Centro de Investigación en Matemáticas A.C.

CIMAT

Topology optimization algorithms for the solution of compliance and volume problems in 2D

Tesis

que para obtener el título de

Maestro en Ciencias Computacionales y Matemáticas Industriales

presenta:

Miguel Angel Ochoa Montes

Asesores:
Dr. Sergio Ivvan Valdez Peña
Dr. Salvador Botello Rionda

Guanajuato, México. 2016

Abstract

The thesis tackles the problem of structural design through topology optimization methods. The main contributions are the following. Firstly, a numerical benchmark is proposed considering the similarities of tests in the related literature, every test of the benchmark is well-established and used to compare the numerical and visual performance of the methods. Then, an analysis of the local optimization method Solid Isotropic Material with Penalization is carried out, in order to introduce two modifications: 1) to reduce the computational cost in problems of stiffness maximization, and 2) to minimize the volume of the structures. Finally, a novel technique to reduce the dimension of the problem through a proposal of a structure representation is introduced. This representation is well suited for mono and multi objective evolutionary algorithms in order to maximize stiffness and minimize volume. All the algorithms take advantages of parallel computing to distribute the computational cost implied in the evaluation of solutions.

To Andrea.
To my parents Dolores and Martin.
To my siblings Diana, Jessica and Daniel.
And to my nephew Daniel.

Thanks

To Andrea and my family for your unconditional love and support during this step of my live.

To my directors Ivvan Valdez and Salvador Botello for your guidance, help and support. To Jordi Pons-Prats for your help during my stay in Barcelona.

To the CIMAT institution and community, especially to my colleagues and friends.

To CONACYT for the economic support during my Master formation and my stay in Barcelona.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation and justification

Structural optimization is a process of mechanical design, its main objective is to find an optimal structure, capable of fulfilling service conditions given by 1) displacement conditions(regions where the structure is fixed) and 2) force conditions(regions where the structure is loaded). An example is shown in figure 4.1, where two displacements conditions are located at the bottom and the top on the left side respectively, and one force condition is located at the bottom on the right side. Structural optimization has represented advantages in fields as the mechanical and civil engineering, due to optimization is oriented to generate benefits in the cost, safety and functionality of the structures.

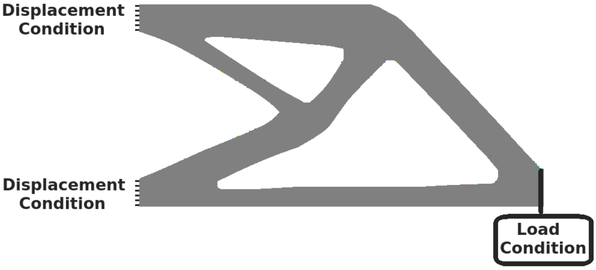


Figure 1.1: Example of structural optimization

One approach in structural optimization is *topology optimization*, which process consists in distributing material over a design domain, assigning the position and shape of the holes and structural elements that generate the optimal design, an example is shown in figure 1.2. The two most common manners to define the topology optimization problem are the compliance minimization (which is directly related to the stiffness maximization of the structure) and the volume minimization of the structure.

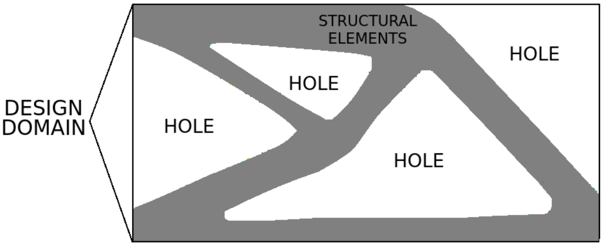


Figure 1.2: Topology optimization approach

Based on the above, the main objective of this work is to solve the problem of optimal design through the following two perspectives:

- Structural functionality: related to the design of structures with maximum rigidity.

- Structural cost: related to the design of structures with minimal volume, and capable of fulfilling the service conditions with the minimum material.

In order to solve these problems, it is necessary the use of optimization algorithms. There are many algorithm categories to solve a topology optimization problem, as *homogenization methods*[21], *level set methods*[9],*phase field methods*[24], *heuristics*[155] and *meta-heuristics*[16]. For this work, a homogenization method: Solid Isotropic Material with Penalization(SIMP), and meta-heuristics: Evolutionary Algorithms, are used. SIMP is a computationally efficient local method that generates excellent results in mechanical performance and, especially, in aesthetic. Evolutionary algorithms are global optimization methods, which have been successfully used to tackle a variety of optimal design problems, for this reason we consider that they have the potential for the solution of topology optimization problems.

In this work, we establish a set of contributions for the SIMP and evolutionary algorithms for topology optimization.

- SIMP: the first contribution is a convergence criterion that turns the method considerably more efficient, through the reduction of iterations, and generating structures with a similar performance than the original criteria. The second contribution is a proposal for the solution of both perspectives(functionality and cost).

- Evolutionary algorithms: the contribution is a proposal for the reduction of the problem dimension. A tractable dimension in the problem makes feasible the use of mono and multi-objective evolutionary algorithms for the solution of both perspectives(functionality and cost).

Another important contribution of this work is the establishment of a benchmark set of problems for topology optimization. In an extensive bibliographic review, we note that, despite authors use similar tests to demonstrate the performance of their algorithms, there is not a set of well-established problems. Tests differ, mainly, in domain sizes, position and magnitudes of conditioned regions, properties of the material, and meshes. For this reason, a set of 13 tests is established (according to the most frequently used in literature) in order to generate fair comparisons between algorithm implementations.

This document is structured as follows: chapter 1 introduces a motivation for this work, describing briefly the problem of topology optimization and the importance of the use of these techniques in engineering. Then, it briefly reviews the methods in the state of the art. Chapter 2 introduces the theoretical framework, for contextualizing this work. This chapter presents a review of some necessary concepts as general optimization, the finite element method, topology optimization and parallel computing. Chapter 3 contains the problem definition and the specific objectives. In this chapter, the problems to solve are described in detail: 1) the benchmark establishment and 2) the proposed solution for local and global topology optimization, using different optimization models and algorithms. Chapter 4 contains a detailed description of a well established benchmark, integrated by tests commonly used in specialized literature, with the purpose of applying each algorithm to these tests, getting a real fair comparison between them. Chapter 5 contains a description of the work related to local topology optimization. This chapter includes a detailed analysis of the SIMP method, used for the solution of compliance problems, and a variant of this method for the solution of volume problems. Chapter 6 contains a description of the work related to global topology optimization. This chapter includes a description of a proposed representation for structures, used to reduce the problem dimension, and making it suitable for the use of evolutionary algorithms. Also the implementation of mono and multi-objective evolutionary algorithms for the solution of compliance and volume problems. Chapter 7 contains the results from every algorithm implementation. Results include material distribution, stresses and displacements behavior; efficiency, objective functions, security factor and other interesting data. Chapter 8 contains observations and conclusions about the local and global optimization algorithms implemented, as well as observations and conclusions about the results obtained from each test. In addition, we propose directions for future work.

1.2 State of the art

The topology optimization problem has been solved through different methodologies, they include from methods with a priori knowledge in the elastic mechanic problem to meta-heuristics capable of solving optimization problems using only the objective function. The following is a categorization of methods based on the way of tackling the topology optimization problem:

- Homogenization methods [21],[166],[69],[63],[62],[42]: this kind of methods is based on the idea that the volume and density of an element can be varied by modifying the micro-structure of such element. A macro parameter x_e is assigned to each element, it can take values between a range $[x_{min}, x_{max}]$, which allows simulating the variation of the micro-structure as well as its changes in density(volume) and stiffness(Young modulus). The most popular homogenization method is the Solid Isotropic Material with Penalization: SIMP [20],[21], [144]. In this method, the variables x_e have a range with x_{min} around $1E-3$

and $x_{max} = 1$, they are used to scale Young modulus (denoted by E) in the way $E_e = E_0 x_e^p$ (where p is suggested as $p = 3$), reducing the stiffness on the element e. The updating equation for variables x is derived from a relaxed fixed point method applied to the derivative of the compliance equaled to zero. The derivative is calculated under the assumption that structural self-weight is not considered. It is a local minimization method. This method is analyzed in detail in chapter 5.

- Level set methods [9], [11], [7], [5]: a level set function $\gamma(x)$ is defined and known over the entire design domain. If $\gamma(x) < T$ the structure has material in the x location, otherwise there is no material in this position. An initial level set function is defined by the user and it evolves iteratively using a Hamilton-Jacobi equation along to the objective function gradient respect to the normal direction of the boundary. In other words, a level set function is used to modify boundaries depending on the objective function derivative in a direction normal to the boundaries. Level set methods have advantages related to computational efficiency. As a disadvantage, they depend strongly on the initial level set function, so the optimal result is dependent on expertise of the user.

- Phase field methods [24]: also, a function describes the material existence. A phase field function represents the properties of the material in every zone over all the domain. This phase field function describes a smooth change between material and void phases. Numerical problems are reduced, in contrast with level set methods.

- Heuristics : in general, heuristic methods are based on the use of implicit knowledge to get a solution, not necessarily on mathematical principles. Possibly the most popular heuristic method for topology optimization is the Evolutionary Structural Optimization(ESO)[155] and its variants [77]. The main idea of ESO is to calculate a FEM analysis of the elastic problem over the design domain, in order to calculate a criterion in every element, such as Von Mises stress or tensile stress. Then, the less stressed elements are removed according to a rejection criterion. The process is repeated until reaching a convergence condition related to volume or stress. There is a Bidirectional ESO which not only remove elements but add elements [112], [66]. The main advantage of heuristic methods is they are simple and computationally efficient, nevertheless they do not guarantee convergence to a minimum (even local) because they do not update variables according to the minimization of an objective function, but according to implicit knowledge.

- Meta-heuristics [16],[27],[29],[61]: meta-heuristics are stochastic optimization methods, capable of solving a variety of problems. For topology optimization, the most common meta-heuristics methods are based on populations and they are called Evolutionary Algorithms. The main idea of these methods is based on the nature evolution process of the species: a population of solutions is randomly generated, then, the population is evaluated, and a group of the best individuals is selected according to their fitness acquired in the evaluation. Operators of crossover and mutation are applied to the selected set in order to generate a children population which replaces the current population. A detailed description of these methods is given in chapter 2. The main advantage of population-based meta-heuristics is that they are capable avoid local optimal solutions and approximate to global solutions, and the expertise about the problem is not necessary, nevertheless, they are not capable of managing a large amount of variables and they require a high amount of computational resources.

Chapter 2

Theoretical framework

2.1 Optimization

Optimization is the process of finding the parameters x that minimize or maximize an objective function $f(x)$ subject to constraints.

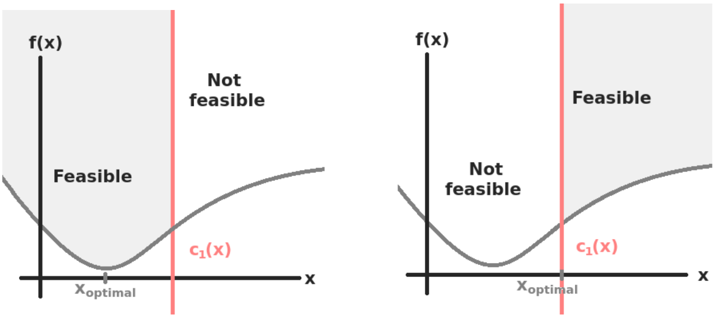
There are different methods to solve optimization problems, all of them require an *optimization model* that describes the problem. An optimization model is composed of the following [104]:

- Objective function $f(x)$: this is the function that describes the behavior of the phenomenon to optimize. This function extends for all a *region of feasible solutions*. The main objective is to minimize or maximize this function depending on the case.

- Design variables x: these are the variables whose objective function depends on. They represent the parameters to find, to get the optimal value in the objective function.

- Constraints $c_i(x)$: there are problems with or without constraints, frequently in real world problems one must consider constraints that delimit the *region of feasible solutions*.

Two optimization models are graphically shown in figure 2.1. Note that to the solution is different due the model constraints.



(a) Optimization model and solution example 1.

(b) Optimization model and solution example 2.

Figure 2.1: Examples of optimization models

In order to find the optimum, we require of an optimization algorithm, it usually consists of searching the values for design variables that minimize (without loss of generality) the objective function and those values do not violate the model constraints.

Optimization algorithms can be categorized as global and local. Also, they can be categorized in deterministic and stochastic.

2.1.1 Local and global optimization

Nocedal et al.(2000) define a global minimum as follows: “A point x^* is a global minimizer if $f(x^*) \leq f(x)$ for all x, where x ranges over all of $\mathbb{R}^n$(or at least over the domain of interest to the modeler)”, in other words, a global minimum is the lowest point in the feasible region of the objective function. On the other hand, Nocedal et al.(2000) defines a local minimum as follows: “A point x^* is a local minimizer if there is a neighbourhood $\mathbb{N}$ of x^* such that $f(x^*) \leq f(x)$ for all $x \in \mathbb{N}$, in other words, a local minimum is the lowest point in a delimited region(sub-region of the domain). This is shown in figure 2.2.

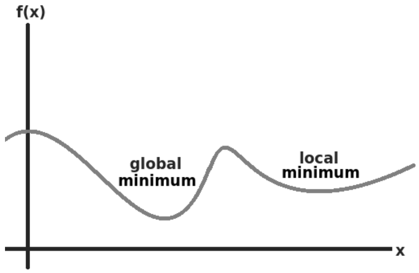


Figure 2.2: Examples of global and local optimization.

A method is considered global or local based on the capability to avoid local optimums reaching global solutions. Nevertheless, a local optimization method is capable to reach global optimums if it starts over a promissory solution, and a global optimization method can be stuck on local minimums.

2.1.2 Deterministic and stochastic optimization

Deterministic optimization

Deterministic optimization methods have a set of well-established steps based on deterministic computations. For a specific entry data, there is always the same result. Ming-Hua Lin et al.(2012) mentions about it: "Deterministic optimization take advantage of the analytical properties of the problem to generate a sequence of points that converge to an optimal solution" [83].

Stochastic optimization

There are many real-world problems where deterministic optimization is not effective, for example, the solution of combinatorial problems, discontinuous or non-derivable optimization models, and problems with multiple maximums or minimums on the objective functions.

In contrast, stochastic optimization methods use a random factor in the updating process which could favor scaping from local minimums. They generate random updates expecting to improve the current solution and expecting to lead the search to regions with better solutions. A kind of stochastic optimization methods are the evolutionary algorithms [119], which are inspired by the theory of evolution. In agreement with algorithm 1, evolutionary algorithms generate a set of random solutions called *individuals* (2. Generate population). Every individual is evaluated getting a *fitness*(4. Evaluate population). A *selection* process is applied to population: individuals with the best fitness are selected as *parents*(5. Selection). Then, an operator of *crossover*(6. Crossover) and *mutation*(7. Mutation) are applied to parents, in order to generate a (*children*) population, then, applying a survival operator, it is decided which individuals from original population and children form the next population(8. Replacement). The process is repeated until a stopping criterion is reached.

Algorithm 1 Evolutionary Algorithm Structure

```
 1: $t = 1$
 2: $X^t$ = Generate population()
 3: while Stopping criterion is not reached do
 4:     $F^t$=Evaluate population ($X^t$)
 5:     $X^t_{SEL}$=Selection ($X^t, F^t$)
 6:     $X^t_{child}$=Crossover ($X^t_{SEL}$)
 7:     $X^t_{\hat{child}}$=Mutation ($X^t_{child}$)
 8:     $X^{t+1}$=Replacement ($X^t, X^t_{\hat{child}}$)
 9:     $t = t + 1$
10: end while
11: Solution = $X^t_{best}$
```

Figure 2.3 graphically shows each step in algorithm 1. Each individual in the current population is colored differently, the best solutions are selected, genetic information of the first parent is shown in purple and the second in red. Then, this information is recombined to generate children via a crossover operator, notice that children preserve information from parents. Then, small changes non-dependent on the parents are inserted via a mutation operator. Finally, children replace old individuals, and the process is repeated.

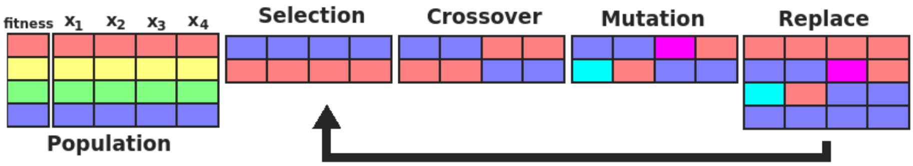


Figure 2.3: Graphical example of steps on evolutionary algorithm

Figure 2.4 shows the performance of an evolutionary algorithm through iterations. Red points are the current population and blue points are the selected set. Notice that the population is grouped into regions with minimal objective values, and finalizes on the global minimum.

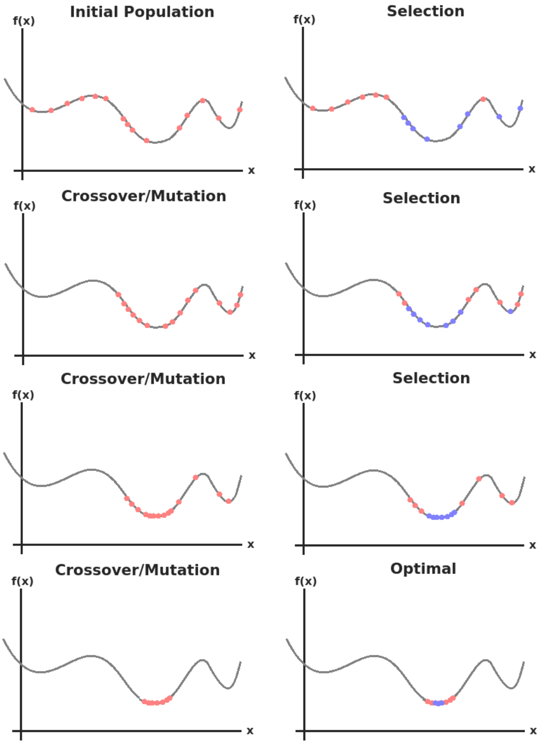


Figure 2.4: Graphical example of the evolution of solutions

Selection, crossover, mutation and replacement procedures are called operators. Each operator is used for different purposes:

- Selection: its objective is to select a set of the best n_{selec} individuals based on their fitness. n_{selec} can be given as a parameter or managed during execution. Common selection methods are [129]:
  - Roulette wheel: the probability of choosing an individual is proportional to its fitness. All fitnesses are accumulated and a uniform random value is taken between the accumulated limits. This way, individuals with the best fitness have more probability to be selected. This is repeated n_{selec} times until filling the *selected set.*
  - Binary tournament: two individuals are uniformly random chosen and they compete to be selected. The competition consists in comparing their fitness, the fittest is selected. This is repeated n_{selec} times until filling the *selected set.*
  - Truncation : the population is sorted according to the individual fitnesses, and the first n_{selec} individuals are selected.

- Crossover: its objective is to generate the children from the selected set. This step uses a parameter that determines the probability of crossing two selected individuals, otherwise, children are a replica of their parents. This step depends on individual representation: binary strings[146], combinations [113],[101] or real[48],[93].

- Mutation: this process is applied to the children. A probability is used to determine whether a gen (variables that form the solution) of the child is mutated. This step depends on the individual representation [146],[39],[47]

- Replacement: its objective is to generate a new population, considering individuals from the original population and the children. A process called *elitism* can be used in the replacement step, which consists in preserving the best $n_{elitism}$ individuals from the original population for the next iteration[59].

There are different many proposals on evolutionary algorithms, but the idea is similar: the most suited individuals survive and transmit their information to the next generations. The main difference among them are the operators: *selection*, *crossover*, *mutation*, and *replacement*.

