J. Comp. App. Res. Mech. Eng. Vol. X. No. X, pp. X-X, XXXX *DOI: 10.22061/jcarme.XXXX.XXXX.XXXX*

Journal of Computational and Applied Research in Mechanical Engineering

jcarme.sru.ac.ir

SRTTU

JCARME

ISSN: 2228-7922

Research paper

Topology optimization using the SIMP model and Level Set method for two-dimensional structural design algorithms

Paulo Henrique Lixandrão Fernando[a*], Kelly Cristina de Lira Lixandrão[b], Daniel Pezza Tchernov[c], Mauro Machado de Oliveira[c] and Francisco Yastami Nakamoto[c]

[a]*Department of Mechanical Engineering, Santo André Foundation University Center (FSA), Santo André, São Paulo, Brazil*
[b]*Department of Education Management, National Service for Industrial Training (SENAI), São Paulo, Brazil*
[c]*Department of Mechanical Engineering, Federal Institute of São Paulo (IFSP), São Paulo, Brazil*

Article info:

Article history:

Received:	06/17/2026
Revised:	11/13/2026
Accepted:	11/01/2026
Online:	00/00/0000

Keywords:
Finite element method
MATLAB®,
SIMP,
Structural optimization,
Topology optimization.

[*]Corresponding author:
paulo.fernando@fsa.br

Abstract

The increasing demand to reduce project development costs has placed significant pressure on product and process engineers. In response, engineers and designers are increasingly adopting advanced techniques from other disciplines—particularly computational optimization methods—to enhance the design of mechanical components. One such method gaining prominence is topology optimization, a powerful approach within the broader field of structural optimization. Topology optimization (TO) involves the strategic distribution of material within a given design space to achieve optimal structural performance while minimizing weight and material usage. Two of the most widely used methodologies in this domain are the SIMP (solid isotropic material with penalization) model, which is based on material properties, and the level set (LS) method, which is rooted in geometric representation, including the prescribed supports and loading conditions representative of a bridge-like model. Topology optimization provides a systematic framework for achieving lightweight, high-performance designs tailored to specific applications. This article focuses on identifying and evaluating topology optimization strategies based on the SIMP and LS methods. It also explores the fundamental principles of topology optimization and the classification of structural optimization techniques. For practical implementation, the MATLAB® platform (MathWorks) was used. However, the scope of the study is limited to the selected bridge-type configuration, boundary conditions, and two-dimensional numerical models. Therefore, the findings should not be generalized to other structural configurations or considered as a formal validation of the methods.

1. Introduction

The investigation of appropriate topology optimization techniques for specific engineering applications has been extensively explored by numerous international research groups. Among the most widely adopted methods over the past decade are the SIMP model and the LS method [1-3].

These approaches, based respectively on material interpolation and geometric contour evolution, offer distinct advantages depending on the characteristics of the design problem.

The choice of topology optimization method significantly influences various aspects of the computational process, including processing speed, robustness, and convergence behavior. Each method presents trade-offs that must be carefully evaluated in relation to the application domain [4, 5].

In the broader context of engineering design, computer-aided engineering (CAE) tools such as the finite element method (FEM)—also known as finite element analysis (FEA)—are commonly employed to support the simulation of physical systems. FEM is a well-established numerical approach used to approximate solutions to partial differential equations (PDEs) and integral equations that describe physical phenomena [6].

In contrast, the topology optimization method (TOM) is a more recent and evolving discipline that has yet to be fully integrated into many national engineering industries. While TOM provides a powerful framework for material distribution and structural efficiency, it also poses certain numerical challenges.

These include mesh dependency, local minima, and instability artifacts such as the "chessboard pattern"—a condition where alternating elements exhibit the presence or absence of material in a non-physical, checkerboard-like distribution.

Topology optimization techniques can generally be categorized as material-based methods, such as the homogenization and SIMP approaches, or geometry-based methods, such as the LS method.

Additionally, there are heuristic methods, including evolutionary structural optimization (ESO) and bi-directional evolutionary structural optimization (BESO), though these will not be addressed in this article [7].

As a key focus, the present study demonstrates the implementation of computational algorithms for topology optimization, drawing on foundational works such as [1, 8].

The proposed implementations were indeed developed starting from the educational codes based on the SIMP model [8] and the Level Set approach [1]. However, these original codes were modified to accommodate the specific structural configuration investigated in this work, namely a bridge-domain scenario. In particular, the boundary conditions and loading configuration were reformulated to represent two essential features of the bridge problem: lower fixed support points and an upper distributed load. These conditions are not part of the original educational implementations and required corresponding modifications to the finite-element model, the definition of the boundary conditions, and the application of the external loading.

Therefore, the contribution of the present work is not the development of a new SIMP or Level Set optimization algorithm, but rather the adaptation and extension of the established implementations to a bridge-specific structural domain and its associated support and loading conditions. This adaptation enables the two topology-optimization approaches to be applied consistently to the structural scenario considered in this study and allows their resulting topologies to be investigated under the same bridge-specific conditions.

