010.
[33] G. Liuzzi, L. Palagi, and M. Piacentini. On the convergence of a jacobi-type algorithm for singly linearly-constrained problems subject to simple bounds. *Optim Lett*, 5(2):347–362, 2011.
[34] O. Sigmund. A 99 line topology optimization code written in matlab. *Struct Multidisc Optim*, 21(2):120–127, 2001.
[35] E. Andreassen, A. Clausen, M. Schevenels, B.S. Lazarov, and O. Sigmund. Efficient topology optimization in matlab using 88 lines of code. *Struct Multidisc Optim*, 43(1):1–16, 2011.
[36] Alberto Donoso and Pablo Pedregal. Optimal design of 2d conducting graded materials by minimizing quadratic functionals in the field. *Struct Multidisc Optim*, 30(5):360–367, 2005.

Supplement A. 115-line MATLAB code for multimaterials minimum compliance topology optimization

The following code includes the 115 lines MATLAB code for the solution of multimaterials minimum compliance topology optimization problem. The loading and boundary conditions are related to MBB-beam case in which only one half of domain is considered (the visualization is performed for whole beam).

```
%% 115 LINES MATLAB CODE MULTIPHASE MINIMUM COMPLIANCE TOPOLOGY OPTIMIZATION
function multitop(nx,ny,tol_out,tol_f,iter_max_in,iter_max_out,p,q,e,v,rf)
 alpha = zeros(nx*ny,p);
 for i = 1:p
  alpha(:,i) = v(i);
 end
 %% MAKE FILTER
 [H,Hs] = make_filter (nx,ny,rf);
 change_out = 2*tol_out; iter_out = 0;
 while (iter_out < iter_max_out) && (change_out > tol_out)
  alpha_old = alpha;
  for a = 1:p
   for b = a+1:p
    [obj,alpha] = bi_top(a,b,nx,ny,p,v,e,q,alpha,H,Hs,iter_max_in);
   end
  end
  iter_out = iter_out + 1;
  change_out = norm(alpha(:)-alpha_old(:),inf);
  fprintf('Iter:%5i Obj.:%11.4f change:%10.8f\n',iter_out,obj,change_out);
  %% UPDATE FILTER
  if (change_out < tol_f) && (rf>3)
   tol_f = 0.99*tol_f; rf = 0.99*rf; [H,Hs] = make_filter (nx,ny,rf);
  end
  %% SCREEN OUT TEMPORAL TOPOLOGY EVERY 5 ITERATIONS
  if mod(iter_out,5)==0
   I = make_bitmap (p,nx,ny,alpha);
   image([flipdim(I ,2) I]), axis image off, drawnow;
  end
 end
end
%% MAKE FILTER
function [H,Hs] = make_filter (nx,ny,rmin)
 ir = ceil(rmin)-1;
 iH = ones(nx*ny*(2*ir+1)^2,1);
 jH = ones(size(iH)); sH = zeros(size(iH)); k = 0;
 for i1 = 1:nx
   for j1 = 1:ny
     e1 = (i1-1)*ny+j1;
     for i2 = max(i1-ir,1):min(i1+ir,nx)
       for j2 = max(j1-ir,1):min(j1+ir,ny)
         e2 = (i2-1)*ny+j2; k = k+1; iH(k) = e1; jH(k) = e2;
         sH(k) = max(0,rmin-sqrt((i1-i2)^2+(j1-j2)^2));
       end
     end
   end
 end
 H = sparse(iH,jH,sH); Hs = sum(H,2);
end