Evolutionary algorithms used in this work are:

- Genetic Algorithms(GA): developed by John H. Holland in 1970 decade [64]. The formative algorithm only solved binary problems, nevertheless, now it is possible to solve real-code variable problems [44],[146],[48],[47]. The genetic algorithm implemented in here uses all the operators in algorithm 1 and works with real variables. Operators are used as follows:
    - Stopping criterion is established by a maximal iterations number.
    - Selection is done by the truncation method.
    - Crossover is done by a real variable method called *simulated binary crossover*(SBX), proposed by Deb and Agrawal in 1994. This method simulates a binary representation. SBX controls the children-parent similarity to preserve the population diversity. For a detailed description, consult [44] and [48].
    - Mutation is done by a real variable method known as the *polynomial mutation*, proposed by Deb and Agrawal in 1999. This method generates a controlled mutation on the neighborhood of the parents. For a detailed description consult, [45] and [47].
    - Replacement is done by preserving with the selected set, and filling the rest of population with the children.
- Estimation Distribution Algorithms(EDA). Developed by Mühlenbein at 1996 [99], these algorithms do not employ *crossover*, neither *mutation*, instead, they try to infer the underlying probabilistic distribution of the selected set, then, using that distribution, new individuals are generated. Thus, *crossover* and *mutation* steps are replaced by a *Calculate distribution* step. There are EDAs for solving binary, combinatorial [40] [38] and real variable problems, [78], [129],[130],[94]. Algorithm 2 shows the general structure of an EDA.

Algorithm 2 EDA Structure

```
$t = 1$
$X^t$=Generate population()
while Stopping criterion is not reached do
    $F^t$=Evaluate population ($X^t$)
    $X^t_{SEL}$=Selection ($X^t, F^t$)
    $p(x)^t$=Calculate distribution ($X^t_{SEL}$)
    $X^{t+1}$=Replacement ($X^t, p(x)^t$)
    $t = t + 1$
end while
Solution = $X_{best}$
```

The EDA used in this work is the *continuous Univariate Marginal Distribution Algorithm* (UMDA_c), introduced by Larrañaga and Lozano in 2002 [78]. For the *Calculate distribution* step in algorithm 2, the UMDA_c gets a search distribution by calculating the parameters of a *normal univariate distribution* for each dimension. Algorithm 3 shows the computation of the search distribution parameters in UMDA.

Algorithm 3 Calculate of search distribution for UMDA

```
for $i = 1$ to $n$ do
    $\mu_i = \frac{1}{N^S} \sum_{j=1}^{N^S} x_{i,j}$
    $\sigma_i^2 = \frac{1}{N^S} \sum_{j=1}^{N^S} (x_{i,j} - \mu_i)^2$
end for
```

Where x is the selected set, n is the dimension of the problem, N^S is the number of individuals in the selected set. The rest of the operators in algorithm 2 are used as follows:

- Stopping criterion is established by a maximal iterations number.
- Selection is done by the truncation method.
- Replacement is done by preserving with the selected set, and filling the rest of the population with the children generated

with the calculated distribution.

A detailed description of the steps of genetic algorithm and UMDA is given in the *Global optimization* chapter.

Multi-objective evolutionary algorithms In many practical optimization cases, it is desirable to find the optimal solutions for more than one objective function. Figure 2.5 shows the objective values of a population for two objective functions **F1** and **F2**.

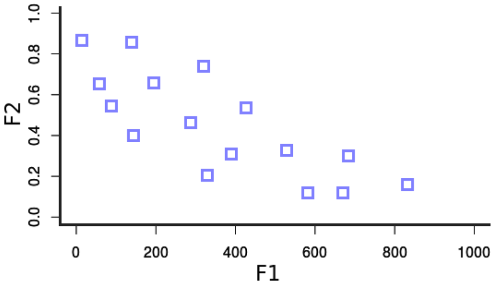


Figure 2.5: Example of population fitnesses on a multi-objective algorithm.

The approach of this kind of algorithms is practically the same than in mono-objective. They are based on the survival of the fittest individuals and the transfer of their information to next generations. The difference is the criterion to determine superiority among individuals. In mono-objective it depends uniquely on comparing the fitness, in multi-objective, identifying superiority is more complicated and depends on two or more measurements according to the method used. By instance, many of them use the Pareto criterion for determining whether an individual is better than another.

For this work, the Non-dominated Sorting Genetic Algorithm-II(NSGA-II) method[46] is implemented, which is a multi-objective version of the genetic algorithm, where the individual superiority is determined by the individual Pareto dominance and a crowding distance,[111],[97] .

According to the Pareto dominance, a solution p dominates another q (denoted by $p \preceq q$) in a set of objective functions f, if:

$$\forall i \in \{1, ..., k\} : f_i(p) \leq f_i(q) \quad \wedge \quad \exists i \in \{1, ...k\} : f_i(p) < f_i(q). \tag{2.1}$$

Which means that the solution p dominates q if p is better in at least one objective function and it is not worse in any other.

The main objective of the non-dominated sorting is to categorize the population individuals in fronts (or ranks): the first front is formed by the non-dominated individuals of the population, then they are removed and the second front is formed by the new non-dominated individuals, the process is repeated until every individual is ranked. Figure 2.6 shows an example.

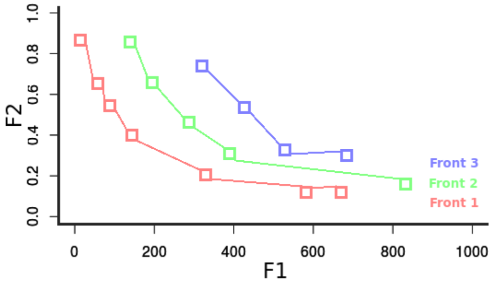


Figure 2.6: Multi Objective Fronts Example

Then, to determine superiority among individuals in the same front, the *crowding distance* of every individual is calculated. It is an average of the distances to the two closest neighbors in the same front. This value is higher for farthest individuals. A high crowding distance determines superiority. Figure 2.7 shows an example, individual **A** is clearly farthest from its neighbors than individual **B**, so, individual **A** has the highest crowding distance and is superior to **B**

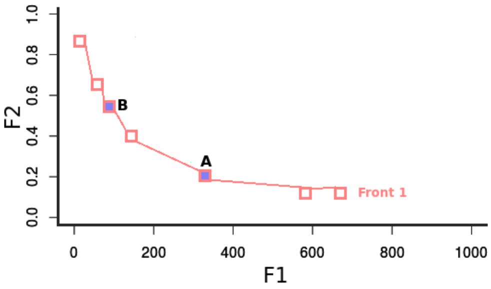


Figure 2.7: Multi Objective Crowding Distance

The NSGA-II uses these two criteria to get a selected set and for the replacement process, the rest of the process is the same than in mono-objective algorithms 1.

The main advantage of mono and multi-objective evolutionary algorithms is the flexibility they have for solving a variety of kinds of optimization problems, meanwhile, we have a function (or functions) to evaluate individuals. An evolutionary algorithm can be seen as a black-box method, where the explicit optimization model of the problem is unknown, and the only objective is to find an optimal solution. The main disadvantage is the high computational cost implied, nevertheless, population based methods are parallelizable in order to distribute the computational cost among different processing units.

2.2 Finite element method

The finite element method is used for the numerical solution of differential equations, they are solved in order to describe the behavior of physics phenomena, such as mechanical or thermal among others. Our interest in this method is the solution of the elastic problem, which consist in the calculation of the displacements generated by forces to an elastic body, which is fixed in determined regions. The role of the finite element method in structural optimization, is related to the optimization models, due to objective functions and constraints are calculated with the displacements and stresses. Note that every evaluation of a solution implies a finite element method analysis.

This work does not consider the finite element method implementation, thus the description in here is brief and the reader can consult to [169] and [23] for detailed sources. For this work, a C++ open source library is used for the solution of the elastic problem: FEMT [1].

2.2.1 Linear elastic problem

Displacements, strain and stress

Displacements field in planar stress and strain problems is defined as the displacements in x and y directions on each point belonging to the domain. The displacements vector is defined by:

$$\mathbf{u} = \mathbf{u(x,y)} \quad = \begin{bmatrix} u(x,y) \\ v(x,y) \end{bmatrix} = \begin{bmatrix} u \\ v \end{bmatrix} \tag{2.2}$$

Where $u(x,y)$ and $v(x,y)$ represent the displacements on x and y-axis respectively, at a specific point (x,y).

The strain field is calculated using the displacements in the xy plane, based on the general theory of elasticity. The strain vector is defined by:

$$\boldsymbol{\varepsilon} = \begin{bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{bmatrix} = \begin{bmatrix} \frac{\partial u}{\partial x} \\ \frac{\partial v}{\partial y} \\ \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \end{bmatrix} = \begin{bmatrix} \frac{\partial}{\partial x} & 0 \\ 0 & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \mathbf{Bu} \tag{2.3}$$

Where ε_x and ε_y are the normal strain on x and y-axis respectively and γ_{xy} is the shear strain in the xy plane. In planar strain problems, the strain in z-axis is considered as null($\varepsilon_z = 0$), as well as shear strain related to z-axis ($\gamma_{xz} = \gamma_{yz} = 0$)[23].

Stress field is defined by the vector:

$$\boldsymbol{\sigma} = \begin{bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{bmatrix} \tag{2.4}$$

Where σ_x and σ_y are the normal stresses in x and y-axis. τ_{xy} is the tangential stress in the xy plane. In planar stress problems, the stress in z-axis is considered as null($\sigma_z = 0$), as well as tangential stresses related to z-axis are null($\tau_{xz} = \tau_{yz} = 0$). The relation between stress and strain is the following:

$$\boldsymbol{\sigma} = \boldsymbol{D\varepsilon} \tag{2.5}$$

The general relation considering initial stress and strain:

$$\boldsymbol{\sigma} = \boldsymbol{D}(\boldsymbol{\varepsilon} - \boldsymbol{\varepsilon_0}) + \boldsymbol{\sigma_0} \tag{2.6}$$

This relation has a linear behavior until a limit known as elastic limit. This is shown in figure 2.8. The phase before elastic limit is known as *linear static*, where the behavior is proportional between stress and strain. Within this phase, the geometry has elastic properties in response to the forces that work on it, this means, when the forces are removed, the body returns to its original shape. If the forces generate stresses over a maximal theoretical permissible(or yield stress (YS)), the body passes from an elastic to a plastic behavior, where the body suffers permanent changes, even if the forces stop to work over it.

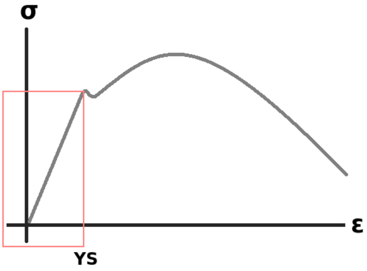


Figure 2.8: Stress and strain relation

D is the constitutive matrix that contains the information about the elastic properties of the material.

$$\boldsymbol{D} = \begin{bmatrix} d_{11} & d_{12} & 0 \\ d_{21} & d_{22} & 0 \\ 0 & 0 & d_{33} \end{bmatrix} \tag{2.7}$$

Based on the Maxwell-Betti theorem, $\boldsymbol{D}$ is symmetric [51]. Using isotropic materials and, for a planar stress problem, $\boldsymbol{D}$ is calculated as:

$$\begin{aligned} d_{11} = d_{22} &= \frac{E}{1-v^2} \\ d_{12} = d_{21} &= v d_{11} \\ d_{33} &= \frac{E}{2(1+v)} \end{aligned} \tag{2.8}$$

And for a planar strain problem, $\boldsymbol{D}$ is calculated as:

$$\begin{aligned} d_{11} = d_{22} &= \frac{E(1-v)}{(1+v)(1-2v)} \\ d_{12} = d_{21} &= d_{11}\frac{v}{1-v} \\ d_{33} &= \frac{E}{2(1+v)} \end{aligned} \tag{2.9}$$

Where E is the Young modulus and v is the Poisson radius of the material.

Discrete solution

The virtual work principle is an equation used in structural mechanics. It allows representing the equilibrium between the work generated by internal and external forces in a geometry. This equation is set using the relations between displacements, strain and stress as following:

$$\begin{gathered} \int\int_A (\delta\varepsilon_x\sigma_x + \delta\varepsilon_y\sigma_y + \delta\gamma_{xy}\tau_{xy})hdA = \\ \int\int_A (\delta u b_x + \delta v b_y)hdA + \\ \int_s (\delta u t_x + \delta v t_y +)hds + \\ \sum_i (\delta u_i U_i + \delta v_i V_i) \end{gathered} \tag{2.10}$$

The virtual work principle in a matrix form is:

$$\int\int_A \boldsymbol{\delta\varepsilon^T \sigma} hdA = \int\int_A \boldsymbol{\delta u^T b} hdA + \int_s \boldsymbol{\delta u^T t} hds + \sum_i \boldsymbol{\delta u_i^T q_i} \tag{2.11}$$

Where vectors $\boldsymbol{\delta u}$, $\boldsymbol{\delta\varepsilon}$ and $\boldsymbol{\sigma}$ represent the virtual displacements, virtual strain and stress fields as described before. Vectors **b**, **t** and **q** represent the forces that work over the geometry, specifically:

Forces over volume unities:

$$\mathbf{b} = \begin{bmatrix} b_x \\ b_y \end{bmatrix} \tag{2.12}$$

Forces over contour unities:

$$\mathbf{t} = \begin{bmatrix} t_x \\ t_y \end{bmatrix} \tag{2.13}$$

Forces over points:

$$\mathbf{q} = \begin{bmatrix} U_x \\ V_y \end{bmatrix} \tag{2.14}$$

These forces are given by the problem conditions. h is the thickness of the geometry, for planar strain problems $h = 1$ and for planar stress problems h is the real thickness.

Finite element method is capable of solving this equation in a discrete way, this means, computing the solution in some points(nodes) and interpolating the solution in intermediate points. Nodes and connections among them form elements, generating a complete mesh of the geometry. An example is shown in figure 2.9.

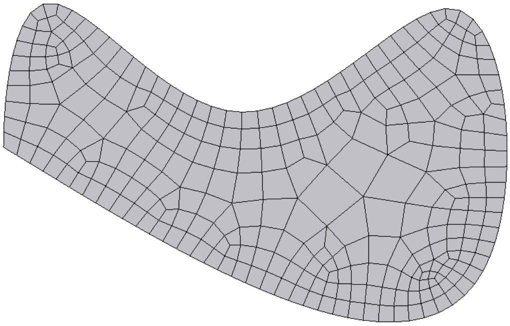


Figure 2.9: Example of geometry meshed.

The virtual work principle is used on every element. Based on equation 2.3, it is possible to transform the first term in the following:

$$\int\int_{A^e} \boldsymbol{\delta u^T B^T \sigma} h dA^e = \int\int_{A^e} \boldsymbol{\delta u^T b} h dA^e + \int_{s^e} \boldsymbol{\delta u^T t} h ds^e + \boldsymbol{\delta u^T q^e} \quad (2.15)$$

Since virtual displacements are arbitrary, they can be factorized and eliminated from the complete virtual work principle equation:

$$\int\int_{A^e} \boldsymbol{B^T \sigma} h dA^e = \int\int_{A^e} \boldsymbol{b} h dA^e + \int_{s^e} \boldsymbol{t} h ds^e + \boldsymbol{q^e} \quad (2.16)$$

Now, substituting the general relation in equation 2.6 in the first term:

$$\int\int_{A^e} \boldsymbol{B^T(D(\varepsilon - \varepsilon_0) + \sigma_0)} h dA^e = \int\int_{A^e} \boldsymbol{b} h dA^e + \int_{s^e} \boldsymbol{t} h ds^e + \boldsymbol{q^e} \quad (2.17)$$

The algebraic development is:

$$\begin{gathered} h\int\int_{A^e} \boldsymbol{B^T D \varepsilon} dA^e = \\ h\int\int_{A^e} \boldsymbol{B^T D \varepsilon_0} dA^e - h\int\int_{A^e} \boldsymbol{B^T \sigma_0} dA^e + \\ h\int\int_{A^e} \boldsymbol{b} dA^e + h\int_{s^e} \boldsymbol{t} ds^e + \boldsymbol{q^e} \end{gathered} \quad (2.18)$$

Substituting equation 2.3 in the first term:

$$\begin{gathered} h\int\int_{A^e} \boldsymbol{B^T D B} dA^e \boldsymbol{u} = \\ h\int\int_{A^e} \boldsymbol{B^T D \varepsilon_0} dA^e - h\int\int_{A^e} \boldsymbol{B^T \sigma_0} dA^e + \\ h\int\int_{A^e} \boldsymbol{b} dA^e + h\int_{s^e} \boldsymbol{t} ds^e + \boldsymbol{q^e} \end{gathered} \tag{2.19}$$

The terms in this equation can be expressed as follows:

$$\mathbf{K^e u^e} = [f^e_{\boldsymbol{\varepsilon}} - f^e_{\boldsymbol{\sigma}} + f^e_{\boldsymbol{b}} + f^e_{\boldsymbol{s}}] + \boldsymbol{q}^e = f^e + \boldsymbol{q}^e \tag{2.20}$$

After an assembling process of elemental matrices and vectors, a linear equation system is generated as follows:

$$\mathbf{Ku} = \mathbf{f} \tag{2.21}$$

Where **K** is the stiffness matrix, **u** is the displacements vector and **f** is the force vector. Solving this system, nodal displacements are calculated. Stresses can be calculated by using the displacements derivatives.

2.2.2 FEMT library

"FEMT is an open source muli-platform library and tools (Windows, Linux, and Mac OS) for solving large sparse systems of equations in parallel. This software is specially set to solve systems of equations resulting from the finite element, finite volume, and finite differences discretizations"(Miguel Vargas, 2014 [1]). FEMT has different methods developed for solving heat diffusion, electric potential, and solid deformation problems, all these modules work with the pre and post processor GID [2]. The general process carried out by GID-FEMT teamwork is the following:

- Pre-process: GID is capable of designing and meshing geometries, set materials, and boundary conditions. This data is known as pre-process information.

- All pre-process information is used by FEMT to solve a particular problem(heat, mechanical or electrical). Solution data is written in a specific format established by GID.

- Post process: GID visualizes the results.

For this work, we use the FEMT module for solid deformations, which computes nodal displacements and elemental Von Mises stresses.

2.3 Topology optimization

Structural optimization is a mechanical design process, whose objective is to find an optimal structure within a given design domain. The structure must satisfy service conditions, given by displacements and forces.

This problem can be solved by three different optimization approaches:

- Size optimization: the optimization variables are given by structural parameters, such as height, width, thickness, angles, etc.
- Shape optimization: the boundaries of an initial structure are modified in order to search for an optimal shape.
- Topology optimization: the design domain is discretized into elements and an optimization parameter is set for each one. Those parameters determine the existence of material on elements, allowing to set holes in the domain. The objective is to find a parameter configuration that defines an optimal distribution of the material over the domain.

In a topology optimization problem, it is necessary the solution of the elastic problem, it implies to solve the following system of linear equations:

$$\mathbf{K}\mathbf{u} = \mathbf{f} \tag{2.22}$$

Where $\mathbf{K}$ is the stiffness matrix, $\mathbf{u}$ is the displacement vector and $\mathbf{f}$ is the forces vector. The stiffness matrix $\mathbf{K}$ can be factorized as:

$$\mathbf{K} = E\hat{K} \tag{2.23}$$

Where E is the Young modulus and $\hat{K}$ is the factorized matrix. Young modulus is a scalar with the property that determines the stiffness of the material. It is valid factorizing the elemental stiffness matrices:

$$\mathbf{k_e} = E_e \hat{k}_e \tag{2.24}$$

Where E_e is the Young modulus of the element e. In a common elastic problem with homogeneous material $E_e = E$, this means, Young modulus is the same in every element, due to all elements have the same material. In contrast, in topology optimization, the elemental Young modulus could be different($E_e \leq E$), simulating that an element e has a lower material proportion(hence, a lower stiffness proportion) than a solid element. This is a numerical form to represent the absence of material in the element e.