The findings confirm that topology optimization is a highly promising method for structural synthesis, offering broad applicability across multiple engineering domains, including the aerospace, automotive, and chemical industries.

2. Theoretical reference

2.1. Structural optimization theory

Topology optimization is one of several techniques within the broader field of structural optimization. A comprehensive overview of structural optimization is provided in [9], which traces the evolution of the field from classical analytical formulations—such as Michel's foundational work in 1904—to the dominance of

computational models that emerged with the advent of digital computing in the 1960s and 1970s.
Structural optimization can be classified into at least five major categories [10]:

- Parametric optimization,
- Shape optimization,
- Topology optimization,
- Multi-objective optimization, and
- Robust or multidisciplinary optimization,

as illustrated in Fig. 1.

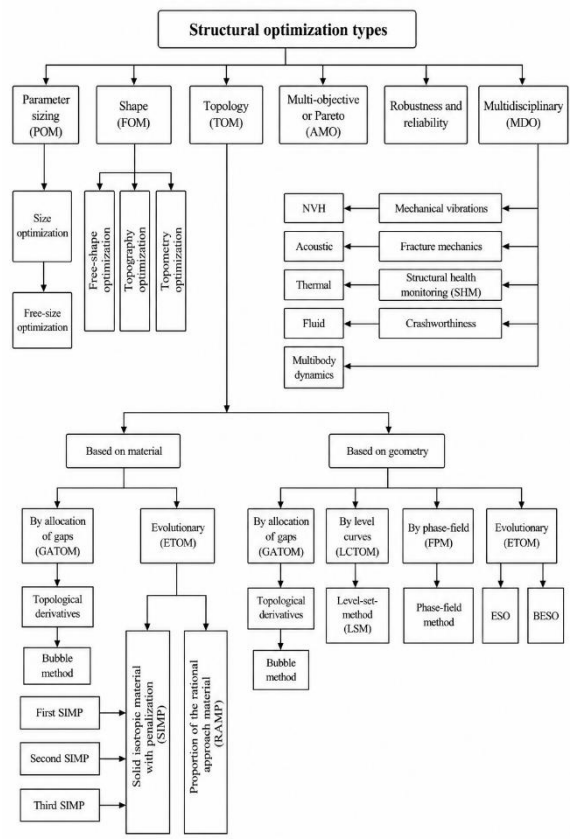


Fig. 1. Summary of current types of structural optimization classes.

Among these, the three classical approaches that have historically defined the field are:

i. Parametric optimization (also referred to as dimensional optimization),
ii. Shape optimization (geometric refinement), and
iii. Topology optimization (material layout within a design domain),

as depicted in Fig. 2.
In parametric optimization, the structural design is refined by adjusting geometric parameters such as beam length, width, and thickness. This method typically focuses on reducing the number of structural elements (e.g., trusses) while maintaining performance criteria.

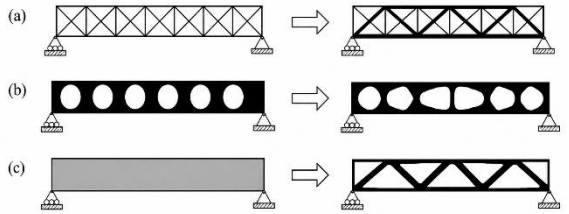


Fig. 2. Structural optimization classes; (a) parametric optimization, (b) shape optimization, and (c) topology optimization [10].

In shape optimization, modifications are made directly to the geometry by altering the positions of nodes or elements. This approach is often used to refine existing voids or contours in the structure, resulting in more efficient and aerodynamically or mechanically favorable forms.
Topology optimization, sometimes referred to as free-form structural design, differs fundamentally from the previous methods. It does not require predefined shapes or dimensions.
Instead, it determines the optimal material distribution within a defined design domain by systematically removing material from regions experiencing low energy or low mechanical performance.
The result is an efficient structural layout that may include novel voids or load paths not anticipated in traditional design approaches.
TO is a computational technique that enables the synthesis of innovative structural topologies. Unlike parametric optimization, TO does not rely on fixed dimensions but instead evolves the material layout to achieve optimal structural performance under specified constraints and loading conditions.

2.2. Theory of topology optimization

The concept of TO was pioneered by Bendsøe and Kikuchi in 1988 [11], building upon earlier research in structural optimization and the analytical study of material minimization problems with volume constraints initiated during the 1980s [12].

Initially, TO was formulated based on the homogenization method, and its early applications were predominantly focused on the aerospace sector, where minimizing material weight was critical to reducing manufacturing costs.

The motivation to develop topology optimization arose partly due to the limitations of shape optimization (SO), which was constrained in its ability to introduce and control complex voids ("holes") within structures without compromising their physical integrity. TO algorithms have since evolved to address these challenges by enabling systematic material distribution within a design domain.