%% MODIFIED BINARY-PHASE TOPOLOGY OPTIMIZATION SOLVER
```

```
function [o,alpha] = bi_top(a,b,nx,ny,p,v,e,q,alpha_old,H,Hs,iter_max_in)
 alpha = alpha_old; iter_in = 0; nu = 0.3;
 %% PREPARE FINITE ELEMENT ANALYSIS
 A11 = [12 3 -6 -3; 3 12 3 0; -6 3 12 -3; -3 0 -3 12];
 A12 = [-6 -3 0 3; -3 -6 -3 -6; 0 -3 -6 3; 3 -6 3 -6];
 B11 = [-4 3 -2 9; 3 -4 -9 4; -2 -9 -4 -3; 9 4 -3 -4];
 B12 = [ 2 -3 4 -9; -3 2 9 -2; 4 9 2 3; -9 -2 3 2];
 KE = 1/(1-nu^2)/24*([A11 A12;A12' A11]+nu*[B11 B12;B12' B11]);
 nodenrs = reshape(1:(1+nx)*(1+ny),1+ny,1+nx);
 edofVec = reshape(2*nodenrs(1:end-1,1:end-1)+1,nx*ny,1);
 edofMat = repmat(edofVec,1,8)+repmat([0 1 2*ny+[2 3 0 1] -2 -1],nx*ny,1);
 iK = reshape(kron(edofMat,ones(8,1))',64*nx*ny,1);
 jK = reshape(kron(edofMat,ones(1,8))',64*nx*ny,1);
 %% DEFINE LOADS AND SUPPORTS (HALF MBB-BEAM)
 F = sparse(2,1,-1,2*(ny+1)*(nx+1),1); % MBB
 fixeddofs = union([1:2:2*(ny+1)],[2*(nx+1)*(ny+1)]);
 U = zeros(2*(ny+1)*(nx+1),1);
 alldofs = [1:2*(ny+1)*(nx+1)];
 freedofs = setdiff(alldofs,fixeddofs);
 %% INNER ITERATIONS
 while iter_in < iter_max_in
  iter_in = iter_in + 1;
  %% FE-ANALYSIS
  E = e(1)*alpha(:,1).^q;
  for phase = 2:p
   E = E + e(phase)*alpha(:,phase).^q;
  end
  sK = reshape(KE(:)*E(:)',64*nx*ny,1);
  K = sparse(iK,jK,sK); K = (K+K')/2;
  U(freedofs) = K(freedofs,freedofs)\F(freedofs);
  %% OBJECTIVE FUNCTION AND SENSITIVITY ANALYSIS
  ce = sum((U(edofMat)*KE).*U(edofMat),2);
  o = sum(sum(E.*ce));
  dc = -(q*(e(a)-e(b))*alpha(:,a).^(q-1)).*ce;
  %% FILTERING OF SENSITIVITIES
  dc = H*(alpha(:,a).*dc)./Hs./max(1e-3,alpha(:,a)); dc = min(dc,0);
  %% UPDATE LOWER AND UPPER BOUNDS OF DESIGN VARIABLES
  move = 0.2;
  r = ones(nx*ny,1);
  for k = 1:p
   if (k ~= a) && (k ~= b)
    r = r - alpha(:,k);
   end
  end
  l = max(0,alpha(:,a)-move);
  u = min(r,alpha(:,a)+move);
  %% OPTIMALITY CRITERIA UPDATE OF DESIGN VARIABLES
  l1 = 0; l2 = 1e9;
  while (l2-l1)/(l1+l2) > 1e-3
   lmid = 0.5*(l2+l1);
   alpha_a = max(l,min(u,alpha(:,a).*sqrt(-dc./lmid)));
   if sum(alpha_a) > nx*ny*v(a); l1 = lmid; else l2 = lmid; end
  end
  alpha(:,a) = alpha_a;
  alpha(:,b) = r-alpha_a;
 end
end
%% MAKE BITMAP IMAGE OF MULTIPHASE TOPOLOGY
```

```
function I = make_bitmap (p,nx,ny,alpha)
 color = [1 0 0; 0 0 .45; 0 1 0; 0 0 0; 1 1 1];
 I = zeros(nx*ny,3);
 for j = 1:p
  I(:,1:3) = I(:,1:3) + alpha(:,j)*color(j,1:3);
 end
 I = imresize(reshape(I,ny,nx,3),10,'bilinear');
end
```

The following pieces of MATLAB code is related to parameter set and main functions to solve MBB − Beam #3 test problem.