To calculate E_e, each element is associated with a design variable x_e, which scales the Young modulus E_e as follows:

$$E_e = x_e E \tag{2.25}$$

Replacing (2.25) in (2.24), we get:

$$\mathbf{k_e} = x_e E \hat{k}_e = x_e \check{k}_e \tag{2.26}$$

Design variables x are the parameters in a topology optimization problem. They assign a gap to elements with $x_e = 0$ or a solid material to elements with $x_e = 1$. Some methods use binary design variables, others use them as continuous between 0 and 1, which allows assigning intermediate properties, no matter, a formal final solution must differentiate empty from solid material. Finding the optimal parameters x means to distribute adequately the material over the domain, which results in the optimal structure, as shown in figure 2.10.

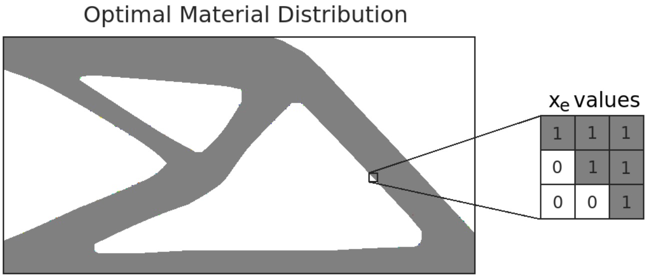


Figure 2.10: Resulting structure delivered by topology optimization method.

For real-world problems where the structure self-weight is considered, the material density(ρ) is also modified by the design variables. It is added to the force vector assembling:

$$\hat{\mathbf{f}} = \rho \hat{f} \tag{2.27}$$

Where $\hat{\mathbf{f}}$ is the vector of self-weight forces, which is formed by the elemental weight forces $\hat{\mathbf{f}_\mathbf{e}}$. The material density is scaled from $\hat{\mathbf{f}_\mathbf{e}}$ in a similar manner than the Young modulus:

$$\hat{\mathbf{f}_\mathbf{e}} = \rho_e \hat{f}_e \tag{2.28}$$

$$\rho_e = x_e \rho \tag{2.29}$$

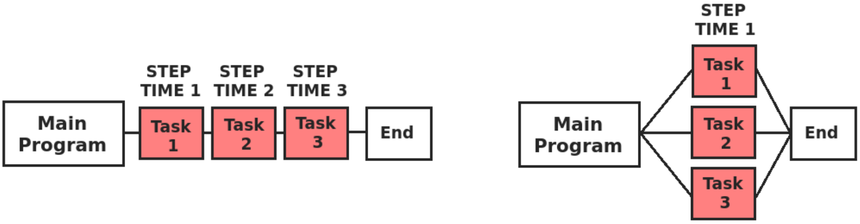
$$\hat{\mathbf{f}_\mathbf{e}} = x_e \rho \hat{f}_e = x_e f_e \tag{2.30}$$

This way, $x_e \in [0, 1]$ values determine the contribution of the elemental weight to the complete structural weight.

2.4 Parallel computing

In this section concepts about parallel computing are reviewed, as well as the schemes used in this work and the role of parallel computing in evolutionary algorithms. For a more detailed information about this topic consult [52], [132], [121], [96], [35], [54].

Parallel computing includes programming techniques that allow the execution of many instructions simultaneously, unlike the serial computing, where instructions are executed one after another. Processing units can not execute more than one instruction at the same time, so parallel computing is based on the sectioning a process and executing every section in different processing units, then achieve synchronization through the communication of their results [52]. This accelerates the data processing. It is important to ensure that algorithms can be sectioned for independent data, in order to avoid affecting the complete process. Figure 2.11 shows the difference between serial and parallel computing. Notice that in the second case every task is independent from the others.



(a) Example 1. Serial computing.

(b) Example 2. Paralell computing.

Figure 2.11: Serial and Paralell Computing

Parallel computing requires hardware with specialized architecture for its use. It can be *multi-core* computers or computer *clusters*, these architectures correspond to the parallelization schemes known as *shared memory* and *distributed memory* respectively.

2.4.1 Shared memory parallelization scheme

In this scheme all *cores*:(processing units) $\mathbf{P1}, \mathbf{P2}, \mathbf{P3}, ..., \mathbf{PN}$, have access to the same memory unit through a data bus. Current computers use this kind of architecture. In this scheme, any core can access to the information that another core is using, this makes easy the data communication between cores.

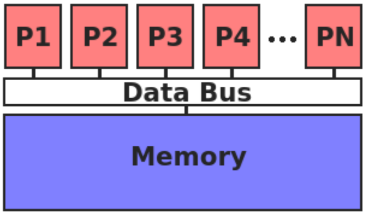


Figure 2.12: Example of a multi-core architecture

OpenMP is a standard for an API for this type of architecture. It is based on the use of compilation directives embedded in the code that allows parallelizing fragments of it, specially cycles [35]. OpenMP is frequently used for linear algebra operations, such as matrix and vector operations or solving linear equations systems, where many processes are independent between them. For example, the sum of two vectors $\hat{A} + \hat{B} = \hat{C}$, both $\hat{A}$ and $\hat{B}$ are sectioned in N parts corresponding to the same indexes, where N is the number of available cores, each core sums a chunk of indexes and save it in the corresponding indexes of $\hat{C}$.

OpenMP is used indirectly in this work, due to FEMT uses it for linear algebra operations.

2.4.2 Distributed memory work scheme

This parallelization scheme employs a *computer cluster*, which is a set of computers(*nodes*), that communicate each other through a high-velocity network, each node can be multi-core or not. Figure 2.13 shows an example. In the distributed memory scheme, each node has its own memory unit independent of the rest. This scheme gives to every node an identifier called *rank* and allows to all nodes execute the same program simultaneously. Each node does a specific task depending on its rank. All task together built a global process. So, it is necessary that all nodes keep a periodic communication. The way that a node can access to data in the memory of another one is by the sending and receiving messages.

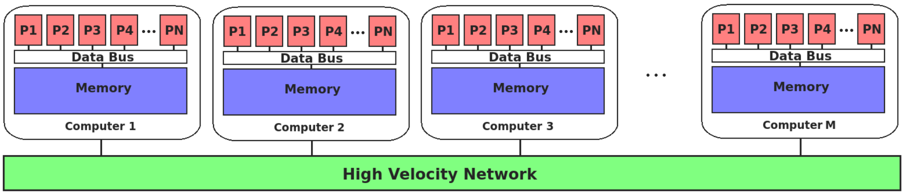


Figure 2.13: Example of a cluster architecture

Message Passing Interface(MPI) is a standard for a message passing system used for this scheme of parallelization. These libraries contain functions that allow an efficient communication between nodes through the high-velocity network [54]. A way to use this parallelization scheme is a model known as *master-slave*, where one of the nodes is assigned as *master* and all remaining are *slaves*. The master task is to execute the main structure of the algorithm, sectioning the most computational expensive tasks and sending them to available slaves for its execution.

The master-slave model is frequently used for evolutionary algorithms, where the master executes the structure shown in algorithm 1. In the Evaluate Population step, the master divides the complete population and sends every section to the available slaves via a message. Each slave receives a section of the population, then executes the evaluation and sends back the calculated fitness through another message to the master.

Evaluating a population usually is the most expensive process in evolutionary algorithms. In topology optimization problems, every individual evaluation requires solving an elastic problem via the finite element method.

A way to quantify the performance of a parallelized algorithm is comparing the execution time for the serial algorithm T_s with the processing time using a number of n cores or nodes depending on the case $T(n)$, this value is known as Speed Up. It determines how many times a parallel algorithm is faster than the serial one.

$$SU(n) = \frac{T_s}{T(n)} \tag{2.31}$$

Chapter 3

Thesis scope

3.1 Proposal of a numerical benchmark for topology optimization methods

Topology optimization algorithms require the following entry data: 1) the design domain and its mesh, 2) the service conditions: location and magnitudes, 3) the properties of the material and 4) the optimization model. In specialized literature, different research groups report similar tests. Those tests frequently differ in the entry data. This makes difficult to performs a fair comparison between the results of different algorithms due to differences in the tests. For the purpose of circunvent this issue, an objective of this work is the establishment of a benchmark for topology optimization in 2 dimensions, considering the similarities and statistics of the most frequently used tests in 103 papers in the specialized literature.

3.2 Solving local and global topology optimization problems for compliance and volume models

We have the interest in two optimization models: *compliance minimization with volume constraint*(related to optimal structural functionality) and *volume optimization with stress constraint*(related to optimal structural cost).

3.2.1 Compliance minimization with a volume constraint

This optimization model is given by the following objective function:

$$min\ C(x) = \mathbf{u}(\mathbf{x})^{\mathbf{T}}\mathbf{f}(\mathbf{x}) = \mathbf{u}(\mathbf{x})^{\mathbf{T}}\mathbf{K}(\mathbf{x})\mathbf{u}(\mathbf{x}) \tag{3.1}$$

Subject to the constraints:

$$\mathbf{K}(\mathbf{x})\mathbf{u}(\mathbf{x}) = \mathbf{f}(\mathbf{x}) \tag{3.2}$$

$$\frac{V(x)}{V_0} - V_r = \frac{\sum_{e=1}^{N} V_e x_e}{V_0} - V_r = G(x) = 0 \tag{3.3}$$

$$0 < x_{min} \leq x_e \leq 1 \tag{3.4}$$

Equation (3.1) is the evaluation function. It is formed by forces multiplying displacement. Work is the displacement generated by a force in a body, this means that we are searching a distribution of material that minimizes the work generated by the forces in the structure. Compliance is the inverse of stiffness, so this model also means to maximize stiffness. In equation (3.1) the vector $x = [x_1, x_2, ..., x_N]$ contains the design variables. $\mathbf{u}(\mathbf{x})$, $\mathbf{K}(\mathbf{x})$ and $\mathbf{f}(\mathbf{x})$ are the global displacements vector, stiffness matrix and forces vector, respectively calculated with the design variables x. Equations (3.2), (3.3), (3.4) show the constraints of the model.

- The first constraint (equation (3.2)) indicates that displacements $\mathbf{u}(\mathbf{x})$ must be calculated as the solution of the system formed by the stiffness matrix $\mathbf{K}(\mathbf{x})$ and the force vector $\mathbf{f}(\mathbf{x})$. These components are assembled according to the FEM.

- The second is the volume constraint (equation (3.3)), where $V(x)$ is the volume of the structure formed by the design variables x. It is calculated as $\sum_{e=1}^{N} V_e x_e$, where V_e is the elemental volume. V_0 is the volume of the solid full structure, V_r is the volume constraint parameter, it is a continuum value between $(0, 1]$ and represents the desired volume of the optimal rigid structure.
  The volume constraint is employed as $G(x) = \frac{V(x)}{V_0} - V_r = 0$.

- The third constraint(equation (3.4)), indicates that the design variables must take values between $[x_{min}, 1]$, where x_{min} is a small value close to 0 (for this work $x_{min} = 1E - 3$) for avoiding singularities in the linear equations system.

An important characteristic of the compliance objective function is that it avoid the formation of structures with unconnected elements, due to the objective function value is high for that kind of structures.

The minimal compliance structures are those with full solid material, due to compliance is calculated using displacements caused by forces, and displacements are higher when the forces are acting over a low amount of material. This is the reason why compliance optimization model has a volume constraint, if it does not, the optimal structure is the full material structure.

3.2.2 Volume minimization with a stress constraint

This optimization model is given by the following objective function:

$$min\ V(x) = \sum_{i=1}^{N} x_e V_e \tag{3.5}$$

Subject to the constraints:

$$\mathbf{K}(\mathbf{x})\mathbf{u}(\mathbf{x}) = \mathbf{f}(\mathbf{x}) \tag{3.6}$$

$$\sigma_e \leq \sigma_{YS} \quad e = [1, 2, ..., N] \quad \forall\ x_e > x_{min} \tag{3.7}$$

$$0 < x_{min} \leq x_e \leq 1 \tag{3.8}$$

Equation (3.5) is the evaluation function, where the values x_e and V_e are design variables and volumes of each element e in the design domain respectively, and N is the number of elements. This sum represents the total volume of the structure formed by variables x.

Equations (3.6), (3.7), (3.8) show the constraints of the model.

- The first constraint (equation (3.6)) indicates that displacements $\mathbf{u}(\mathbf{x})$ must be calculated as the solution of the system formed by the stiffness matrix $\mathbf{K}(\mathbf{x})$ and the force vector $\mathbf{f}(\mathbf{x})$. These components are assembled according to the FEM.

- The second is the stress constraint (equation (3.7)), where σ_e are elemental stresses and, σ_{YS} is the theoretical *yield stress*. This means that all elements in the structure must be within the elastic limit, ensuring that the structure is feasible. This constraint is applied only to solid elements.

- The third constraint(equation (3.8)), indicates that design variables must take values between $[x_{min}, 1]$, where x_{min} is a small value close to 0 (for this work $x_{min} = 1E - 3$) in order to avoid singularities in the linear equations system.

Minimal volume structures are those with no material, this is the reason why the model has a stress constraint. Considering that compliance structures are minimal with full solid material, compliance and volume are opposed objective functions.

3.2.3 Proposed solution for optimization models

The compliance model does not consider stress, in our opinion, this is not convenient for real-world problems because the resulting approximate optimal structures could be physically infeasible. No matter, compliance is an objective function that guide algorithms to find a distribution of material that built a highly rigid structure for a volume constraint. On the other hand, the volume model considers stress and ensures the physical feasibility of the result, but the objective function does not guide the algorithms to find an efficient material distribution. Compliance problems are solved with no other considerations than those in the model, no matter if results could be physically infeasible, while volume problems are solved considering compliance as well as the stress constraint.

In this thesis, local compliance optimization problems are solved with the Solid Isotropic Material with Penalization(SIMP) method. A detailed analysis of this method is carried out in section 5.1. A proposal for the reduction of computational cost for SIMP is introduced, as well as a modification for the solution of volume optimization problems with stress constraint(SIMP-SC).

Global optimization problems are solved using evolutionary algorithms. Evolutionary algorithms for topology optimization require high computational resources due to the high dimension of the problems and the complexity of the function evaluation. These inconveniences suggest that evolutionary algorithms could not be convenient for this kind of problems, no matter, evolutionary algorithms are capable of approximating global optimums. The proposals for circumventing the computational cost issues are, 1) an adequate individual representation, in order to reduce the dimension of the problem, and 2) the use of parallel computing to distribute the computational cost that structural evaluation implies. Mono-objective optimization algorithms are used to solve compliance problems and multi-objective algorithms are used to solve both, compliance and volume problems.

Chapter 4

Proposal of a numerical benchmark for topology optimization

Despite the many categories of topology optimization algorithms and optimization models [41],[33], most of them use the finite element method to evaluate candidate solutions; then all of these algorithms require the previous specification of the following problem properties:

- **Design domain.** Search space dimensions and shape.
- **Boundary conditions.** Fixed displacements and load conditions:
  - **Zero displacement conditions.** Regions where the structure is fixed.
  - **Load conditions.** Regions where the structure is affected by external forces. It is necessary to establish the position and magnitude.
  - **Self-body forces.** The most common is the self-weight force, but there could exist external forces dependent on the geometry of the body.
- **Properties of the material.** Properties of the material used to make numerical simulations: *1*)Young modulus *2*) Poisson modulus

3) maximal permissible stress or yield stress(YS). Even if the yield stress is not used for the simulation, it can be used to determine whether the material is working within the elastic range; then it must be used not only to determine if the structure is physically feasible but also to know if the numerical simulation using elastic theory is valid.

- **Mesh.** Type, number, and dimensions of elements used to mesh the domain, considering that the numerical error of the simulation depends on the mesh.

Topology optimization researchers from different groups [4, 141, 13] solve, in many cases, similar problems using different design domains, boundary conditions, properties of the material and meshes, which could cause the following issues:

- It is difficult to perform fair comparisons between algorithms and its implementations.
- Some articles report results using properties of materials that does not exist or without considering self-weight, which is not convenient for the optimization of realistic structures.
- Using unrealistic materials and service conditions is a major concern for an adequate method, because most of the researchers ([55],[154],[141]), assume elastic properties in the material, but they do not verify if the material is actually working in the elastic range, even more, the problem formulation does not considers a constraint to verify that the candidate structure is in the elastic range neither they provide a real-world material where the elastic range is well defined.

Our proposal intends to alleviate the aforementioned concerns by introducing a well-defined benchmark. Other benchmark problems have been proposed; for instance, Rozvany had proposed a benchmark in a paper called: *Exact analytical solutions for some popular benchmark problems in topology optimization* [116]. In this benchmark, the analytic solutions for a set of topology optimization problems are computed. These problems are based on the optimization of a configuration of bars(determining positions, sizes, etc.). However, this article proposes a benchmark for optimizing the distribution of material on a domain. So, the benchmark of Rozvany is not comparable with our benchmark because it cannot be used to compare algorithms of a similar kind.

The objective on this part of the thesis is to establish a set of problems for topology optimization, that detail all the characteristics mentioned above as well as realistic problem configurations. Using this benchmark, we can ensure that comparable algorithms are solving exactly the same problems, that the assumptions carried out by the simulations (elasticity) are true, and that the final solutions consider real-world conditions and materials. Therefore, we can perform fair comparisons among different approaches.

4.1 Methodology for selecting the benchmark problems

We have gathered test problems from a set of 103 articles in the specialized literature. The frequency of similar problems has been computed and grouped according to the following characteristics:

- Boundary conditions. The most frequent boundary conditions regarding the distance between fixed lines-areas and loads as well as loads directions.

- The geometrical shape of the search domain, that is to say, rectangular, quadrilateral, etc.

- Dimensions. The most frequent dimensions and proportions among them.

The 13 most frequent problems are used significantly more than the others. Usually, their geometrical shape is exactly defined, but the relative position of the loads is only approximately specified. Most of the times, they neither define exactly the size of fixed segments of the geometry nor do they specify whether the loads are applied to a single point or to a line. In addition, neither the thickness nor the system of units are reported, and the properties of the material do not correspond to any real-world material.

To circumvent these issues, we propose the following:

- We set the units to the standard International System; lengths are in meters (**m**), force units are Newtons (**N**), Young modulus and yield stress are given in Pascals, and Poisson modulus is dimensionless.

- We set the side's lengths to those most frequently reported.

- We set the thickness to 1 percent of the maximal length size in order to standardize the thickness and to fulfill the hypothesis of planar stress.

- The external loads or forces are applied in a line whose length is 10% of the length of the side they are applied to, in order to avoid numerical issues that arise with single-point loads.

- The properties of the material are set as those given by steel ASTM A-36, whose properties are similar to the most common materials used in real-life structures and in academic problems.

4.2 Statistics

In Tables 4.1, 4.2, 4.3 and 4.4 the obtained data from the 103 consulted articles are condensed. The columns in tables report the following information:

- Column 1. Analyzed article. The citation to the proper article referenced at the bibliography.

- Column 2. Type of problem. Common problem types according to their geometrical properties; they can be categorized as Cantilever, Short Cantilever, LShape, MBBB, Two Bars and Michell. Some Michell problems are marked as Michell*, this means that they are defined with two loads as boundary conditions as shown in subfigures (b) and (c) in Figure 4.4.

- Column 3. Load region. In Cantilever, Short Cantilever and LShape tests, the load position sometimes varies at Top, Center or Bottom of the load side as shown in figures 4.1, 4.2 and 4.3. In Two Bars, MBBB and Michell tests, load positions never vary as shown in figures 4.4, 4.5 and 4.6.

- Column 4. Load magnitude. For Michell* tests(multi-load) every load is presented separated with a "/".