According to Sigmund [5], topology optimization fundamentally answers a key engineering question: How should material be optimally placed or removed within a given design space to achieve the best possible structural performance? This capability has established TO as a vital tool in modern structural design and engineering.

2.3. SIMP and LS approach

The SIMP model was developed as an evolution of the homogenization method [13, 14], with significant contributions by [15] that formulated the approach based on material density variables. The objective of the SIMP method is to determine the optimal material distribution within a fixed design domain to minimize an objective function—often structural compliance—subject to constraints such as volume or stress limits. Material distribution is represented through a pseudo-density variable that ranges continuously between 0 (void) and 1 (solid) at any point in the domain.

This method discretizes the extended design domain into finite elements, assigning a density variable to each element. The original SIMP formulation was inspired by porous material models, wherein microstructures with voids exhibit varying densities.

It serves as a computationally efficient approximation that simplifies the complex homogenization process by penalizing intermediate densities to promote a near 0-1 (void-solid) solution.

The SIMP approach enables the identification of clear solid regions (density aprox. 1), void regions (density aprox. 0), and intermediate "grayscale" zones, which represent transitional material states. According to [16], iterative updates in topology optimization progressively refine this material distribution to minimize compliance and maximize stiffness, often exploiting domain symmetries to simplify computations.

The LS method represents an alternative topology optimization framework based on geometric boundary evolution rather than material density interpolation [17]. Instead of local density variables, the domain boundary is implicitly represented by a level set function, which defines material interfaces via contours.

This approach provides a clear, explicit definition of structural boundaries throughout the optimization process, as highlighted by [4].

Pioneering works on LS methods include [18-21]. Although the LS method offers advantages in boundary representation, its practical use has often been combined with other strategies such as SIMP and BESO to enhance convergence and robustness.

One notable variant is the Pareto LS method, discussed by [17], which incorporates topology sensitivity—a measure quantifying the effect of introducing infinitesimal holes on the structural response. Regions exhibiting high topology sensitivity indicate areas where material removal significantly alters performance, guiding the optimization toward effective material redistribution.

When compliance is selected as the structural response metric, analytical expressions for topology sensitivity enable precise boundary updates. The LS method uses level set values to track evolving boundaries and topologies throughout optimization.

According to [22], the level set approach is inherently non-gradient-based and constrains the density variables to binary states—1 for solid and 0 for void—enabling sharp and well-defined material boundaries. This discrete representation facilitates the generation of clear structural contours that minimize compliance while satisfying volume constraints.

2.4. Optimization algorithms in MATLAB®

To effectively translate the topology optimization process into a computational framework, the MATLAB® software environment was employed. MATLAB® offers an extensive suite of functions and subroutines designed to solve both linear and nonlinear optimization problems using explicit approximation methods.

As described by [23], numerical optimization algorithms interactively seek the optimal material distribution within the design domain.

The literature encompasses a wide range of optimization algorithms, spanning both method-specific and generic approaches. According to [24], MATLAB® serves as a powerful computational tool capable of addressing constrained and unconstrained, linear and nonlinear optimization problems through its comprehensive function libraries.

For instance, the built-in function linprog utilizes the Simplex method to solve small-scale linear programming problems; subsequent enhancements have introduced solvers optimized for larger problems, significantly reducing the computational time required for topology optimization synthesis. This functionality has been extensively adopted to assist programmers in solving engineering optimization problems due to its robustness and efficiency [25].

As noted by [26], the rise of topology optimization has prompted the development of numerous MATLAB® codes tailored to different problem dimensions—one-dimensional (1D), two-dimensional (2D), and three-dimensional (3D).

Sigmund's seminal 2D topology optimization code [8] consists of 99 lines and employs stiffness matrix assembly and filtering via nested loops. While this structure enhances code readability and organization, it limits scalability due to slower execution times on larger problems.

Despite this, the code is widely regarded as a foundational and accessible tool for academic purposes, fostering numerous subsequent developments and applications.

Building upon this, [20] introduced the TOPLSM program, a 199-line MATLAB® implementation for 2D compliance optimization utilizing the LS method. This program incorporates the Upwind scheme to solve the Hamilton-Jacobi partial differential equation (PDE), PDE-based resetting, geometric quantity computations, and finite element analysis within a 2D framework.

Similarly, [1] developed a 129-line LS-based algorithm that employs discrete variables to address single-objective optimization problems, marking a significant contribution to LS method applications.

In multi-objective optimization, [27] utilized the ParetoOptimalTracing program, which traces Pareto fronts across varying volume fractions by leveraging topological sensitivity within the LS framework. This approach facilitates simultaneous optimization of objectives such as compliance minimization and displacement maximization.

Further code refinements include the MATLAB® implementation by [28], which condensed Sigmund's original 99-line code to 88 lines. This updated version replaces the sensitivity filter with a density filter, enhancing the smoothness of pseudo-density variable transitions across the design domain.