```
function [nx,ny,tol,tolf,im_in,im_out,p,q,e,v,rf] = set_parameters ()
  nx = 96; ny = 48; tol = 0.001; tolf = 0.05; im_in = 2; im_out = 200;
  p = 4; q = 3; e = [9 3 1 1e-9]'; v = [0.16 0.08 0.08 0.68]'; rf = 8;
end
```

```
function main
 [nx,ny,tol_out,tol_f,iter_max_in,iter_max_out,p,q,e,v,rf] = set_parameters ();
 multitop(nx,ny,tol_out,tol_f,iter_max_in,iter_max_out,p,q,e,v,rf);
end
```

Supplement B. 115-line MATLAB code for multimaterials minimum thermal compliance topology optimization

The following code includes the 115 lines MATLAB code for the solution of multimaterials minimum thermal compliance topology optimization problem. The loading and boundary conditions are related to case #2 of our thermal test problems.

```
%% 115 LINES MATLAB CODE MULTIPHASE THERMAL TOPOLOGY OPTIMIZATION
function multitop_h(nx,ny,tol_out,tol_f,iter_max_in,iter_max_out,p,q,e,v,rf)
 alpha = zeros(nx*ny,p);
 for i = 1:p
  alpha(:,i) = v(i);
 end
 %% MAKE FILTER
 [H,Hs] = make_filter (nx,ny,rf);
 change_out = 2*tol_out; iter_out = 0;
 while (iter_out < iter_max_out) && (change_out > tol_out)
  alpha_old = alpha;
  for a = 1:p
   for b = a+1:p
    [obj,alpha] = bi_top_h(a,b,nx,ny,p,v,e,q,alpha,H,Hs,iter_max_in);
   end
  end
  iter_out = iter_out + 1;
  change_out = norm(alpha(:)-alpha_old(:),inf);
  fprintf('Iter:%5i Obj.:%11.4f change:%10.8f\n',iter_out,obj,change_out);
  %% UPDATE FILTER
  if (change_out < tol_f) && (rf>3)
   tol_f = 0.99*tol_f; rf = 0.99*rf; [H,Hs] = make_filter (nx,ny,rf);
  end
  %% SCREEN OUT TEMPORAL TOPOLOGY EVERY 5 ITERATIONS
```

```
  if (mod(iter_out,5)==0)
    I = make_bitmap (p,nx,ny,alpha);
    I = [flipdim(I,1); I]; image(I), axis image off, drawnow;
  end
 end
end
%% MAKE FILTER
function [H,Hs] = make_filter (nx,ny,rmin)
 ir = ceil(rmin)-1;
 iH = ones(nx*ny*(2*ir+1)^2,1);
 jH = ones(size(iH));
 sH = zeros(size(iH));
 k = 0;
 for i1 = 1:nx
   for j1 = 1:ny
     e1 = (i1-1)*ny+j1;
     for i2 = max(i1-ir,1):min(i1+ir,nx)
       for j2 = max(j1-ir,1):min(j1+ir,ny)
         e2 = (i2-1)*ny+j2; k = k+1; iH(k) = e1; jH(k) = e2;
         sH(k) = max(0,rmin-sqrt((i1-i2)^2+(j1-j2)^2));
       end
     end
   end
 end
 H = sparse(iH,jH,sH); Hs = sum(H,2);
end
%% MODIFIED BINARY-PHASE TOPOLOGY OPTIMIZATION SOLVER
function [o,alpha] = bi_top_h(a,b,nx,ny,p,v,e,q,alpha_old,H,Hs,iter_max_in)
 alpha = alpha_old; iter_in = 0;
 %% PREPARE FINITE ELEMENT ANALYSIS
 KE = [2/3 -1/6 -1/3 -1/6; -1/6 2/3 -1/6 -1/3; ...