- Columns 5, 6 and 7. Lengths of the initial domain. M, L are the side lengths, as shown in Figures 4.1,4.2,4.3,4.4,4.5 and 4.6. t is the thickness used. Thickness cells with "v" represent volume, so these tests are in 3D.

- Column 8. Fixed regions. An asterisk means that fixed segments of the test problems reported in the article are the same as those fixed segments in the test problems of the proposed benchmark.

A sign - indicates that there is no information about the problem settings.

Reference	**Test Type**	**Load region**	**Load Size**	M	L	t	**Hold Type**
[30]	Cantilever	Central	3000	6.4	4	0.1	*
[32]	Short Cantilever	Bottom	-	250	250	-	*
	Short Cantilever	Bottom	-	20	20	20(v)	*
[108]	Cantilever	Central	-	30	10	10(v)	*
	Cantilever	Central	-	20	10	10(v)	*
	Lshape	Top	-	12	8	5(v)	
	MBBB	-	-	30	10	10(v)	
[114]	Michell	-	-	1	1	-	
	Short Cantilever	Top	-	1	1	-	
[135]	Cantilever	Central	10000	0.4	0.1	-	*
	Cantilever	Central	2000	0.08	0.02	0.02(v)	*
[145]	Cantilever	Bottom	1	100	50	-	*
	Two Bar	Central	1	80	40	-	*
	Michell*	-	1/1	50	50	-	*

Table 4.1: Statistics from topology optimization literature.

Article	**Test Type**	**Load region**	**Load Size**	M	L	t	**Hold Type**
[149]	MBBB	-	1	-	-	-	
	Michell	-	1	-	-	-	
[153]	-	-	-	-	-	-	
[157]	-	-	-	-	-	-	
[167]	Michell	-	1	1	1	-	*
	Short Cantilever	Bottom	1	1	1	-	*
[168]	MBBB	-	-	-	-	-	*
[4]	Cantilever	Central	1	2	1	-	
	Tshape	-	1	120	80	-	
	Michell	-	1	1	1.2	-	
	Cantilever	Central	1	5	3	2.4(v)	*
	Tshape	-	-	-	-	-(v)	*
	Lshape	Central	-	-	-	-(v)	*
	MBBB	-	-	-	-	-(v)	*
[148]	Cantilever	Central	0.5	-	-	-	*
	Cantilever	Central	1.5	-	-	-	*
[31]	Lshape	Central	1	0.6	0.4	-	*
[71]	Lshape	Central	1	0.6	0.4	-	*
[58]	Cantilever	Central	1	2	1	-	*
	Michell*	-	1/1	1	1	-	
[70]	Lshape	Top	500	0.06	0.04	-	*
	Cantilever	Central	1500	0.2	0.1	-	*
[68]	Lshape	Top	250	0.06	0.04	-	*
	Cantilever	Central	900	0.2	0.1	-	*
[79]	-	-	-	-	-	-	
[136]	Cantilever	Central	144000	1	0.25	0.1	*
	Cantilever	Central	240000	1	0.25	0.1	*
	Cantilever	Central	300000	1	0.25	0.1	*
	Cantilever	Central	500000	1	0.25	0.1	*
[126]	MBBB	-	1	-	-	-	
	Cantilever	Bottom	1	-	-	-	*
[124]	Lshape	Top	8000	0.06	0.04	-	*
[134]	Short Cantilever	Bottom	-	2	2	-	*
[143]	Cantilever	Bottom	1	20	10	-	*
[147]	MBBB	-	1	-	-	-	
	Michell	-	1	-	-	-	
[163]	Cantilever	Central	1	2	1	-	*
[152]	Short Cantilever	Bottom	1	1	1	-	*
[76]	Cantilever	Top	21000	-	-	-(v)	*
	Cantilever	Central	-	-	-	-(v)	*
	Cantilever	Bottom	-	-	-	-(v)	*

Table 4.2: Statistics from topology optimization literature

Article	**Test Type**	**Load region**	**Load Size**	M	L	t	**Hold Type**
[53]	Cantilever	Central	1.2	-	-	-	*
	Cantilever	Central	1	-	-	-	*
	Cantilever	Central	0.8	-	-	-	*
[57]	Two Bar	-	1	2	1	-	*
	Lshape	Central	1	0.6	0.4	-	*
[73]	-	-	-	-	-	-	
[82]	Lshape	Top	-	-	-	-	
[88]	Two Bar	-	1	0.9	0.3	-	*
	Lshape	Top	2.5	0.6	0.4	-	*
[105]	Cantilever	Top	500	2	1	-	*
[100]	Cantilever	Central	150	40	20	-	*
[107]	Cantilever	Central	2000000	2	1	0.2	*
[109]	MBBB	-	1	3	1	-	
[117]	-	-	-	-	-	-	
[128]	Short Cantilever	Bottom	100	1	1	0.001	
	Cantilever	Central	100	1.6	1	0.001	
	Lshape	Central	100	0.6	0.4	0.001	
[133]	Cantilever	Bottom	1	2	1	-	*
	Michell	-	1	1	1	-	*
[150]	Cantilever	Central	1	0.2	0.1	-	*
	Cantilever	Bottom	1	0.15	0.1	-	*
	Cantilever	Bottom	1	0.09	0.06	0.03(v)	*
[156]	Cantilever	Central	-	4	2	-	
	MBBB	-	-	-	-	-	*
[159]	-	-	-	-	-	-	
[12]	Lshape	Top	40	1.5	1	-	*
[49]	Cantilever	Central	-	4	1	1(v)	*
	Cantilever	Central	-	2	1	-	*
	Michell*	-	-/-	2	1	-	
[131]	Cantilever	Central	1	80	40	-	*
	Cantilever	Bottom	1	80	40	-	*
[67]	Lshape	Central	-	-	-	-	*
[72]	Cantilever	Central	1	4	2	-	*
	MBBB	-	1	6	2	-	
[87]	Two Bar	Central	200	0.4	0.1	0.001	*
	Lshape	Central	20000	0.06	0.04	-	*
[89]	Cantilever	Central	1	60	30	-	*
	Michell	-	1	60	30	-	
	MBBB	-	1	90	30	-	
[103]	Cantilever	Central	-	32	16	-	*
	Cantilever	Central	-	24	12	12(v)	*
	MBBB	-	-	-	-	-	*

Table 4.3: Statistics from topology optimization literature

Article	**Test Type**	**Load region**	**Load Size**	M	L	t	**Hold Type**
[151]	Lshape	Central	1	0.6	0.4	-	*
	Lshape	Top	1	0.6	0.4	-	*
	Michell	-	1	1	1	-	*
[75]	Cantilever	Central	500000	0.4	0.25	-	*
	Cantilever	Central	-	0.6	0.3	0.2(v)	*
[56]	MBBB	-	1	3	1	-	
	Cantilever	Central	1	1.5	1	-	*
	Lshape	Central	1	0.6	0.4	-	*
[165]	Michell*	-	-/-	120	120	-	*
[74]	Cantilever	Central	1	40	25	-	*
	Lshape	Top	1	20	20	-	*
	MBBB	-	1	60	20	-	
[84]	Cantilever	Central	3000	0.16	0.1	0.001	
	Cantilever	Bottom	3000	0.16	0.1	0.001	
	Cantilever	Top	3000	0.16	0.1	0.001	
[115]	Cantilever	Central	30000	0.16	0.1	0.02	*
	Michell*	-	1000 /1000	0.5	0.5	0.02	
[127]	Cantilever	Bottom	100	0.16	0.08	-	*
[55]	Cantilever	Central	1	40	25	-	
	Cantilever	Central	1	40	25	25(v)	
	MBBB	-	1	60	20	-	
[65]	MBBB	-	100	0.05	0.05	-	
	MBBB	-	1	1	0.04	-	
[110]	Cantilever	Central	10000	3	1	0.001	
	Cantilever	-	10000	1	0.5995	0.001	
[158]	Cantilever	Central	1000	10	5	-	*
	Cantilever	Central	1000	4	2	2(v)	*
[164]	Cantilever	Bottom	-	240	150	-	*
	MBBB	-	-	120	40	-	
[85]	Cantilever	Top	3000	0.16	0.1	0.001	
	Cantilever	Central	3000	0.16	0.1	0.001	
	Cantilever	Bottom	3000	0.16	0.1	0.001	
	Cantilever	-	3000	0.16	0.1	0.001	
[91]	Cantilever	Central	450	1.2	0.6	-	*
	Michell	-	1	2	1	-	
[15]	Short Cantilever	Central	-	1	1	-	
	Cantilever	Central	-	2	1	-	
[102]	Short Cantilever	Top	40	10	10	-	*
	Short Cantilever	Central	40	10	10	-	*
	Michell	-	-	100	100	-	
[37]	Two Bar	-	-	2	1	-	*
	Michell	-	-	100	100	-	

Table 4.4: Statistics from topology optimization literature

Article	**Test Type**	**Load region**	**Load Size**	M	L	t	**Hold Type**
[28]	Cantilever	Top	-	2	1	-	*
	MBBB	-	-	2	1	-	*
[81]	-	-	-	-	-	-	
[86]	Michell	-	1	1	1	-	
	Cantilever	Bottom	1	2	1	-	*
[90]	-	-	-	-	-	-	
[154]	Cantilever	Central	1	2	1	1	*
	Michell	-	1	1	1.2	1	
[80]	Lshape	Central	-	-	-	-	
[141]	Cantilever	Central	1	2	1	-	*
	Michell	-	1	1	1.2	-	*
[13]	Cantilever	Central	1	2	1	-	*
	Michell	-	1	1	1.2	-	
	Michell*	-	1/1	1	1.2	-	
	Tshape	-	1	4	6	-	*
[26]	Cantilever	Top	-	2	1	-	*
	MBBB	-	-	3	1	-	*
	Michell	-	-	1.5	1	-	*
[92]	Cantilever	Central	-	2	1	-	*
[122]	Two Bar	-	1	12	4	-	*
	Lshape	Central	1	3	3	-	*
[142]	Cantilever	Central	1	2	1	-	*
	Two Bar	Central	1	2	1	-	*
[6]	Cantilever	Central	-	2	1	-	*
	Michell*	-	1/1	1	1.2	-	
[139]	Michell*	-	30/15	6	6	-	*
	Cantilever	Bottom	80	3.2	2	-	*
[137]	MBBB	-	1	3	1	1	
	Two Bar	-	1	24	10	1	*
	Cantilever	Central	1	5	3	1	*
[10]	Cantilever	Central	1	2	1	-	*
	Michell	-	1	1	1.2	-	
	Cantilever	Central	1	5	3	2.4(v)	*
	Tshape	-	-	-	-	-	*
[95]	Cantilever	Top	1000000000	4	2	1	*
	MBBB	-	1	3	1	-	
[138]	Two Bar	Central	80	2	1	-	*
	Michell*	-	30/15	6	6	-	*
[25]	Cantilever	Central	10	0.2	0.05	-	*
	MBBB	-	10	0.3	0.05	-	
[8]	Cantilever	Central	-	-	-	-	*
	Two Bar	Central	-	-	-	-(v)	*

Table 4.5: Statistics from topology optimization literature

Article	**Test Type**	**Load region**	**Load Size**	M	L	t	**Hold Type**
[60]	Lshape	Central	-	-	-	-	*
	Cantilever	Top	-	-	-	-	*
[125]	MBBB	-	-	3	1	-	
[162]	Cantilever	Bottom	800	34	22	-	*
	Michell*	-	300/150	6	6	-	*
	Two Bar	-	800	60	25	-	*
	Michell	-	800	60	60	-	*
	Michell	-	800	60	60	-	
	MBBB	-	80	12	4	-	*
[140]	Two Bar	-	40000	2	1	-	*
	Cantilever	Bottom	-	-	-	-	*
	MBBB	-	20000	1.2	0.4	-	*
	Michell*	-	30/5	-	-	-	*
[22]	Cantilever	Central	1	3.2	2	-	*
	MBBB	-	1	3	1	-	
[36]	-	-	-	-	-	-	
[106]	Cantilever	Central	500	0.48	0.08	0.08(v)	*
[120]	Cantilever	Central	-	8	5	-	*
	MBBB	-	-	3	1	-	*
[34]	Cantilever	Central	-	1.5	1	1(v)	*
	Cantilever	Bottom	-	1.5	1	0.4(v)	*
[19]	Cantilever	Central	-	-	-	-	*
[116]	Cantilever	Central	-	-	-	-	*
	MBBB	-	-	-	-	-	
[123]	Cantilever	Central	10	80	50	-	
	Michell*	-	10/10	20	20	1	
	Michell	-	10	20	20	1	
	MBBB	-	10	500	200	-	
[50]	Cantilever	Central	-	-	-	-	*
[18]	Cantilever	Central	-	-	-	-	*
	Cantilever	Bottom	-	-	-	-	*

Table 4.6: Statistics from topology optimization literature

4.3 Proposed benchmark problems

In this section we provide and analyze the frequencies of different geometries, service conditions and properties which are the basis for the proposed benchmark. First, we analyse the frequency of domain dimensions, fixed regions and loads, based on information gathered from the corresponding articles. Using this information, we propose a benchmark of problems for the 13 most common configurations.

4.3.1 Load magnitudes and properties of the material

Loads

Loaded segments have been established in Figures 4.1,4.2,4.3,4.4,4.5 and 4.6 using literals P and Q. The magnitude of the loads is defined in Table 4.7.

Test	**Test type**	**P**	**Q**
Cantilever(Figure 4.1)	Load at Center(CLC)	8.6e4 N	-
	Load at Bottom(CLC)	6.2e4 N	-
	Load at Top(CLC)	6.2e4 N	-
Short Cantilever(Figure 4.2)	Load at Center(SCLC)	9.0e4 N	-
	Load at Bottom(SCLB)	5.8e4 N	-
	Load at Top(SCLT)	5.8e4 N	-
LShape(Figure 4.3)	Load at Center(LLC)	1.5e4 N	-
	Load at Top(LLT)	1.5e4 N	-
Michell(Figure 4.4)	One Load(OLM)	5.6e4 N	-
	Two Equal Loads(TELM)	2.8e4 N	-
	Two Different Loads(TDLM)	3.72e4 N	1.86e4 N
MBBB(Figure 4.5)	-	2.7e4 N	-
Two Bars(Figure 4.6)	-	15.3e4 N	-

Table 4.7: Loads values for tests

All the loads are uniformly distributed among 10% of the length of the side to which they are applied. The loads are set in such a way that the *Security Factor(SF)*, calculated with the full domains (initial structures), is approximately *0.75*. Therefore, we can be sure that every initial structure is feasible and that a realistic load is set, that is to say, that the work performed by the structure is close to its limit. SF is the value given by $SF = \frac{\sigma_{max}}{\sigma_y}$ where σ_{max} is the maximal von Mises stress in the structure, and σ_y is the yield stress which is a material property.

Properties of the material

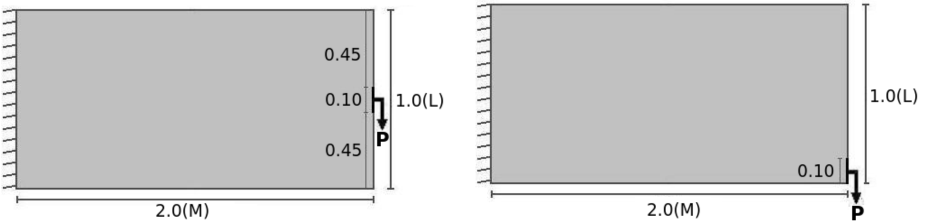
For the proposed test problems the material used is the ASTM A-36 steel, one of the most commonly used material in real-world structural mechanics.

- Young modulus = 2.11E11 Pa
- Poisson ratio = 0.29
- Density = 7874 kg/m3
- Yield stress = 2.2E8 Pa

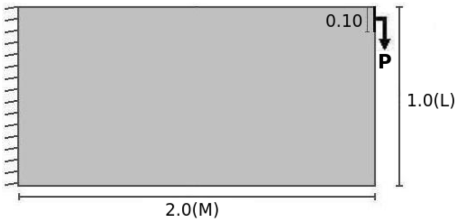
4.3.2 Domains, dimensions and boundary conditions

Cantilever

Figure 4.1 shows a *Cantilever* domain and boundary conditions in three different fashions. This test problem is reported in 67 of the 103 articles reviewed. In these 67 articles, there are 91 different *Cantilever* tests problems. In 64 of the 91 tests, the load is at the center on the right side as shown in Figure 4.1(a). In 19 tests it is at the bottom on the right side as shown in Figure 4.1(b). In 8 tests, it is at the top on the right side as shown in Figure 4.1(c). In 19 tests, the dimensions are the same as those in Figure 4.1, and in 40 tests, the dimensions have the same relation $(R_1 : R_2) = (1 : 2)$. In 79 tests, the displacement-conditioned segment is the same as the one proposed in this benchmark.



(a) Cantilever Load at Center (CLC) (b) Cantilever Load at Bottom (CLB)

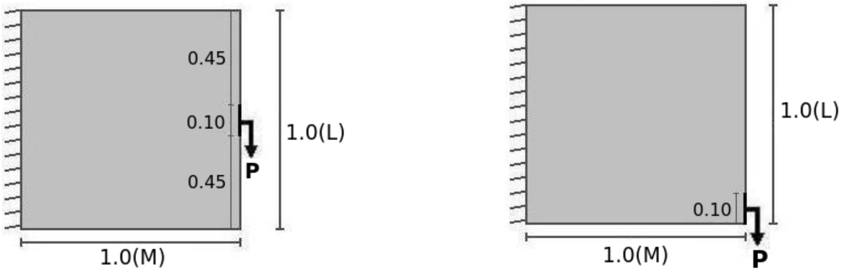


(c) Cantilever Load at Top (CLT)

Figure 4.1: Cantilever tests.

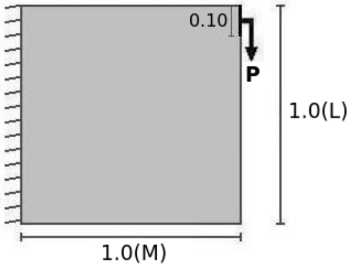
Short Cantilever

Figure 4.2 shows the *Short Cantilever* domain and boundary conditions. This test problem is reported in 8 of the 103 articles reviewed. In these 8 articles, there are 10 different *Short Cantilever* test problems. In 2 of these 10 tests, the load is at the center on the right side as shown in Figure 4.2(a). In 6 tests it is at the bottom on the right side as shown in Figure 4.2(b). In 2 tests, it is at the top on the right side, as shown in Figure 4.2(c). In 5 tests the dimensions are the same as those in Figure 4.2, and in 10 tests the dimensions have the same relation $(R_1 : R_2) = (1 : 1)$. In 7 tests, the displacement-conditioned segment is the same as the one proposed in this benchmark.



(a) Short Cantilever Load at Center (SCLC)

(b) Short Cantilever Load at Bottom (SCLB)

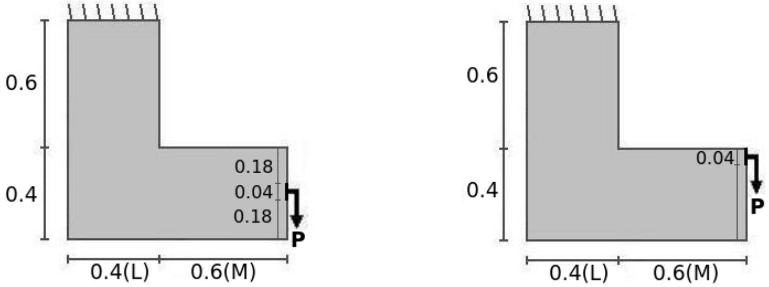


(c) Short Cantilever Load at Top (SCLT)

Figure 4.2: Short cantilever tests.