Comparative studies by [26] on a 7500-element structure demonstrated that the 88-line implementation achieves approximately twice the computational speed of the original 99-line version, highlighting the efficiency gains from these algorithmic improvements.

3. Methodology

3.1. Materials

The two-dimensional topology optimization algorithms were adapted and implemented based on the original codes developed by [8] and [1]. All computations were performed using MATLAB® student version R2017a (64-bit).

The SIMP-based algorithm employs gradient-based optimization coupled with the optimality criteria method for problem formulation, whereas the Level Set approach utilizes iterative updates of the level set function to evolve the topology.

Finite element analysis and topology optimization simulations were conducted on a Dell Inspiron notebook equipped with an Intel®

Core™ i7-3630QM processor running at 2.4 GHz, supported by 12 GB of RAM, and operating on Windows 10 (64-bit). This computational setup provided sufficient resources for efficient execution of the implemented algorithms.

3.2. Methods

3.2.1. Domain boundary condition

The design domain considered in this study corresponds to a simplified structural model, representative of a small bridge-like structure. The model features fixed support points located at the ends, designated as nsup1, nsup2, nsup3, and nsup4, with a uniformly distributed load applied along the upper surface, as illustrated in Fig. 3. This setup was chosen to investigate how the boundary conditions—specifically the fixed supports—and the uniformly distributed load influence the development of an optimized free-form structural profile through topology optimization.
The goal was to assess how these constraints and loading conditions drive the material distribution and structural configuration within the design domain.

3.2.2. Topology optimization algorithm based on SIMP - 2d

The proposed code development was based on the algorithm presented in [8], a seminal 99-line MATLAB® implementation widely recognized as one of the earliest topology optimization methods for synthesizing two-dimensional (2D) design domains. The compliance is known and can be seen in any development, as in [8].
Formulation Eq. (1):

Minimize

$$min_x: c(x) = U^T KU = \sum_{e=1}^{N} (x_e)^p \, u_e^T k_0 u_e$$

where K is the overall stiffness matrix of the structure

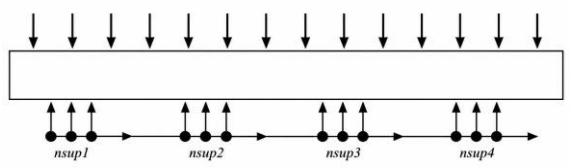


Fig. 3. Domain geometry.

Subject to $\frac{V(x)}{V_0} = f$

$$KU = F$$

$$0 \le x_{min} \le x \le 1 \qquad (1)$$

where $x_i (i=1 \ldots, M)$, which represents the density at each point in the discretized design domain, are the variables of the problem; M is the total number of variables, and V* is the volume of material desired at the end of the optimization.
Table 1 summarizes the input parameters as defined in [8], followed by a detailed explanation of the code structure, organized by individual lines or functional groups.
Although this algorithm does not guarantee rapid convergence and is not computationally efficient for large-scale problems, its straightforward implementation and extensive academic adoption make it an appropriate choice for this study.

3.2.2. Topology optimization algorithm based on SIMP - 2d

The proposed code development was based on the algorithm presented in [8], a seminal 99-line MATLAB® implementation widely recognized as one of the earliest topology optimization methods for synthesizing two-dimensional (2D) design domains. The compliance is known and can be seen in any development, as in [8].
Formulation Eq. (1):

Minimize

$$min_x: c(x) = U^T KU = \sum_{e=1}^{N} (x_e)^p \, u_e^T k_0 u_e$$

where K is the overall stiffness matrix of the structure.

Table 1. Description of input parameters of SIMP.

Variables	Description	Input
Nelx	No. elements in the horizontal direction	60
Nely	No. elements in a vertical direction	30
Volfrac	Is the solid fraction of volume	0.5
Penal	Is the penalty p, varies from 1 to 3	3
Rmin	It is the minimum radius of the filter	1.5

3.2.2. Topology optimization algorithm based on SIMP - 2d

The proposed code development was based on the algorithm presented in [8], a seminal 99-line MATLAB® implementation widely recognized as one of the earliest topology optimization methods for synthesizing two-dimensional (2D) design domains. The compliance is known and can be seen in any development, as in [8].
Formulation Eq. (1):

Minimize

$$min_x: c(x) = U^T KU = \sum_{e=1}^{N} (x_e)^p \, u_e^T k_0 u_e$$

where K is the overall stiffness matrix of the structure The current implementation employs a finite element mesh composed of simplified square elements; however, alternative element shapes and mesh configurations could be explored by modifying the element connectivity matrix.