       -1/3 -1/6 2/3 -1/6; -1/6 -1/3 -1/6 2/3];
 nodenrs = reshape(1:(1+nx)*(1+ny),1+ny,1+nx);
 edofVec = reshape(nodenrs(1:end-1,1:end-1)+1,nx*ny,1);
 edofMat = repmat(edofVec,1,4)+repmat([0 ny+[1 0] -1],nx*ny,1);
 iK = reshape(kron(edofMat,ones(4,1))',16*nx*ny,1);
 jK = reshape(kron(edofMat,ones(1,4))',16*nx*ny,1);
 %% DEFINE LOADS AND SUPPORTS
 F = sparse((ny+1)*(nx+1),1); F(:,1) = 1;
 %fixeddofs = union([1:ny+1],[(ny+1):(ny+1):(nx+1)*(ny+1)]); %case1
 fixeddofs = [1:5]; % case 2
 U = zeros((ny+1)*(nx+1),1);
 alldofs = [1:(ny+1)*(nx+1)];
 freedofs = setdiff(alldofs,fixeddofs);
 %% INNER ITERATIONS
 while iter_in < iter_max_in
  iter_in = iter_in + 1;
  %% FE-ANALYSIS
  E = e(1)*alpha(:,1).^q;
  for phase = 2:p
   E = E + e(phase)*alpha(:,phase).^q;
  end
  sK = reshape(KE(:)*E(:)',16*nx*ny,1);
  K = sparse(iK,jK,sK); K = (K+K')/2;
  U(freedofs) = K(freedofs,freedofs)\F(freedofs);
  %% OBJECTIVE FUNCTION AND SENSITIVITY ANALYSIS
  ce = sum((U(edofMat)*KE).*U(edofMat),2);
  o = sum(sum(E.*ce));
```

```
  dc = -(q*(e(a)-e(b))*alpha(:,a).^(q-1)).*ce;
  %% FILTERING OF SENSITIVITIES
  dc = H*(alpha(:,a).*dc)./Hs./max(1e-3,alpha(:,a)); dc = min(dc,0);
  %% UPDATE LOWER AND UPPER BOUNDS OF DESIGN VARIABLES
  move = 0.2;
  r = ones(nx*ny,1);
  for k = 1:p
   if (k ~= a) && (k ~= b)
    r = r - alpha(:,k);
   end
  end
  l = max(0,alpha(:,a)-move);
  u = min(r,alpha(:,a)+move);
  %% OPTIMALITY CRITERIA UPDATE OF DESIGN VARIABLES
  l1 = 0; l2 = 1e9;
  while (l2-l1)/(l1+l2) > 1e-3
   lmid = 0.5*(l2+l1);
   alpha_a = max(l,min(u,alpha(:,a).*sqrt(-dc./lmid)));
   if sum(alpha_a) > nx*ny*v(a); l1 = lmid; else l2 = lmid; end
  end
  alpha(:,a) = alpha_a;
  alpha(:,b) = r-alpha_a;
 end
end
%% MAKE BITMAP IMAGE OF MULTIPHASE TOPOLOGY
function I = make_bitmap (p,nx,ny,alpha)
 color = [1 0 0; 0 0 .45; 0 1 0; 0 0 0; 1 1 1];
 I = zeros(nx*ny,3);
 for j = 1:p
  I(:,1:3) = I(:,1:3) + alpha(:,j)*color(j,1:3);
 end
 I = imresize(reshape(I,ny,nx,3),10,'bilinear');
end
```

The following pieces of MATLAB code is related to parameter set and main functions to solve `Case2` #3 test problem.

```
function [nx,ny,tol,tolf,im_in,im_out,p,q,e,v,rf] = set_parameters ()
 nx = 100; ny = 50; tol = 0.001; tolf = 0.05; im_in = 2; im_out = 400;
 p = 5; q = 3; e = [1000 400 200 100 1]'; v = [0.2 0.1 0.1 0.1 0.5]'; rf = 8;
end
```

```
function main_h
 [nx,ny,tol_out,tol_f,iter_max_in,iter_max_out,p,q,e,v,rf] = set_parameters ();
 multitop_h(nx,ny,tol_out,tol_f,iter_max_in,iter_max_out,p,q,e,v,rf);
end
```