L Shape test

Figure 4.3 shows the *L Shape* domain and boundary conditions. This test is reported in 20 of the 103 articles reviewed. In these 20 articles, there are 21 different *L Shape* tests problems. In 12 tests, the load is at the center on the right-bottom side, as shown in Figure 4.3(a). In 9 tests, it is at the top on the right-bottom side, as shown in Figure 4.3(b). In 8 tests, the dimensions are the same as those in Figure 4.3, and in 13 tests, the dimensions have the same relation $(R_1 : R_2) = (0.6 : 0.4)$. In 17 tests, the displacement-conditioned segment is the same as the one proposed in this benchmark.

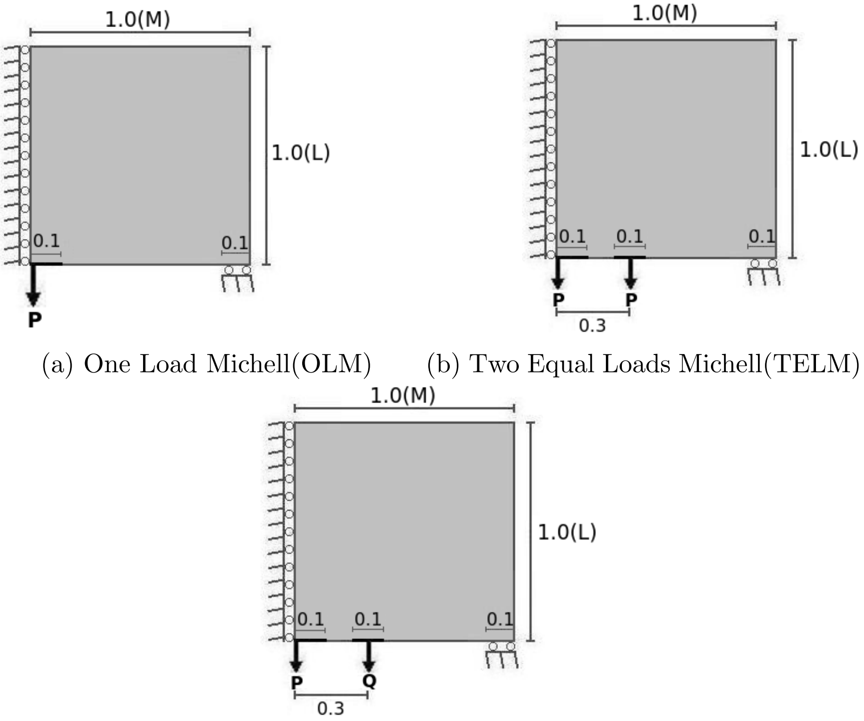


(a) LShape Load at Center(LLC) (b) LShape Load at Top(LLT)

Figure 4.3: LShape tests.

Michell

Figure 4.4 shows the *Michell* domain and boundary conditions. This test problem is reported in 28 of the 103 articles reviewed. In 28 articles there are 32 different *Michell* problems tests. In 20 of these 32 tests, there is only one load on the left-bottom side, as shown in figure 4.4(a). In 7 tests there are two equal loads, as shown in figure 4.4(b). In 5 tests, there are two different loads, as shown in Figure 4.4(c). In 7 tests, the dimensions are the same as those in Figure 4.4, and in 18 the dimensions have the same relation $(R_1 : R_2) = (1 : 1)$. In 20 tests, the displacement-conditioned segment is the same as the one proposed in this benchmark.



(a) One Load Michell(OLM)

(b) Two Equal Loads Michell(TELM)

(c) Two Different Loads Michell(TDLM)

Figure 4.4: Michell tests.

MBBB

Figure 4.5 shows the *MBBB* domain and boundary conditions. This test problem is reported in 28 of the 103 articles reviewed. 29 different *MBBB* problem tests are performed in the 28 articles. In all 29 tests, fixed and loaded areas are located in the same position in the geometry. In 8 tests, the dimensions are the same as those in Figure 4.5, and in 16 tests, the dimensions have the same relation $(R_1 : R_2) = (1 : 3)$. In 20 tests, the displacement-conditioned segment is the same as the one proposed in this benchmark.

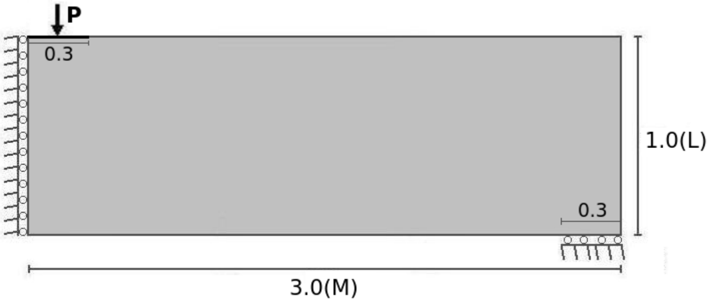


Figure 4.5: MBBB test

Two Bars

Figure 4.6 shows the *Two Bars* domain and boundary conditions. This test problem is reported in 12 of the 103 articles reviewed. There are 12 different *Two Bars* problem tests. In all 12 tests, fixed and loaded areas are located in the same position in the geometry. In 5 tests, the dimensions are the same as those in Figure 4.6, and in 6 tests, the dimensions have the same relation $(R_1 : R_2) = (1 : 2)$. In 12 tests, the displacement-conditioned segment is the same as the one proposed in this benchmark.

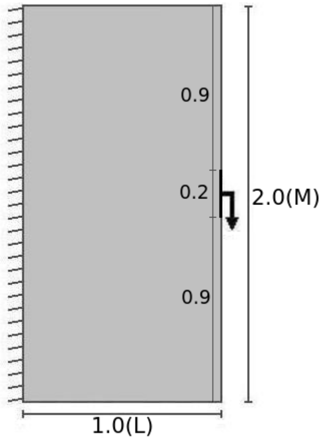


Figure 4.6: Two Bars tests

In the next section, we propose a set of meshes for the FEM that could be used to approximate the displacements and stresses of the structures.

4.3.3 Meshes

Figures 4.7, 4.8, 4.9, 4.10, 4.11 and 4.12 show meshes designed for each test problem. Every side of each figure shows the number of elements into which it is divided. The meshes were defined according to the following criteria:

- They must use exact square elements. In general, it is assumed that the final shape is not known, square elements represent a kind of equilibrium for numerical error caused by the element shape.

- Nodes positions must permit the exact setting of the boundary conditions, that is to say, one must be able to apply the load in the exact segment where it is defined.

- All meshes must have at least 10 000 elements in order to produce adequate numerical results and maintain a relatively short computational solving time.

Using the shortest length side as a basis, and with the purpose of accomplishing the above conditions, we use the following algorithm:

1. Set a partition parameter $p = 1$.
2. The side of minimal length is partitioned by $10p$.
3. The other sides are partitioned so that square elements are generated.
4. If the number of elements is less than 10 000, increase $p = p + 1$ and go to step 2. Otherwise, the procedure terminates.

Cantilever

Figure 4.7 shows the mesh for the Cantilever domain. The number of elements/partitions for each side is indicated. For all of the Cantilever test problems we propose to use 12 800 square elements with an area of 0.0125x0.0125 = 0.000156 m^2. A uniformly distributed load P is applied to 9 nodes.

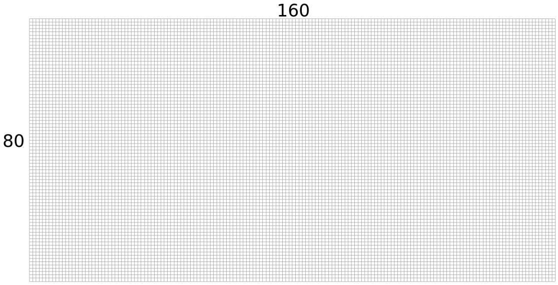


Figure 4.7: Cantilever mesh

Short Cantilever

Figure 4.8 shows the mesh for the Short Cantilever domain. The number of elements/partitions for each side is indicated. For all of the Short Cantilever test problems we propose to use a mesh with 10 000 elements. Every square element has an area of 0.01x0.01 = 0.0001 m^2. A uniformly distributed load P is applied to 11 nodes.

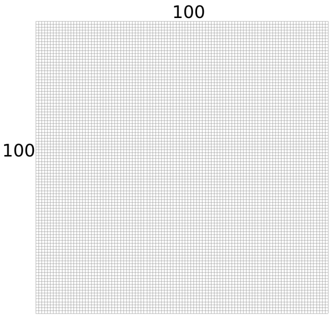


Figure 4.8: Short Cantilever mesh

L Shape

Figure 4.9 shows the mesh for the L-shape domain. The number of elements/partitions for each side is indicated. For all the LShape tests we propose a mesh with 14 400 elements. Every square element has an area of 0.01x0.01 = 0.0001 m^2. A uniformly distributed load P is applied to 7 nodes in each test.

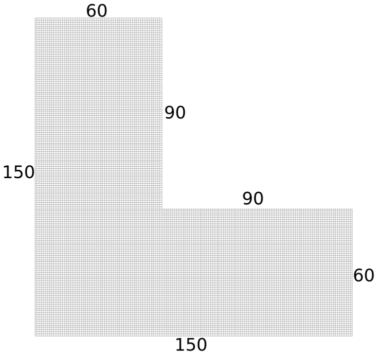


Figure 4.9: LShape mesh

Michell

Figure 4.10 shows the mesh for the Michell domain. The number of elements/partitions for each side is indicated. For all the Michell tests we propose a mesh with 10 000 elements. Every square element has an area of 0.01x0.01 = 0.0001 unit^2. Uniformly distributed loads P and Q can be applied to 11 nodes in each test.

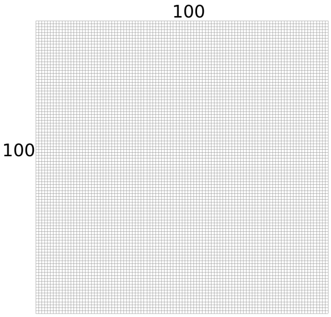


Figure 4.10: Michell mesh

MBBB

Figure 4.11 shows the mesh for the MBBB domain. The number of elements/partitions for each side is indicated. We propose to use 10 800 square elements with an area of 0.01666x0.01666 = 0.0002777 m^2. A uniformly distributed load P is applied to 19 nodes.

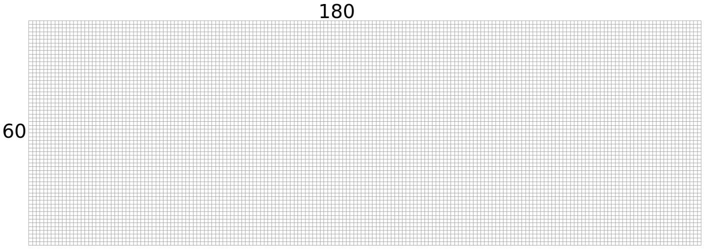


Figure 4.11: MBBB mesh

Two Bars

Figure 4.12 shows the mesh for the Two Bars domain. The number of elements/partitions for each side is indicated. We propose to use 12 800 square elements with an area of 0.0125x0.0125 = 0.000156 m^2. A uniformly distributed load P is applied to 17 nodes.

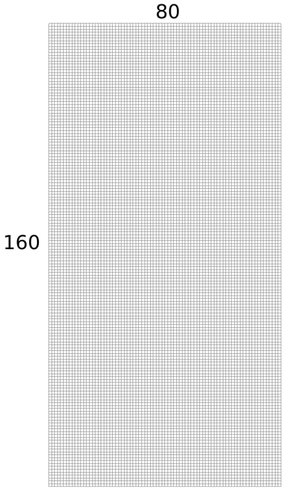


Figure 4.12: Two Bars mesh

Chapter 5

SIMP method: analysis and proposal of modifications

The Solid Isotropic Material with Penalization(SIMP) is one of the most popular methods for topology optimization, developed by M. P. Bendsoe [17] in 1989 and strongly studied since then. In this section of the work, a detailed analysis of this method is presented. In addition, two modification to the SIMP are introduced: 1) a new convergence criterion that considerably reduces the computational cost, and 2) a SIMP version which could be used to solve the minimal volume problem.

5.1 Analysis of the SIMP method

SIMP is an homogenization method where every element has a particular micro-structure. A macro-parameter x_e is assigned to the micro-structure of every element e, it can take values in the range $[x_{min}, x_{max}]$ (where x_{min} is around to $1E-3$ and $x_{max} = 1$). The x_e value corresponds to the material proportion in the micro-structure of the element e, making possible the simulation of the structure under different material distributions. Macro-parameters Young modulus E and material density ρ can be modified according to the formulas: $E_e = x_e^p E$ and $\rho_e = x_e^p \rho$. An element with a lower material proportion is not as stiff as a solid one. The parameter p is called *power penalization*, its objective is penalizing intermediate values of x_e in function of its own value, carrying out a polynomial penalization as shown in graph of figure 5.1, where a small change on material proportion x_e results a high penalization in stiffness and density of the element e. Graph on figure 5.1 shows the performance of $p = [1, 2, 3, 4]$. For the numerical simulations in this work $p = 3$.

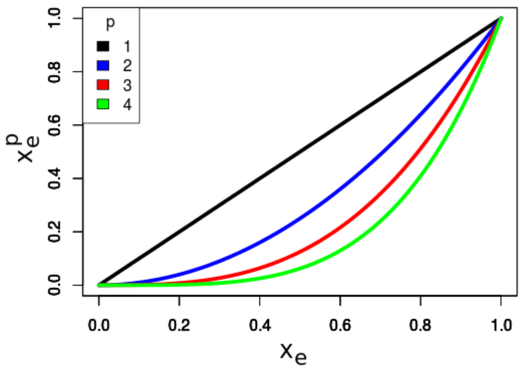


Figure 5.1: Behavior of power penalization parameter

SIMP is a non-population based method for compliance problems. It starts over an initial solution and updates the design variables, searching for better solutions iteratively. The updating formula for the design variables is based on mathematical and heuristic approaches. The mathematical approach computes the objective function gradient and applies a regularized fixed point method to get the updating equation, which leads to values of x_e where the gradient is zero. The heuristic approach is composed by some steps that modify the x_e values obtained by the updating equation in order to guide the algorithm through "smoother" solutions.

5.1.1 Derivation of the variable update step

In the compliance optimization model, if the problem considers self-weight, the force vector $\mathbf{f}$ depends on design variables x, that is to say $\mathbf{f} = \mathbf{f}(\mathbf{x})$. If self-weight is not considered, $\mathbf{f}$ is constant and the compliance model in equation (3.1) can be written an follows:

$$min\ C(x) = \mathbf{u(x)^T f} = \mathbf{u(x)^T K(x) u(x)}. \tag{5.1}$$

According to the first constraint in equation (3.2):

$$\mathbf{K(x)u(x)} = \mathbf{f}. \tag{5.2}$$

In SIMP method, the objective function in equation (5.1) is augmented considering the volume constraint in equation (3.3) through a Lagrangian:

$$min\ J(x) = C(x) + \lambda G(x). \tag{5.3}$$

The derivative of the function (5.3) is calculated by:

$$\frac{\partial J(x)}{\partial x_e} = \frac{\partial C(x)}{\partial x_e} + \lambda \frac{\partial G(x)}{\partial x_e}. \tag{5.4}$$

Where the first term is obtained from the derivative of equation (5.1) respect to the design variables x_e:

$$\frac{\partial C(x)}{\partial x_e} = \mathbf{u'(x)^T f} = \mathbf{u'(x)^T K(x) u(x)}. \tag{5.5}$$

Deriving equation (5.2)(considering that $\mathbf{f}$ is constant):

$\mathbf{K'(x)u(x)} + \mathbf{K(x)u'(x)} = 0.$

We get the equality:

$\mathbf{K(x)u'(x)} = -\mathbf{K'(x)u(x)}.$

This equality allows to calculate the derivative $\mathbf{u'(x)}$ in terms of the derivative $\mathbf{K'(x)}$, which is easier to calculate considering the factorization of the variable x_e in equation (2.26)
Transposing:

$\mathbf{u'(x)^T K(x)} = -\mathbf{u(x)^T K'(x)}.$

Replacing this equality in equation (5.5)

$$\frac{\partial C(x)}{\partial x_e} = -\mathbf{u^T(x) K'(x) u(x)}. \tag{5.6}$$

This way, we only require to calculate the derivative of the stiffness matrix respect to each design variable x_e, given by:

$$\frac{\partial C(x)}{\partial x_e} = -u(x_e)^T k'(x_e)u(x_e). \tag{5.7}$$

Based on equation (2.26) and considering the *penalization power* p, the derivative of the elemental stiffness matrix is:

$$k'(x_e) = p x_e^{p-1} k_e \tag{5.8}$$

Replacing equation (5.8) in (5.7)

$$\frac{\partial C(x)}{\partial x_e} = -p(x_e)^{p-1} u^T(x_e) k_e u(x_e) \tag{5.9}$$

Continuing with the derivative of the second term in equation (5.3):

$$\frac{\partial G(x)}{\partial x_e} = \frac{V_e}{V_0} \tag{5.10}$$

due to $G(x) = \frac{\sum_{e=1}^{N} x_e V_e}{V_0}$ and

$$\lambda = Lagrangian\ multiplier\ for\ volume\ constraint \tag{5.11}$$

A local minimum is found when the first optimality condition is fulfilled:

$$\frac{\partial J(x)}{\partial x_e} = \frac{\partial C(x)}{\partial x_e} + \lambda \frac{\partial G(x)}{\partial x_e} = 0 \tag{5.12}$$

A fixed point method is applied to find the parameters x that fulfill $\frac{\partial J(x)}{\partial x_e} = 0$. Substituting the derivatives in equation (5.9) and equation (5.10):

$$p x_e^{p-1} u_e^T k_e u_e = \lambda \frac{V_e}{V_0} \tag{5.13}$$

$$p x_e^p x_e^{-1} u_e^T k_e u_e = \lambda \frac{V_e}{V_0} \tag{5.14}$$

$$x_e = \frac{p[x_e^p] u_e^T k_e u_e}{\lambda \frac{V_e}{V_0}} = \frac{p[x_e^{p-1} x_e] u_e^T k_e u_e}{\lambda \frac{V_e}{V_0}} \tag{5.15}$$

The update function is :

$$x_e = x_e \frac{p x_e^{p-1} u_e^T k_e u_e}{\lambda \frac{V_e}{V_0}} = x_e \frac{-\frac{\partial C(x)}{\partial x_e}}{\lambda \frac{\partial G(x)}{\partial x_e}} = x_e B_e \tag{5.16}$$

Some observations about the update function in equation (5.16):

- The B_e numerator $(-\frac{\partial C(x)}{\partial x_e} = p x_e^{p-1} u_e^T k_e u_e)$ is always a positive scalar, due to x_e is a value in the range $(x_{min}, 1]$, $p = 3$ and for any $u_e \neq 0$, $u_e^T k_e u_e > 0$.
- The B_e denominator $(\lambda \frac{\partial G(x)}{\partial x_e})$ is also always positive due to λ is a positive Lagrange multiplier, and $\frac{\partial G(x)}{\partial x_e}$ is a quotient between volumes.

The update equation (5.16) updates positive values on design variables. Nevertheless, it does not ensure that design variables fulfill the constraint in equation (3.4).

5.1.2 Heuristic Approach

Parameter η for regularizing the fixed point method

The next adjustment is applied to equation (5.16):

$$x_e^{t+1} = x_e^t \left(\frac{-\frac{\partial C(x)}{\partial x_e}}{\lambda \frac{\partial G(x)}{\partial x_e}} \right)^{\eta} = x_e^t B_e^{\eta} \tag{5.17}$$

Factorizing equations (5.16) and (5.17) in the fixed point:

$$x_e - x_e B_e = x_e(1 - B_e) = 0 \tag{5.18}$$

$$x_e - x_e B_e^{\eta} = x_e(1 - B_e^{\eta}) = 0 \tag{5.19}$$

The fixed points are the same for both equations, so the solution of the problem is not affected by the use of the parameter η.