3.2.3. Topology optimization algorithm based on LS – 2d

The code developed by [1], in contrast to the SIMP-based approach of [8], utilizes the LS method with discrete variables. This implementation integrates topological sensitivity analysis with the iterative update of the LS function to drive the optimization process. The solution of compliance is known and can be seen in any developments, as in [1].
Formulation Eq. (2):

$$min_x: c(x) = U^T KU = \sum_{e=1}^{N} u_e^T \, k_e u_e = \sum_{e=1}^{N} x_e \, u_e^T k_1 u_e \tag{2}$$

subject to:

$$V(x) = V_{req}$$

$$KU = F$$

$$x_e = 0 \text{ or } x_e = 1 \forall e = 1 \cdots N$$

The [1] code is modularized into several sections, encompassing a total of 199 lines. Table 2 summarizes the input parameters as defined in [1], followed by a detailed explanation of the code structure, organized by individual lines or functional groups.
For all numerical simulations presented in this section, the prescribed volume fraction was set to f=0.5 (50%). This value was consistently adopted in both the SIMP and Level Set implementations.

3.2.4. Proposed changes to the code

Modifications were introduced to the original educational codes based on the SIMP model and the Level Set approach to accommodate new boundary conditions reflecting the bridge domain scenario, specifically incorporating lower fixed support points and an upper distributed load. The fundamental mathematical formulation of the methods was retained; what was specifically adapted was the structural problem, particularly the loading and boundary conditions. The adapted mathematical formulation can then be explicitly presented.
The foundational algorithms are detailed in [8] and [1]. The revised SIMP algorithm, developed without the density filter and derived solely from Sigmund's original 99-line MATLAB® code (2001), can be executed using the command defined in Eq. (3). Similarly, as described in [1], Eq. (4) defines the input function corresponding to the Level Set algorithm.

topopt(nelx, nely, volfrac, penal, rmin, nsup1, nsup2, nsup3, nsup4) (3)

top_levelset(nelx, nely, volreq, steplength, NumReinit, topWeight, nsup1, nsup2, nsup3, nsup4) (4)

Table 2. Description of input parameters for LS.

Variables	Description	Input
Nelx	No. elements in the horizontal direction	60
Nely	No. elements in a vertical direction	30
Volreq	Is the solid fraction of volume	0.5
Steplength	Spec. the time interval of LS function	3
Numreinit	Freq. at which LS function is reset	2
TopWeight	The weight parameter	2

To adapt the original algorithms of [8 and 1] from the MBB beam setup to a full bridge-type beam configuration, initial modifications to the boundary conditions were necessary.
These adjustments enabled the transformation of Sigmund's 99-line code and Challis' 129-line code to accommodate the bridge structural design.
The original MATLAB® codes were modified only in the portions associated with the design domain, support conditions, and external loading. The principal optimization procedures, including the finite element analysis and the SIMP or Level Set updating strategy, were retained from the original implementations. The support variables `nsup1`, `nsup2`, `nsup3`, and `nsup4` identify the nodal ranges used to impose the two support regions in the bridge-like model. Their values are determined according to the adopted mesh and the desired locations of the supports.
The distributed load, originally specified in [N/m], needed to be converted into nodal forces [N], as the load is applied at the degrees of freedom (DOFs) associated with each element node.
To accurately distribute the total applied load along the beam in the x-direction—aligned with the load application—the total force, Ftotal, was divided evenly across the nodes by calculating –Ftotal/ nelx+1.
This approach ensures uniform load distribution across all nodes. For example, as illustrated in Fig. 4, a beam consisting of 10 elements arranged linearly (i.e., nelx = 10 and nely = 1) was used to define the nodal loads along the upper edge of the structure. Also, as illustrated in Fig. 4, each node possesses two degrees of freedom (DOFs): one corresponding to horizontal displacement and the other to vertical displacement. These DOFs are systematically numbered from top to bottom and left to right.

Furthermore, the positions of the 10 elements are defined accordingly. Based on this configuration, the example parameters are as follows:

elx: Position of the elements, which can be between 1 and 10.
2, 6, 10, 14, 18, 22, 26, 30, 34, 38, 42: DOFs in the top row in the vertical position.

Consequently, the load condition that fulfills these requirements is defined by Eq. (5).

$$F(2*(nely+1)*(elx-1)+2,1) = -Ftotal/(nelx+1); \quad (5)$$

$$F(2 * (nely + 1) * (elx -1)+2,1)= -1 \quad (6)$$

Since a unit load was adopted for the example, the resulting load magnitude is defined by Eq. (6).
For the fastening elements depicted in Fig. 3, modifications were required in the algorithm's loop structure. Specifically, as illustrated in Figs. 5 and 6, the fixation commands (fixeddofs) for both the SIMP model and the LS algorithm were updated to accommodate the revised boundary conditions.

4. Results and discussion

The parameters defined in Eqs. 3 and 4 were initially presented but required adjustment according to Eqs. 5 and 6, as illustrated in [1] and [8]. The key parameters—Ftotal, nsup1, nsup2, nsup3, and nsup4—represent physical boundary conditions within the project domain and must be modified in the code for each specific analysis to reflect changes in support locations or load magnitudes.

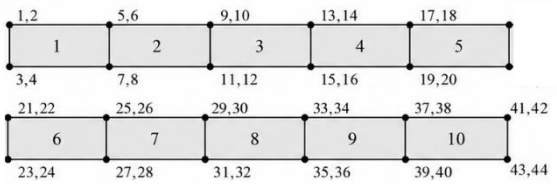


Fig. 4. Example to define distributed loads on domain nodes.

```
% DEFINE LOADS AND SUPPORTS (HALF MBB-BEAM)
Ftotal = 1;
for elx = 1:nelx+1
    F(2*(nely+1)*(elx-1)+2,1) = -Ftotal/(nelx+1);
end
fixeddofs = union(2*(nely+1)*(nsup1:nsup2),2*(nely+1)*(nsup3:nsup4)
alldofs = 1:2*(nely+1)*(nelx+1);
freedofs = setdiff(alldofs,fixeddofs);
```

Fig. 5. Changes in the SIMP algorithm**.**

```
% DEFINE LOADS AND SUPPORTS
F = sparse(2*(nely+1)*(nelx+1),1);
Ftotal = 1;
for elx = 1:nelx+1
    F(2*(nely+1)*(elx-1)+2,1) = -Ftotal/(nelx+1);
end
fixeddofs = union(2*(nely+1)*(nsup1:nsup2),2*(nely+1)*(nsup3:nsup4)
alldofs = [1:2*(nely+1)*(nelx+1)];
freedofs = setdiff(alldofs,fixeddofs);
U(freedofs,:) = K(freedofs,freedofs) \ F(freedofs,:);
```

Fig. 6. Changes in the LS algorithm.

For demonstration, the input functions for the implemented SIMP and LS algorithms were specified as topopt (60, 30, 0.5, 3.0, 1.5) and toplevelset (60, 30, 0.5, 3, 2, 2), respectively.
Here, nelx = 60 denotes 60 finite elements along the x-direction, with supports positioned between nodes 10 to 15 (nsup1 to nsup2) and nodes 45 to 50 (nsup3 to nsup4). The total applied load was set as Ftotal = -1, representing a unit load.
Fig. 7 illustrates the comparison between the two algorithms: Fig. 7(a) displays results from the SIMP implementation, while Fig. 7(b) shows outputs from the LS implementation.
The differences between the SIMP and Level Set results are associated with the distinct design representations and update mechanisms employed by the two methods. In the SIMP formulation, the material distribution is represented by element density variables, and the optimization process progressively modifies these densities according to the sensitivity information and the optimality-criteria update scheme. In contrast, the Level Set Method describes the structural boundary implicitly and updates its position through a boundary-evolution equation.
Therefore, the two approaches may produce different material layouts, boundary configurations, and compliance values, even when the same structural domain and volume constraint are used. The comparison should consequently be interpreted not only in terms of the final topology but also in terms of numerical convergence, material usage, and computational cost. Additionally, the compliance outcomes from the two algorithms differ in both initial and final values, which is expected given their distinct optimization strategies. Consequently, the compliance minimization results vary between the two synthesized designs.
Table 3 summarizes these quantities for the SIMP and Level Set implementations under the same bridge-domain configuration, loading conditions, boundary conditions, and prescribed volume fraction. This provides a more objective assessment of the similarities and differences between the two approaches.
The actual volume fraction is reported based on the final optimized design, rather than only the prescribed target value, allowing the effectiveness of the volume constraint to be assessed. The computational time and number of iterations are also reported to provide information regarding the computational cost and convergence behavior of each method.
Fig. 8 compares the final compliance and volume fraction obtained using the SIMP and Level Set methods. The results indicate the structural performance and material distribution achieved by each approach under the same optimization conditions. The final volume fractions are close to the prescribed target, while the differences in compliance reflect the distinct numerical formulations and boundary-evolution strategies employed by the two methods.

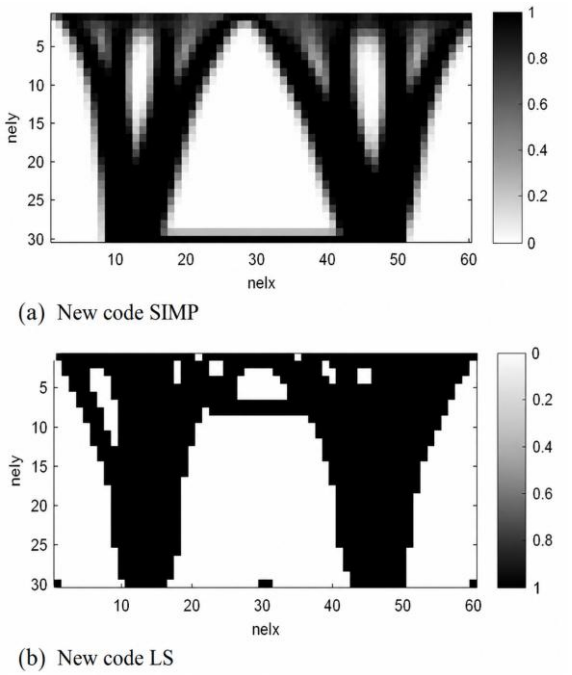


Fig. 7. Implementation of the modified algorithm with the SIMP model [8] and LS approach [1] for a 60 × 30 mesh, volume fraction of 50 %, p=3, rmin=1m5, steplength=3, numreinit=2, and topweight=2.