The use of η is related to improving the performance of the method. Parameter η "smooths" the value of B_e. Note that, in equation (5.17), the update of a design variable, is formed by its current value x_e multiplied by a "factor" B_e^η. A value of B_e^η around 1 produces smooth changes in the update of x_e. The behavior of η on B_e is shown in graphic in figure 5.2, it shows how η approximates the value of B_e^η to 1. For this work $\eta = \frac{1}{2}$

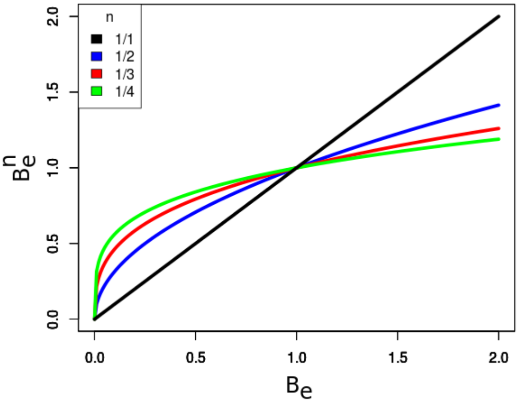


Figure 5.2: Behaviour of parameter η over B_e

Update range

A parameter m is used to define an update range $[x_e - m, x_e + m]$ for each design variable x_e. If equation (5.17) generates a value out of this range, the variable is updated until the limit of the range that has been overcome or, in which case, until the variable limit $[x_{min}, 1]$ that has been overcome. This heuristic is intended for:

- Avoiding abrupt changes through iterations.

- Validating that the design variables do not overcome the range $(x_{min}, 1]$ as is established in the constraint in equation (3.4).

Considering the update range, the update step is carried out as follows:

$$\mathbf{x_e^{t+1}} = \tag{5.20}$$

$$\begin{cases} \mathbf{max(x_{min}, x_e^t - m)} & \text{if } x_e B_e^\eta \leq max(x_{min}, x_e^t - m) \\ \mathbf{x_e^t B_e^\eta} & \text{if } max(x_{min}, x_e^t - m) < x_e^t B_e^\eta < min(1, x_e^t + m) \\ \mathbf{min(1, x_e^t + m)} & \text{if } min(1, x_e^t + m) \leq x_e^t B_e^\eta \end{cases}$$

Calculating λ

This parameter is used in equation (5.17), it is calculated by a bisection method, searching a value that fulfills the volume constraint (equation (3.3)) as it is stated in algorithm 4:

Algorithm 4 Calculating λ

```
A search range is established: $[\lambda_{inf} : \lambda_{sup}]$
while $\lambda_{sup} - \lambda_{inf} > \epsilon$ do
    $\lambda_{tmp} = \frac{\lambda_{inf}+\lambda_{sup}}{2}$
    Temporal design variables($x_{tmp}$) are updated using $\lambda_{tmp}$ and the volume for this temporal structure is calculated ($V(x_{tmp})$).
    If $V(x_{tmp}) = V_r$ : break
    If $V(x_{tmp}) > V_r$ : $\lambda_{inf} = \lambda_{tmp}$
    If $V(x_{tmp}) < V_r$ : $\lambda_{sup} = \lambda_{tmp}$
end while
$\lambda = \lambda_{tmp}$
```

Filtering

A filtering process is applied to the values of the derivatives $\frac{\partial C(x)}{\partial x_e}$ calculated by (5.9). The filtering modifies the current values of the derivatives, considering the values of the derivatives on its neighborhood. Neighborhoods are established using radial basis functions. The filtered value on an element is calculated as follows:

$$\frac{\partial \hat{C}(x)}{\partial x_e} = \frac{1}{x_e \sum_{f=1}^{N} H_f} \sum_{f=1}^{N} H_f x_f \frac{\partial C(x)}{\partial x_f} \tag{5.21}$$

Where:

$$H_f = r_{min} - dist(e, f) \tag{5.22}$$

Where r_{min} is a *ratio* that delimits a region around each element e, all the elements whose centroid lays inside this region are considered neighbours. $dist(e, f)$ is the Euclidean distance between centroids of elements e and f.

The filtering avoids the *checkerboard* phenomenon on the structure, which occurs when derivatives are calculated independently.

5.1.3 General structure for SIMP

Algorithm 5 shows the general structure for the SIMP:

Algorithm 5 SIMP algorithm structure

```
Entries : $p, \eta, r, m, x, t = 1$
Initialize $x^t$
$(k, f)$=Assembling(Mesh,Service Conditions)
while Convergence not reached do
    $u$=FEM$(k, f, x^t)$
    $\frac{\partial C(x)}{\partial x_e}$ = ComplianceDerivative$(p, u, k, x^t)$[Equation 5.9]
    $\frac{\partial \hat{C}(x)}{\partial x_e}$ = Filtering$(\frac{\partial C(x)}{\partial x_e}, r)$[Equation 5.21]
    $\frac{\partial G(x)}{\partial x_e}$ = VolumeDerivative$(x^t)$[Equation 5.10]
    $\lambda$ = Calculating$\lambda$[Algorithm 4]
    $B_e$ = Calculating$B_e(\lambda, \frac{\partial \hat{C}(x)}{\partial x_e}, \frac{\partial G(x)}{\partial x_e})$[Equation 5.17]
    $x^{t+1}$ = Update$(x^t, B_e^\eta, m)$[Equation 5.20]
    $t = t + 1$
end while
```

Despite SIMP has a mathematical approach that ensures optimality, the filtering could improve the aesthetic of the results, and could be convenient for the minimization problem. In other words, sometimes the filtering generates improvements of the aesthetic of the structure, but not for the minimization of the objective function. An example of this phenomenon is shown in Results chapter as an extra experiment in OLM test.

5.2 SIMP with stress constraint: SIMP-SC

SIMP method does not consider information about elemental stresses in the structure because its objective is to build an optimal structure for the compliance problem, without considering if the structure is over or under-stressed.

Stress values are directly calculated because they are derived from displacements. They can be used to get the efficiency of the structure, by comparing the elemental stresses σ_e with the yield stress(σ_{YS}). The closest is σ_e to σ_{YS}(considering it is not over stressed, it means that $\sigma_e < \sigma_{YS}$) the most efficient is such element. If an elemental stress is higher than σ_{YS}, that element (and also the structure) is over stressed and its behavior changes from elastic to plastic. This is not convenient for real structures, due to it indicates that they are prone to fail.

Stress values are used to calculate the security factor(SF) of the structure. It is given by:

$$SF = \frac{\sigma_{max}}{\sigma_{YS}} \tag{5.23}$$

Where σ_{max} is the maximal elemental stress in the structure. If $SF < 1$, the structure is not over-stressed.

A contribution of this work is adding the stress constraint to the SIMP method: SIMP-SC, in order to get efficient structures through a bisection method. This implies to find an adequate volume that generates an structure with an SF just below to 1. Algorithm 6 shows the structure of SIMP-SC.

Algorithm 6 SIMP-SC algorithm structure

```
A search range is established: $[Vr_{inf}, Vr_{sup}] = [0, 1]$
**while** $Vr_{sup} - Vr_{inf} > \epsilon$ **do**
    $V_r = \frac{Vr_{inf}+Vr_{sup}}{2}$
    Execute SIMP using $V_r$ as volume constraint until reaching convergence and calculate the $SF$ of the final structure.
    If $SF < 1 : Vr_{sup} = V_r$
    If $SF \geq 1 : Vr_{inf} = V_r$
**end while**
```

The main issue with SIMP-SC is the convergence criterion for the SIMP (taken from [118]), which is based on the convergence criterion of the fixed point method. This criterion shows an inconvenience for using the same ϵ stopping criterion to reach the convergence for different tests, so, the ϵ value must be adapted for each one. Another issue is that sometimes, the convergence is not reached even when the solution(visual and numerical) does not change significantly in consecutive iterations, this provokes an stagnation. In the SIMP-SC method it is important that SIMP converges in a small number of iterations using the same convergence parameters, in order to adjust the volume constraint V_r, so it is necessary to set a convergence criterion less dependent on the particular problem.

In the following section, we carry out an analysis of the convergence criterion (based on fixed point) for the SIMP, and two proposed convergence criteria which circumvent the mentioned inconveniences. The analysis includes comparatives between all criteria, based on iterations, objective function values, and visual differences of the structures for two tests.

5.2.1 Convergence criteria

The convergence criterion for the SIMP is based on the fixed point method, it is considered that the fixed point method converges if the values of x_e are enough similar in two consecutive iterations. Thus, the algorithm stops if the following criterion is accomplished:

$$|x^{t+1} - x^{t}| < e \qquad (5.24)$$

Equation (5.24) represents that the norm of the difference of the design variables (x^t and x^{t+1}) of two consecutive iterations, must be lower than a value ϵ close to zero. Consider that the larger the number of elements, the larger the norm of the error is, due to the norm is calculated by the sum of the differences of every element, this issue suggest that the ϵ value must be adjusted to every test depending on the mesh size, which is not convenient for a robust convergence metric. Another issue is that the fixed point method could guide the updating process contrary to the heuristic, this could generates an oscillatory phenomenon on the updating every iteration, which affects the value of the norm of the convergence criterion. The fixed point convergence criterion considers only two consecutive iterations, it does not capture the oscillatory phenomenon and complicates the convergence.

To avoid these issues, we propose the use of the *security factor* of the structures of two consecutive iterations, due to it is independent of the number of elements and it could represent the similarity between two consecutive structures. The convergence is reached if the following criterion is accomplished:

$$|SF^{t+1} - SF^t| < e \tag{5.25}$$

Where SF^t and SF^{t+1} are the security factor of two consecutive iterations. An small change in the security factor of two consecutive structures, usually, represents a high similarity on the shape of those structures. No matter, this criterion could generate an early convergence. In order to avoid this inconvenience, we propose adding another criterion based on the variance of the historic objective function values. This convergence is reached if the following criterion is accomplished:

$$\frac{1}{v}\sum_{i=0}^{v-1}(c(x)^{t-i} - c(\hat{x})_v)^2 < e \tag{5.26}$$

Where v is the number of the last consecutive iterations, $c(x)^{t-i}$ represent every objective function value in those iterations and $c(\hat{x})_v$ is the mean of the objective function values in those iterations. Equation 5.26, calculates the variance of the objective function values of the last v consecutive iterations, then it evaluates if that variance is lower than a (close to zero) value *epsilon*.

Every criterion is tested for two different problems to analyze their performance. Notation for every criterion is 1) Convergence based on Fixed Point: CFP(this is the original criterion used for SIMP), 2) Convergence based on Security Factor: CSF and 3) Convergence based on Security factor and historic Variance: CSV.

Convergence tests

An SCLB and a LLT tests were executed using the three convergence criteria. The tests were stopped until all criterion converged.

Figure 5.3 shows the compliance evolution for the SCLB test, starting in iteration 10 and finalizing in iteration 183 in logarithmic scale. Notice that there is an stagnation in the compliance values which starts around the iteration 20. The mean and variance of compliance from iteration 20 to 183 is $2.8295E1$ and $1.0055E-4$ respectively, this suggest an stagnation in all this range.

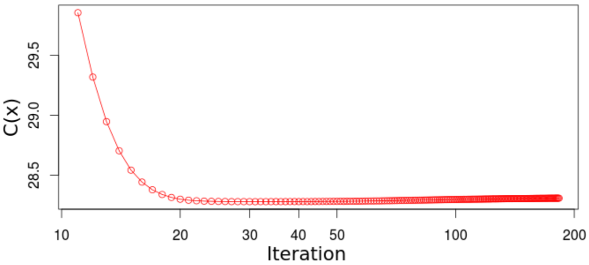
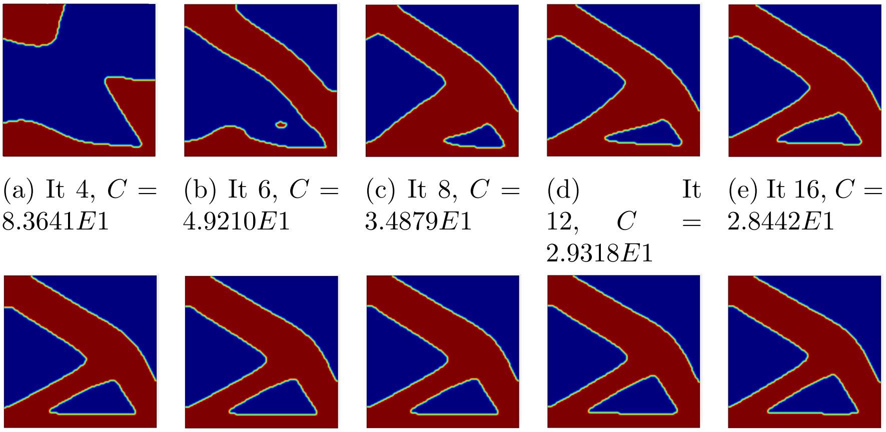


Figure 5.3: SCLB compliance evolution

Figure 5.4 shows the visual evolve of the tests results until the iteration 183. Every sub-figure shows the iteration and compliance(C) reached. Note that visual solution does not change significantly between iterations 20 to 183.



(a) It 4, $C = 8.3641E1$ (b) It 6, $C = 4.9210E1$ (c) It 8, $C = 3.4879E1$ (d) It 12, $C = 2.9318E1$ (e) It 16, $C = 2.8442E1$

(f) It 20, $C = 2.8299E1$ (g) It 30, $C = 2.8279E1$ (h) It 50, $C = 2.8281E1$ (i) It 80, $C = 2.8291E1$ (j) It 183, $C = 2.8307E1$

Figure 5.4: Evolution of Short Cantilever test

Figure 5.6 shows the compliance evolution for the LLT test, starting in iteration 10 and finalizing in iteration 590 in logarithmic scale. Notice that there is an stagnation in the compliance values which starts around the iteration 20. The mean and variance of compliance from iteration 20 to 590 is $2.1044E1$ and $2.6911E-4$ respectively, this suggest an stagnation in all this range.

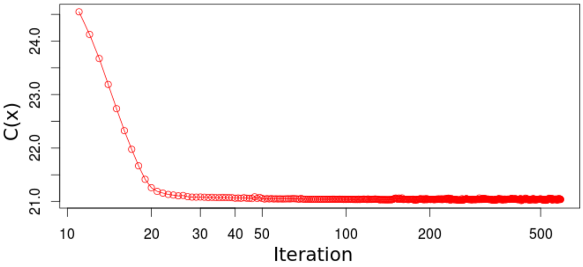
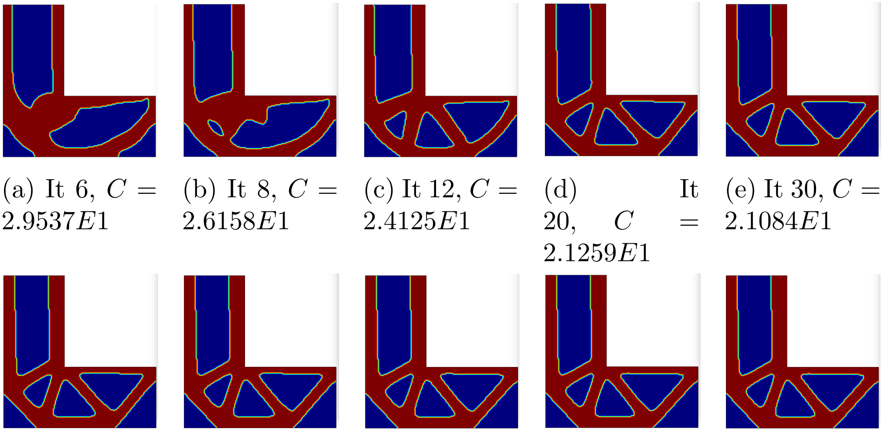


Figure 5.5: LLT compliance evolution

Figure 5.6 shows the visual evolve of the tests results until the iteration 590. Every sub-figure shows the iteration and compliance(C) reached. Note that visual solution does not change significantly between iterations 20 to 590.



(a) It 6, $C = 2.9537E1$

(b) It 8, $C = 2.6158E1$

(c) It 12, $C = 2.4125E1$

(d) It 20, $C = 2.1259E1$

(e) It 30, $C = 2.1084E1$

(f) It 40, $C = 2.1062E1$

(g) It 80, $C = 2.1043E1$

(h) It 160, $C = 2.1040E1$

(i) It 300, $C = 2.1037E1$

(j) It 590, $C = 2.1038E1$

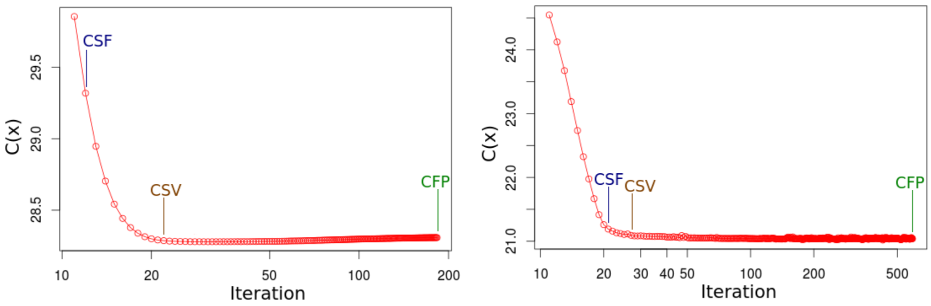
Figure 5.6: Evolution of LShape test

Table 5.1 shows the following data for both tests for each criterion: the e value used for every criterion: e_{CFP},e_{CSF} and e_{CSV}(for CSV criterion, the e value is the same for both criteria used: e_{CSF} and e_{CSV}), the convergence iteration(CI), the percent of iterations reached(%), the compliance reached($C(x)$), mean(M) and variance(V) of compliance values from the iteration 20 to the convergence iteration. This information provides a comparative for the performance of each test.

Figure 5.7 shows the convergence iteration for every criterion for SCLB test in sub-figure (a) and for LLT test in sub-figure (b).

Test	**Crit**	e	**CI**	**%**	$C(x)$	**M**	**V**
SCLB	CFP	$1E^{-2}$	183	100%	$2.830E^{1}$	$2.829E^{1}$	$1.005E^{-4}$
	CSF	$1E^{-3}$	11	6.01%	$2.985E^{1}$	early	early
	CSV	$1E^{-3}$	22	12.02%	$2.828E^{1}$	$2.829E^{1}$	$4.288E^{-5}$
LLT	CFP	$1E^{-2}$	590	100%	$2.103E^{1}$	$2.104E^{1}$	$2.691E^{-4}$
	CSF	$1E^{-3}$	21	3.55%	$2.119E^{1}$	$2.122E^{1}$	$2.409E^{-3}$
	CSV	$1E^{-3}$	27	4.57%	$2.109E^{1}$	$2.114E^{1}$	$3.047E^{-3}$

Table 5.1: Results of convergence tests for CFP, CSF, and CSV



(a) Convergence criteria for SCLB

(b) Convergence criteria for LLT

Figure 5.7: Convergences for criteria on both tests

Results in tests show: a) CFP is an unstable criterion, it requires a high number of iterations for convergence, even when the method is stagnated since a lot of iterations, also, compliance reached is not considerably different from the other two criteria. There is a large difference in the number of iterations to reach convergence, in order to reduce this difference, we must adjust a priori the parameter ϵ, which is not possible. This shows a dependence on the problem size. b) CSF is an early convergence criterion, because it stops the algorithm before the compliance and shape of the structure gets its best value and shape. c) CSV criterion has a well performance according to the opportune convergence considering the stagnation range observed, this avoids the issue with the CFP criterion for SIMP-SC, also for the considerable reduction of the iterations used and the similarity of the compliance reached even for the lower number of iterations. CSV criterion works using the same value of the parameter ϵ, even for different tests, which shows an independence on the problem size.

The integration of the CSV criterion applied to SIMP is named: SIMP with Stress and Variance convergence Criterion(SIMP-SVC). This version is tested with all benchmark problems, in order to get a performance comparative with the original SIMP. A high saving of the computational cost is expected, due to the iteration reduction, without diminishing the quality of the results.

Results obtained for every method implementation is shown at Results chapter. All 13 benchmark tests were executed using SIMP, SIMP-SVC and SIMP-SC described in this chapter.

Chapter 6

Evolutionary algorithms for topology optimization problems

Evolutionary algorithms are populational-metaheuristic methods used to solve a variety of global optimization problems with one or more objective functions. On one hand, topology optimization algorithms are computationally expensive due to, usually, this kind of algorithms map a finite element property to an optimization variable, such as the SIMP case, in consequence, there are as many variables as elements in the finite element mesh. On the other hand, evolutionary algorithms are not well suited to handle more than dozens of variables. Another issue is the computational cost of evaluating the objective function, in which the most expensive step is to solve a system of linear equations. In order to avoid these inconveniences, we propose to modify the individual representation, by means of a technique based on control points, for reducing the dimension of the problem, and to use shared-memory parallel computing to distribute the computational cost of the evaluation of structures.

6.1 Representation based on control points

The proposed individual representation is used to reduce the dimension of the problem. It turns the problem from binary to real, allowing the use of mono or multi-objective evolutionary algorithms with real-value representation.

Figure 6.1 shows the design domain for an OLM test, every red point inside represents an element centroid. In the example, the mesh is formed by 10,000 elements. In a conventional topology optimization problem, every red point would represent a design variable.

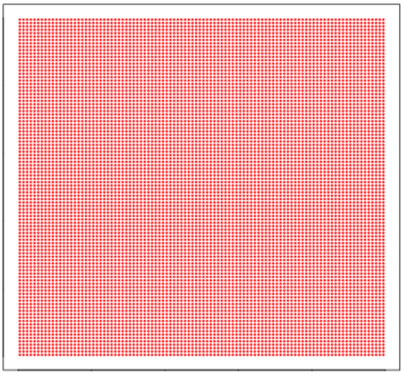


Figure 6.1: Meshed design domain

A group of points are randomly distributed over the design domain and approximately equidistant among them, they are called *control points*. Figure 6.2 shows an example of 100 control points over the domain of figure 6.1. Note that the number of control points is much lower than the number of element centroids.

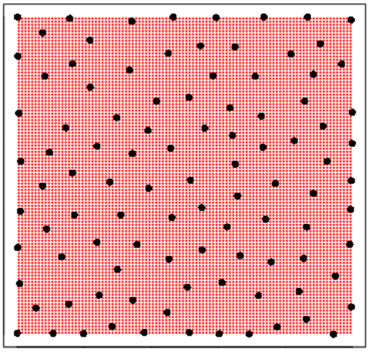


Figure 6.2: Control points example

Every control point can take a random uniform value in the range $[0, 1]$. In the *z-axis*, these values can be seen as heights H, as shown in figure 6.3, wherein the xy plane, the black and red points represent the control points and element centroids over the design domain, respectively, and the blue spheres represent the heights H assigned to control points.

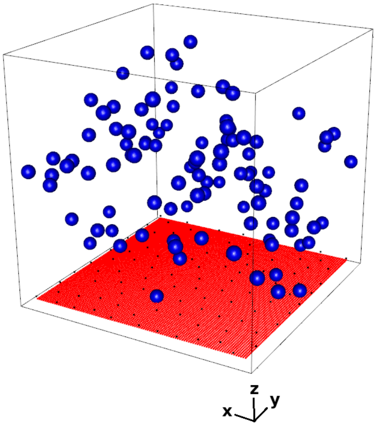


Figure 6.3: Control points heights H

The element heights h can be calculated through a interpolation given by:

$$h_e = \sum_{j=1}^{m} w_{ej} H_j \tag{6.1}$$

Where h_e is the height of the element e, m is the number of control points in its neighborhood, w_{ej} is the weight of the control point j in the neighborhood of the element e, and H_j is the height of the control point j. The heights H of control points are always known, and neighborhoods and weights w are calculated using a radial basis function as follows, the neighborhood of an element e is formed by the control points inside a circle with radius r, and center at its centroid, as shown in figure 6.4, where the four circles represent the neighborhoods for the four example elements, which are represented by the four cross marks. Solid points represent the control points in each neighborhood. Note that some control points belong to many neighborhoods.

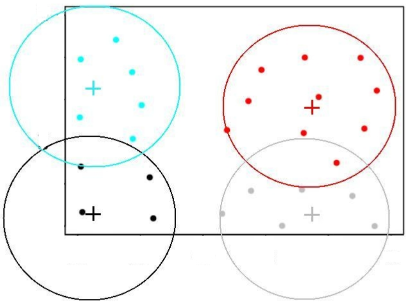


Figure 6.4: Elemental neighbourhood example

The weights w for control points in a neighborhood of an element e are calculated by the following radial basis function:

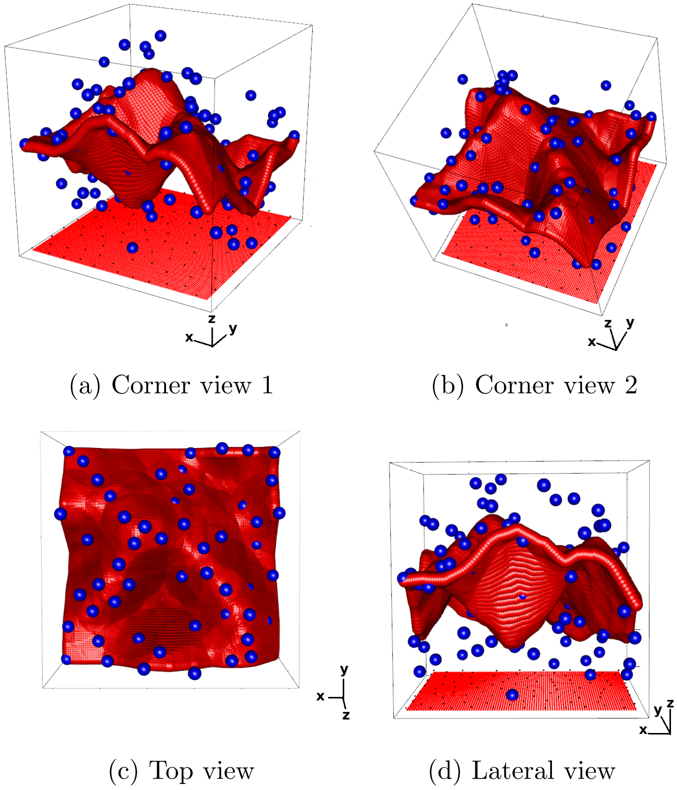
$$w_{ej} = \begin{cases} r - dist(control\ point_j, e) & \text{if } r - dist(control\ point_j, e) \geq 0 \\ 0 & otherwise \end{cases} \tag{6.2}$$

Where r is the radius parameter and $dist(control\ point_j, e)$ is the Euclidean distance between control point j and the centroid of the element e. Then weights w_{ej} are normalized as follows:

$$w_{ej} = \frac{w_{ej}}{Z} \tag{6.3}$$

Where Z is the magnitude of the vector formed with the all the distances $r - dist(control\ point_j, e) \geq 0$.

Figure 6.5 shows, from different views, the interpolation, using equation 6.1. Red spheres are the elemental heights forming a surface.



(a) Corner view 1

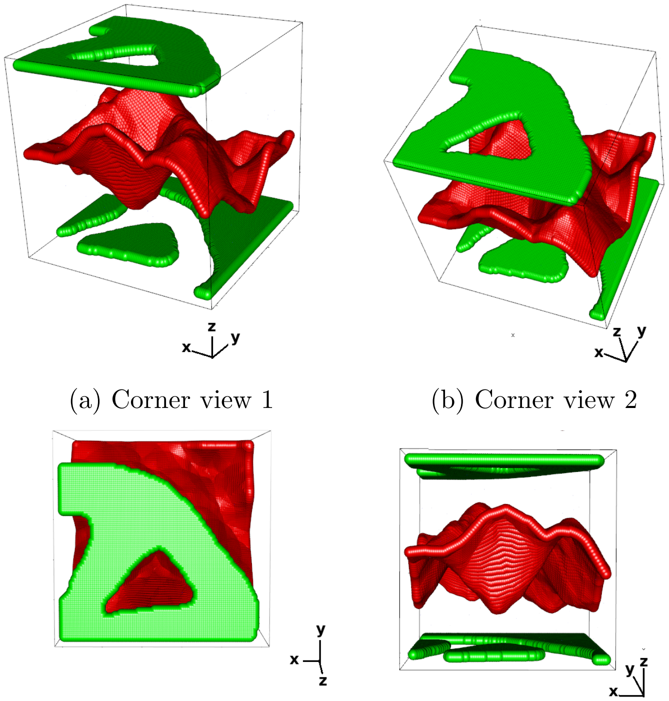
(b) Corner view 2

(c) Top view

(d) Lateral view

Figure 6.5: Interpolation of elemental heights from control point heights

The next step is thresholding the elemental heights h, in order to assign a value to each design variable x_e. The threshold value is $t = 0.5$. This way, $x_e = x_{min}$ if $h_e < t$, or $x_e = 1$ otherwise. Figure 6.6 shows, from different views, the elemental heights h after the thresholding, the green surfaces, above and below the red surface, are the thresholding result, $h_e = x_e = 1$ at the top surface and $h_e = x_e = x_{min}$ at the bottom surface.



(a) Corner view 1

(b) Corner view 2

(c) Top view

(d) Lateral view

Figure 6.6: Thresholding of elemental heights to assign values to design variables

Once design variables values are assigned, the elastic problem is solved by means of the finite element method, in order to calculate elemental displacements, stresses and the objective function or functions.

The number of design variables is considerably reduced from dozens of thousands to hundreds, in consequence, and individual represents the heights of the control points. Hereafter, the described process for generating a structure from control points heights is called: *structure generation.*

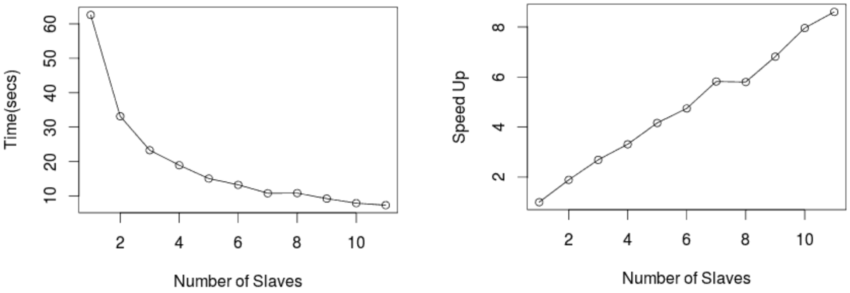
Control point locations and neighborhoods are established only once during the algorithm execution. An individual is coded as an array of heights of control points. This representation allows the use of real-coded evolutionary algorithms. For this work, two mono-objective evolutionary algorithms are implemented for the compliance optimization, and a multi-objective is implemented for the optimization of both conflicting functions: compliance and volume.

6.2 Parallel evaluation

As mentioned in chapter Theoretical framework, shared memory parallelization is used indirectly for linear algebra operations (mainly the solution of systems of linear equations) due to it is already implemented in the FEMT library.

The evaluation step is parallelized using a master-slave scheme with distributed memory. Master sends a section of the population to each slave, which receives it and carry out the *structure generation.* Then, slaves return the calculated fitnesses.

The scalability and performance of the parallelization is tested according to the following conditions: using a population size of 750 with 75 control points, in a mesh with 900 elements, we vary the number of slave nodes, and compute the time and the speed-up(the quotient between the serial and the parallel time). They are shown in figure 6.7. As it is shown, on one hand, the speed-up shows an impressive reduction of the computational time, about eight times faster than the original when using ten slaves, on the other hand, the time reduction shows that the speed-up is, actually, non-linear, and there is an overhead which increases with the number of slaves. A logical explanation is that the master increases the overhead, because of it must manage more slaves, So, it is possible that a different topology of the communication network for the paralellization would reduce this overhead, nevertheless, the current scheme is sufficient for our needs. The execution is carried out using from 1 node to 11 different nodes.



(a) Execution time for 1-11 slaves

(b) SpeedUp for 1-11 slaves

Figure 6.7: Calculating Times and Speed Up for test

6.3 Mono-objective algorithms: solving compliance problems

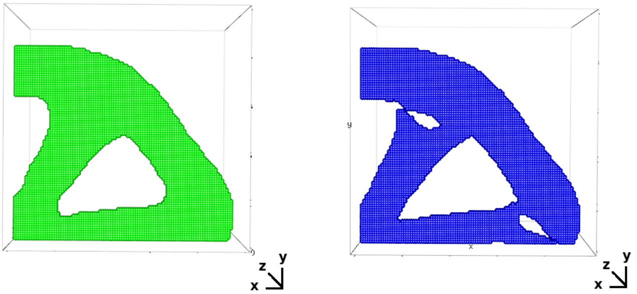
Two mono-objective evolutionary algorithms are implemented: the continuous univariate marginal distribution algorithm (UMDA_c) and a genetic algorithm(GA). A general description of each one is given in the theoretical framework in chapter 2. In this section, the details of these methods oriented to the solution of the compliance problem are described.

6.3.1 Volume constraint

In order to solve the compliance problem in equation 3.1, we need to fullfil the constraint of delivering a fraction of the initial volume. Hence, for this purpose, we apply a volume fitting procedure based on the Evolutionary Structural Optimization(ESO) method [160], which consists in iteratively eliminating a portion of low stressed elements until reaching the desired volume. Note that this procedure involves modifying the x_e values of low stressed elements and, therefore, modifying their heights h_e. Once the structure fulfills the volume constraint, we modify the heights of the control points H in the respective individual, through the solution of a local optimization problem, with the purpose of mapping back the elemental volume modifications.

- The *structure generation* procedure generates structures with a high diversity of volumes, we are only interested in those with a volume over and close to the volume constraint V_r. This group of structures is called the *interest group*, and it is obtained using a parameter called *valid volume*: vv. The interest group is formed by the structures with a volume in the range $[V_r, V_r + vv]$. Figure 6.8a) shows an example of one structure given by the *structure generation* process, its volume is 0.5 and the volume constraint is $V_r = 0.45$.

- For each structure in the *interest group*, elemental Von Mises stresses are calculated. Then, the less stressed elements are removed until reaching the volume constraint V_r. Note that the elements heights h have been removed, but the control points heights H remain unchanged. Figure 6.8 b) shows the structure after removing elements.



(a) Original structure

(b) Structure after removing elements

Figure 6.8: Resulting structure after removing elements to fulfill the volume constraint

- In order to map back the modifications to the control point heights H, after removing elements, we use a gradient descend algorithm. Thus, we propose an objective function in equation 6.4.

$$R = \frac{1}{2Z} \sum_{e=1}^{n} (h_e - \hat{h_e})^2 \tag{6.4}$$

Where n is the number of elements, h_e are the elemental heights in the original structure, $\hat{h_e}$ are the elemental heights after the removing-elements procedure, and Z is a normalization variable. Replacing equation 6.1 in 6.4:

$$R = \frac{1}{2Z} \sum_{e=1}^{n} (\sum_{j=1}^{m} w_{ej} H_j - \hat{h_e})^2 \tag{6.5}$$

Then, we must minimize this function. For this purpose, we compute its gradient respect to every height H_j:

$$\frac{\partial R}{\partial H_j} = \frac{1}{Z} \sum_{e=1}^{l} (\sum_{j=1}^{m} w_{ej} H_j - \hat{h_e}) w_{ej} = \sum_{e=1}^{l} (h_e - \hat{h_e}) \frac{w_{ej}}{Z} \tag{6.6}$$

Where l is the number of elements that contains the control point j into its neighborhood. Weights w_{ej} are calculated with equation 6.2 and Z normalizes them respect to all elements in the neighborhood of the control point j, in order to facilitate the search of an step size for the gradient.

Figure 6.9 shows, with red circles, the regions where the most of the elements have been removed. Hence, the gradients of control points around those regions must be the highest.

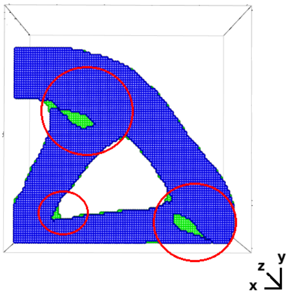
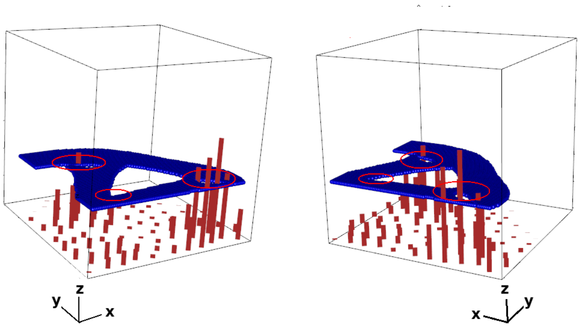


Figure 6.9: Marking zones with element reduction.

Figure 6.10 shows, with red strips the magnitude of every control point gradient, and with red circles, the areas where the gradients are higher. A high gradient represents a high modification of the control point. Note that the elements which depend on a control point with a high gradient are the most reduced, therefore, the most affected elements are those with the lowest height value.



(a) Control point gradients view 1. (b) Control point gradients view 2.

Figure 6.10: Control points gradients.

- To update the control points that generate a structure that fulfills the volume constraint, a steepest descend algorithm is followed using the calculated gradient, in agreement to the updating formula:

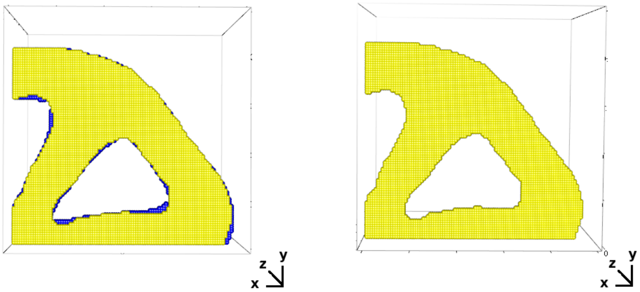
$H_j = H_j - \alpha\frac{\partial R}{\partial H_j}$, where α is a step size. To calculate α, a bisection method is used as follows: we tests an initial α, then it is reduced or increased until the volume constraint is fulfilled. Algorithm 7 shows the calculation of α and the update of the control point heights H.

Algorithm 7 Calculate of α and update of control points heights H

```
1:  $\alpha = 100$
2:  $update = \frac{\alpha}{2}$
3:  $H_{temp} = H - \alpha\frac{\partial R}{\partial H}$
4:  $x_{temp} = generate\ structure(H_{temp})$
5:  if $v_r < V(x_{temp}) < v_r + e$ then
6:      $H = H_{temp}$
7:      exit
8:  end if
9:  if $V(x_{temp}) > v_r$ then
10:     $\alpha = \alpha + update$
11: else
12:     $\alpha = \alpha - update$
13: end if
14: $update = \frac{update}{2}$
15: go to 3 :
```

Where $V(x_{temp})$ is the volume of the temporal structure x_{temp}.

- Once control points are updated, the new structure is generated, and displacements and compliance are calculated ensuring the structure fulfills the volume constraint. Figure 6.11a) shows a comparative between the structure with removed elements by stress(at the bottom with blue) and the new structure generated with the upgraded control points(at front with yellow). Note that the elements around to regions of control points with the highest gradients have been removed. Figure 6.11b) shows the new structure which fulfills the volume constraint V_r.