Table 3. Quantitative description of the results of SIMP and LS.

Parameter	SIMP	Level Set
Final compliance	1.685	2.037
Number of iterations	425	36
Computational time (s)	45.183	13.262
Final volume fraction	0.498	0.503

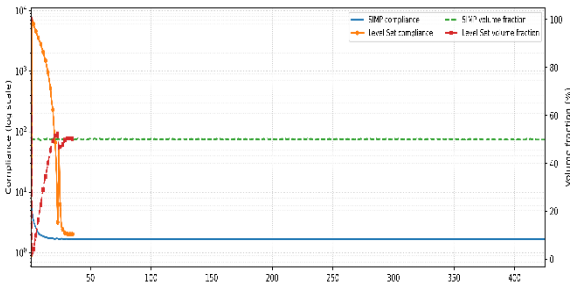


Fig. 8. Compliance convergence and final volume fraction of the SIMP model [8] and LS approach [1].

These results complement the visual comparison of the optimized topologies and provide a more objective basis for discussing the differences between the two methods.

5. Conclusions

The methods employed exhibit distinct characteristics; however, the boundary conditions yield comparable results. Both approaches apply forces along the upper surface, represented by the dark-shaded regions, and feature two fixed support points on the lower surface, also indicated by dark shading.

Notably, near the attachment points and adjacent to the upper edge, a structural pattern emerges resembling a chassis frame stringer. According to topology optimization principles, since the load is distributed on the upper face, material continuity should be maintained in this region, with minimal or no voids or discontinuities.

To address this, multiple volume fraction scenarios will be tested to reduce or eliminate such undesired features.

Overall, the topology optimization approach applied to the bridge-inspired design domain—characterized by distributed loads on the top surface and fixed supports underneath—is validated.

Nonetheless, it is recommended to develop a comprehensive library of two-dimensional topology optimization implementations accommodating a broader range of boundary conditions for enhanced versatility in future applications.

The present study has several limitations. The analyses are restricted to a two-dimensional structural domain and are based on linear elastic finite element modeling. The study does not consider stress constraints, buckling, fatigue, geometric nonlinearity, manufacturing restrictions, or three-dimensional effects. In addition, the results may be influenced by the adopted mesh resolution, optimization parameters, initialization, and convergence criteria. The comparison is therefore specific to the investigated bridge-like configuration and should not be generalized to all topology optimization problems without further investigation.

The present study should be interpreted as a computational implementation and comparative assessment rather than as an independent validation of the two optimization algorithms. Although the adopted formulations are based on established educational implementations, the modified bridge-like problem was not compared with an analytical solution, experimental data, or a commercial topology optimization package.

Consequently, the results demonstrate the behavior of the implemented algorithms under the defined numerical conditions but do not establish general validation for all structural configurations.

Future work should include benchmark reproduction, comparison with independent implementations, and verification using commercial or experimentally supported numerical models. Future work will include benchmark problems with reference solutions, mesh-convergence investigations, comparisons with established topology optimization software, and, where appropriate, experimental verification. Future studies should extend the comparison to three-dimensional structures, alternative support and loading configurations, and a broader range of volume fractions. Additional investigations should also consider mesh-sensitivity analyses, parameter studies, stress-constrained optimization, buckling constraints, manufacturing restrictions, and additive manufacturing requirements. The use of independent implementations and commercial software would further support the verification of the adapted algorithms.

Data availability statement

The MATLAB codes used in this study were adapted from previously published educational

topology optimization codes and are available from the corresponding author upon reasonable request.

Conflict of interest

The authors declare that they have no conflict of interest.

Acknowledgment

The authors also thank the Santo André Foundation (FSA) and the Federal Institute of Education, Science and Technology of São Paulo (IFSP) for their support and for the resources provided for this research.

References

[1] V. J. Challis "A discrete level-set topology optimization code written in MATLAB®". *Struct. Multid. Optim.*, Vol. 41, No. 3, pp. 453–464, (2010).

[2] R. Behrou, M. Lawry, and K. Maute "Level set topology optimization of structural problems with interface cohesion". *Int. J. Numer. Methods Eng.*, Vol. 112, No. 8, pp. 990–1016, (2017).

[3] M. Otomori, T. Yamada, K. Izui, and S. Nishiwaki "Level set-based topology optimisation of a compliant mechanism design using mathematical programming". *Mech. Sci.*, Vol. 2, No. 1, pp. 91–98, (2011).

[4] N. P. Van Dijk, K. Maute, M. Langelaar, and F.Van Keulen "Level-set methods for structural topology optimization: A review", *Struct. Multidiscip. Optim.*, Vol. 48, No. 3, pp. 437–472, (2013).

[5] O. Sigmund and K. Maute, "Topology optimization approaches: A comparative review", *Struct. Multid. Optim.*, Vol. 48, No. 6, pp. 1031–1055, (2013).

[6] A. A. Filho. Elementos Finitos: a base da tecnologia CAE. Editora Erica, São Paulo, Brazil, (2009).

[7] O. Querin, G. P. Steven, and Y. M. Xie "Evolutionary structural optimisation (ESO) using a bidirectional algorithm". *Eng. Comput.*, Vol. 15, No. 8, pp. 1031–1048, (1998).

[8] O. Sigmund "A 99 line topology optimization code written in MATLAB®". *Struct. Multid. Optim.*, Vol 21, No. 2, pp. 120–127, (2001).

[9] J. S. Arora. Optimization of structural and mechanical systems. Singapore: World Scientific Publishing, (2007).

[10] M. P. Bendsøe, O. Sigmund. Topology Optimization: Theory, Methods, and Applications 2nd ed. Berlin, Germany: Springer-Verlag, (2004).

[11] M. P. Bendsøe and N. Kikuchi "Generating optimal topologies in structural design using a homogenization method". *Comput. Methods Appl. Mech. Eng.*, Vol. 71, No. 2, pp. 197–224, (1988).

[12] G. I. N. Rozvany, N. Olhoff, K. T. Cheng, and J. E. Taylor "On the Solid Plate Paradox in Structural Optimization", *J. Struct. Mech.*, Vol. 10, No. 1, pp. 1–32 (1982).

[13] M. P. Bendsøe "Optimal shape design as a material distribution problem". *Struct. Optim.*, Vol. 1, No. 4, pp. 193–202, (1989).

[14] G. I. N. Rozvany and M. Zhou "The COC algorithm, part I: Cross-section optimization or sizing," Comput. *Methods Appl. Mech. Eng.*, Vol. 89, No. 1-3, pp. 281–308, (1991).

[15] H. P. Mlejnek "Some aspects of the genesis of structures", *Struct. Optim.*, Vol. 5, No. 1-2, pp. 64–69, (1992).

[16] E. C. Nelli Silva, Topology optmization applied to the design of linear piezoeletric motors, J. Intell. Mater. Syst. Struct., Vol. 14, No. 4–5, pp. 309–322, (2003).

[17] K. Suresh, M. Takalloozadeh "Stress-constrained topology optimization: A topological level-set approach", *Struct. Multid. Optim.*, Vol. 48, No. 2, pp. 295–309, (2013).

[18] G. Allaire, F. Jouve, A. M. Toader "Une méthode de lignes de niveaux pour l'optimisation de forme", *Comptes. Rendus. Math.*, Vol. 334, No. 12, pp. 1125–1130, (2002).

[19] G. Allaire, F. Jouve, and A. M. Toader "Structural optimization using sensitivity analysis and a level-set method", *J. Comput. Phys.*, Vol. 194, No. 1, pp. 363-393, (2004).

[20] M. Wang, X. Wang, and D. Guo "A level set method for structural topology optimization", *Comput. Methods Appl.*

Mech. Eng., Vol. 192, No. 1-2, pp. 227–246, (2003).
[21] S. J. Osher "Fronts Propagating with Curvature Dependent Speed", *Comput. Phys.*, Vol. 79, No. 1, pp. 12-49, (1988).
[22] J. M. J. Alphen Structural Optmization for 3D concrete printing, Ph.D. Thesis, Eindhoven University of technology, Eindhoven, The Netherlands, pp. 1-115, (2017).
[23] C. R. Lima, G. H. "Paulino, Auxetic structure design using compliant mechanisms: A topology optimization approach with polygonal finite elements". *Adv. in. Eng. Soft*. Vol. 129, pp. 69-80, (2019).
[24] A. Messac A Optimization in Practice with MATLAB®: For Engineering Students and Professionals, Cambrige, UK: Cambridge University Press, (2015).
[25] T. F. A. Martins Otimização de Estruturas de Treliças utilizando Técnicas de Programação Linear, M.Sc. Dissertation. Universidade Federal de Pernambuco (UFPE), Recife, Brazil, (2011).
[26] K. Liu and A. Tovar "A An efficient 3D topology optimization code written in MATLAB®". *Struct. Multid. Optim.*, Vol. 50, No. 6, pp. 1175–1196, (2014).
[27] K. Suresh "A 199-line MATLAB® code for Pareto-optimal tracing in topology optimization". *Struct. Multid. Opti.*, Vol. 42, No. 5, pp. 665–679, (2010).
[28] E. Andreassen, A. Clausen, M. Schevenels, B. S. Lazarov and O. Sigmund "Efficient topology optimization in MATLAB® using 88 lines of code". *Struct. Multid. Optim.*, Vol. 43, No 1, pp. 1–16, (2011).