(a) Structure with removed elements(blue) vs new structure(yellow)

(b) New structure

Figure 6.11: Generating structure with updated control points

6.3.2 Mono-objective implementations details

Mono-objective implementations are taken from algorithm 1 for GA case and algorithms 2 and 3 for the UMDA. A general description of each step is given in the theoretical framework chapter. The following are the details, of each step in the implementations, oriented to the solution of compliance problems:

- Generate population: the initial population is generated using a uniform distribution with a range $[x_{min}, 1]$ for each control point value on every individual.

- Stopping criterion: it is established by a maximal number of iterations.

- Selection: it consist of two sub-steps:

  – Sub-step 1: selection by *valid volume(vv)*. All the individuals that generate structures with a volume in the range $[V_f, V_f + vv]$ are selected.

  - Sub-step 2: selection by truncation. The currently selected individuals are sorted upwardly by compliance, and the first half is selected.

- Mutation and crossover step:
  - For UMDA, the crossover and mutation are replaced by the process described in algorithm 3. Children are generated by estimated distribution.
  - For GA, the crossover and mutation are carried out with the operators *simulated binary crossover*(using parameters $crossover\ probability =$ 0.9 and $\eta_c = 10$ from [48]) and *polynomial mutation* (using parameters $mutation\ probability = \frac{1}{number\ of\ control\ points}$ and $\eta_m = 20$ from in [47]). The implemented codes of these methods and parameters values are taken from [3].

- Replacement: it is carried out by the following process:
  - The selected individuals are preserved for the next generation.
  - The remaining individuals are generated using the respective method, generating n new individuals, where $n = population\ size - selected\ size$. For each algorithm the respective method to generate individuals is as follows:
    - UMDA: using the estimated distribution to generate new n individuals.
    - GA: we randomly take two individuals from the selected set and apply the crossover operator to generate two children, the operator is repeated until generating n individuals. If n is an odd number, the last children is generated as a replica of one individual randomly chosen.

6.4 Multi-objective algorithms: solving compliance and volume problems

The NSGA-II is described in the section Stochastic optimization. It is used to optimize both, compliance and volume objective functions.

Compliance and volume are opposed objective functions, the minimal compliance structure is full of material and the minimal volume structure is empty of material.

For this case, the constraints of both optimization models are not considered during the execution, but they are taken into account as a post-process explained below with an example. The solution the NSGA-II is a Pareto front as shown in figure 6.12. Among these solutions, there are optimal compliance(full material) and optimal volume(empty material) approximations, but additionally, there are other solutions with an intermediate compromise between compliance and volume which are interest because they fullfil the model constraints (equation (3.3) and (3.7)). For example, 1) for compliance problems, we can select only structures around a specific volume, for example $V(x) = 0.41$ as shown with a red box in figure 6.12 or 2) for volume problems, we can select only structures with a specific security factor, for example $SF(x) = 0.98$ as shown with a blue box in figure 6.12. This way, we can deliver solutions for both problems.

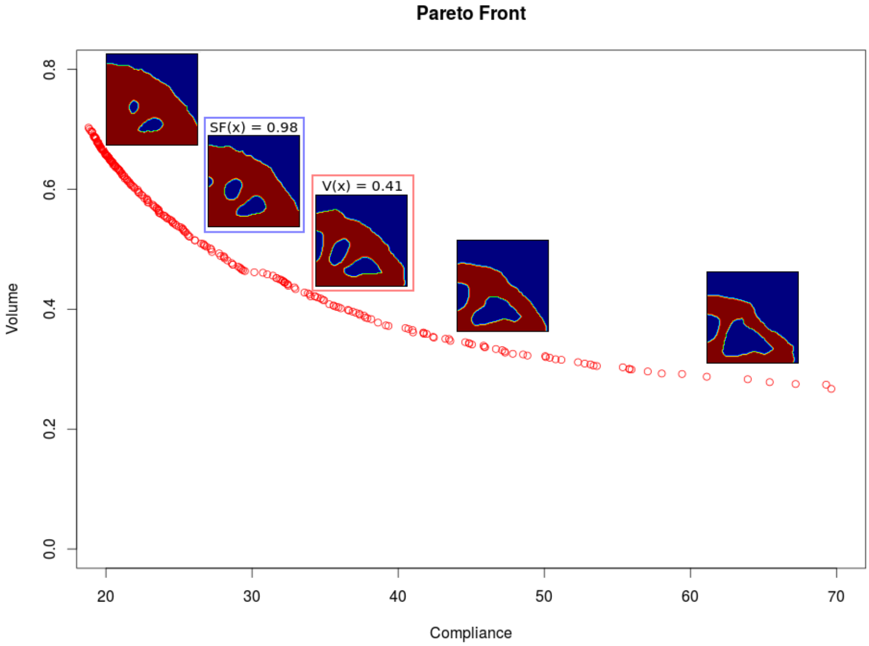


Figure 6.12: Example of pareto front in topology optimization.

Thus, in the results chapter, NSGA-II solutions for compliance problems are reported as NSGA-VC(volume constrained) and NSGA-II solutions for volume problems are reported as NSGA-SC(stress constrained).

6.4.1 Multi-objective implementations details

NSGA-II implementation is based on algorithm 1. A general description of each step is given in the theoretical framework chapter. The following are the details of each step in the implementation of NSGA-II oriented to the solution of compliance and volume problems:

- Generate population: the initial population is generated using a uniform distribution with a range $[x_{min}, 1]$ for each dimension of each individual.

- Stopping criterion: it is established by a maximal number of iterations.

- Selection step: this process is by a binary tournament, two individuals are uniformly random taken and the best is selected according to their Pareto rank and crowding distance. This operation is repeated n times, where n is the population size.

- Mutation and crossover step: they are carried out through the operators: *simulated binary crossover*(using parameters $crossover\ probability =$ 0.9 and $\eta_c = 10$ from [48]) and *polynomial mutation* (using parameters $mutation\ probability = \frac{1}{number\ of\ control\ points}$ and $\eta_m = 20$ from in [47]). The implemented codes of these methods and parameters values are taken from [3]. These steps generate n children, where n is the population size.

- Replacement step: it is carried out adding the children to the original population generating a population of size $2n$, where n is the original population size. The population for the next iteration is generated according to the following criteria:

  - Adding individuals from the first ranks until reach n individuals.
  - If the next front to add, generates a population bigger than n, only the necessary individuals are added according to their crowding distances.

The results obtained by each algorithm are shown at the Results chapter. The 13 benchmark tests are reported with the GA, UMDA and NSGA(VC and SC) methods.

Chapter 7

Results

In this chapter, results for the SIMP, SIMP-SVC, SIMP-SC, GA, UMDA and NSGA are presented. Results are grouped by 1) volume minimization results from: SIMP-SC and NSGA-SC and 2) compliance minimization results from: SIMP-SVC, SIMP, UMDA, GA, and NSGA-VC. They are reported from table 7.2 to 7.14.

In result tables we report the following information: Row 1) Initial security factor(ISF), this is calculated with the whole domain with all design variables equal to 1. It is the same for all methods and relevant only for volume problems. Row 2) Number of iterations(NI). Row 3) Initial volume(IV). Row 4) Final volume(FV). Row 5) Final volume fraction(FVF), which is also the volume constraint for compliance problems. Row 6) Final compliance reached($C(x)$). Row 7) Average of the of X displacements in nodes with external forces(ADX). Row 8) Average of absolute X displacement in nodes with external forces(|ADX|). Row 9) Average of Y displacements in nodes with external forces(ADY). Row 10) Average of absolute Y displacements in nodes with external forces(|ADY|). Row 11) Average of norms of displacements in nodes with external forces(A||DXY||). Row 12) Security factor of the delivered solution(SF). Row 13) and row 14) Average and standard deviation of the elemental von Mises stresses over the yield stress(Avg(Eff) and SD(Eff) respectively). This value gives a general idea of the efficiency of the structure, the most efficient structure has an average of 1 and standard deviation of 0. Row 14) Maximal elemental Von Mises stress in structure(MaxVM). The last four values are not relevant for the compliance problems, because stresses are not considered in these problems. In addition, we report for each test: 1) material distribution, 2) displacement distribution, 3) stress distribution, and 4) histograms of the von Mises stress over yield stress, which describe the structure efficiency,(the most efficient structure has the most of these values close to 1, which means that the most of the elements are not low stressed).

Parameters for the local optimization algorithms (described in chapter 4) are: a) $p = 3$, b) $\eta = \frac{1}{2}$, c) $m = 0.2$, d) $r_{min} = 0.05$ d) V_r is reported in tables as *Final Volume Fraction* (for compliance problems it is a parameter, for volume problems, it is the minimal volume obtained), e) $v = 5$, f) convergence values are, $e = 1.0e - 2$ for SIMP and $e = 1.0e - 3$ for SIMP-SVC and SIMP-SC(for all e values), and g) $iter_{max} = 150$ for SIMP and SIMP-SVC and $iter_{max} = 250$ for SIMP-SC. Despite the derivation of the SIMP does not consider structural self-weight, tests executed in here do it, in order to consider as much as possible real-world problems.

Parameters for the global optimization methods (described in chapter 5) are:

Test	**r**	**vv**	**Control Points (Dimension size)**	**Population size**	**Max Iterations (Stopping criterion)**
CLC	0.15	0.2	200	2000	500
CLB	0.15	0.2	200	2000	500
CLT	0.15	0.2	200	2000	500
SCLC	0.15	0.1	100	1000	500
SCLB	0.15	0.1	100	1000	500
SCLT	0.15	0.1	100	1000	500
LLC	0.15	0.1	100	1000	500
LLT	0.15	0.1	100	1000	500
OLM	0.15	0.1	100	1000	500
TELM	0.15	0.1	100	1000	500
TDLM	0.15	0.1	100	1000	500
MBBB	0.15	0.3	300	1500	500
Two Bars	0.15	0.2	200	2000	500

Table 7.1: Parameters for global optimization methods

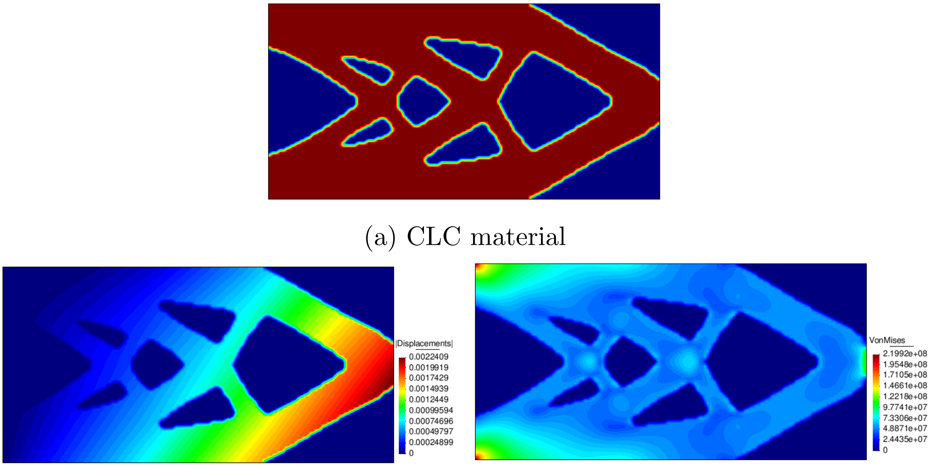
An extra experiment is carried out for OLM test. It is an hybrid (or memetic) optimization test: global + local, mixing the results of an evolutionary algorithm as the entry variables configuration for SIMP-SVC. A detailed explanation of this experiment is described in the section One Load Michell(OLM).

7.1 Cantilever with a Load at Center (CLC)

CLC test is described in chapter 4 in figure 4.1(a). First, we describe the results obtained for volume minimization and then, for compliance minimization.

7.1.1 Results for volume minimization methods

Figures 7.1 and 7.2 show results for the SIMP-SC and NSGA-SC respectively: (a) the material distribution on the design domain, (b) the normalized displacements over the structure, and c) the von Mises stresses over the structure.

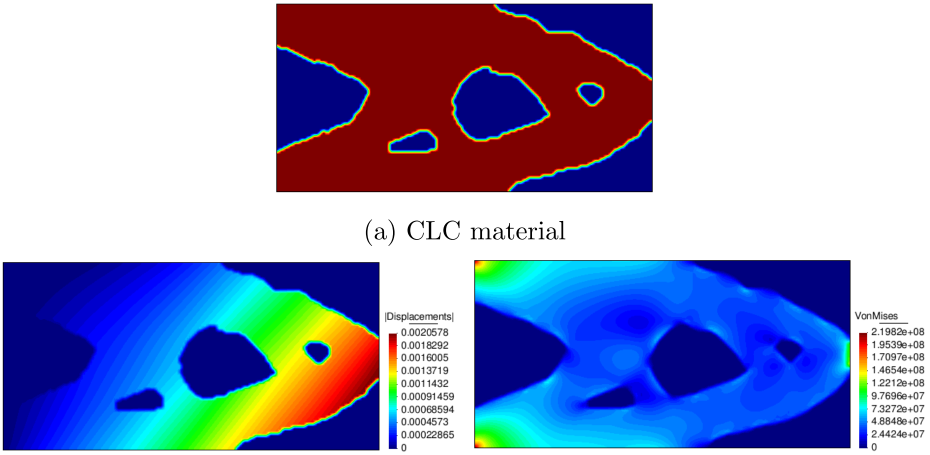


(a) CLC material

(b) CLC displacements

(c) CLC VonMises

Figure 7.1: CLC visual results with SIMP-SC



(a) CLC material

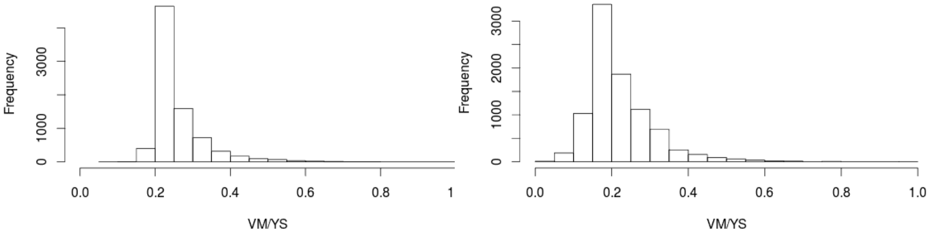
(b) CLC displacements

(c) CLC VonMises

Figure 7.2: CLC visual results with NSGA-SC

SIMP-SC shows the most aesthetic shape according to the material distribution and the best minimization performance according to the volume reached (shown in the volume minimization columns of table 7.2), where the volumes are $1.2714e0$ and $1.3879e0$ for SIMP-SC and NSGA-SC respectively.

Figures 7.3(a) and (b) show the histograms of efficiency for SIMP-SC and NSGA-SC respectively.



(a) SIMP-SC

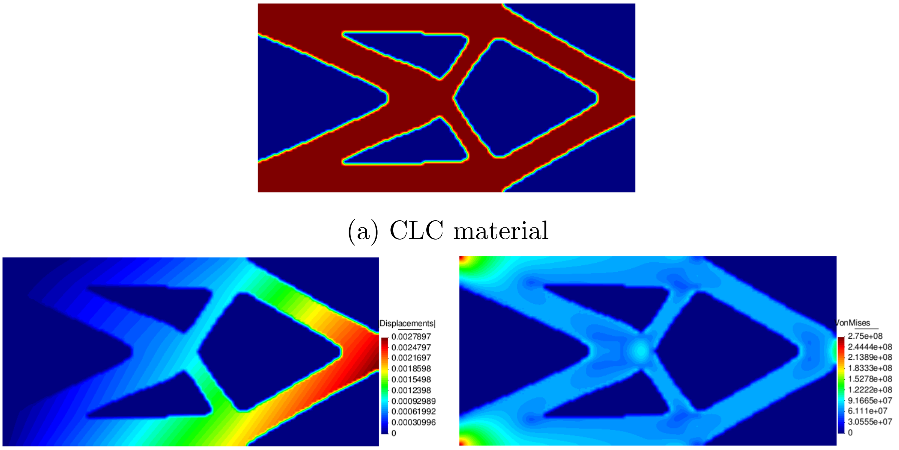
(b) NSGA-SC

Figure 7.3: CLC: VM/YS elemental histogram

SIMP-SC shows the most efficient performance, according to the histograms and the average and standard deviation (shown in the volume minimization columns of table 7.2), due to the average of elemental efficiencies is closer to 1 when using SIMP-SC than when using NSGA-SC.

7.1.2 Results for compliance minimization methods

Figures 7.4, 7.5, 7.6, 7.7 and 7.8 show results for the SIMP-SVC, SIMP, GA, UMDA and NSGA-VC, respectively: (a) the material distribution on the design domain, (b) the normalized displacements over the structure, and c) the von Mises stresses over the structure.

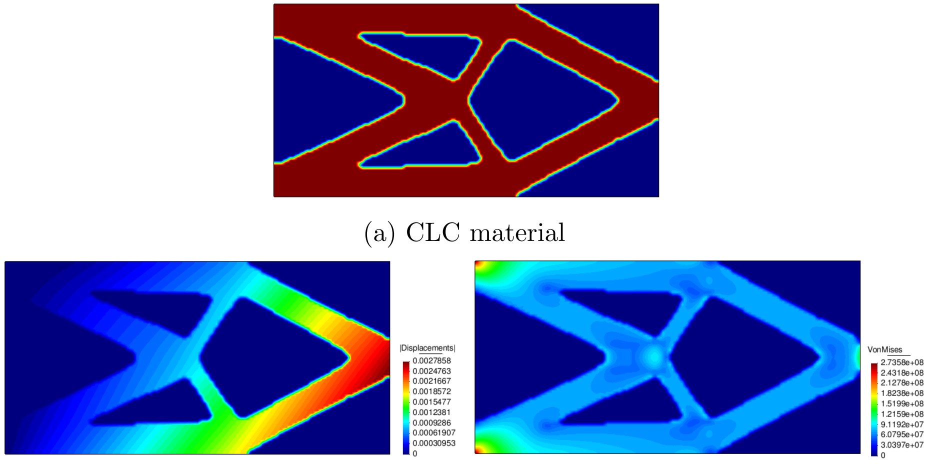


(a) CLC material

(b) CLC displacements

(c) CLC VonMises

Figure 7.4: CLC visual results with SIMP-SVC

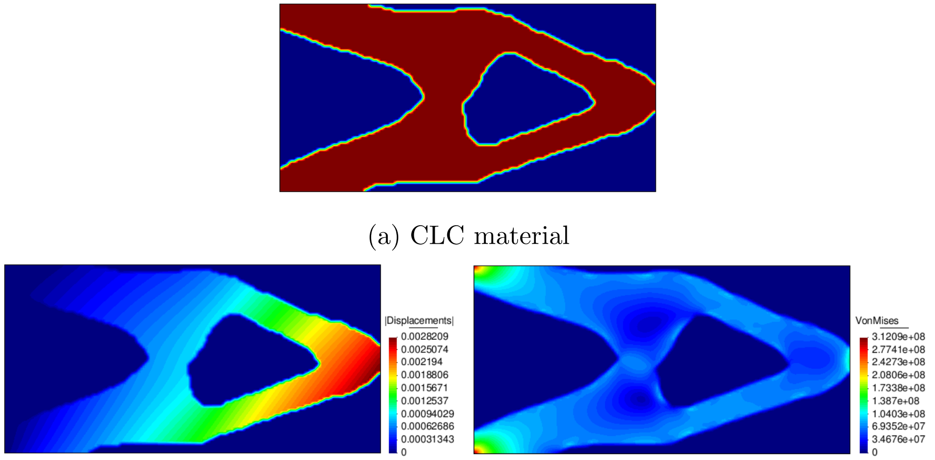


(a) CLC material

(b) CLC displacements

(c) CLC VonMises

Figure 7.5: CLC visual results with SIMP

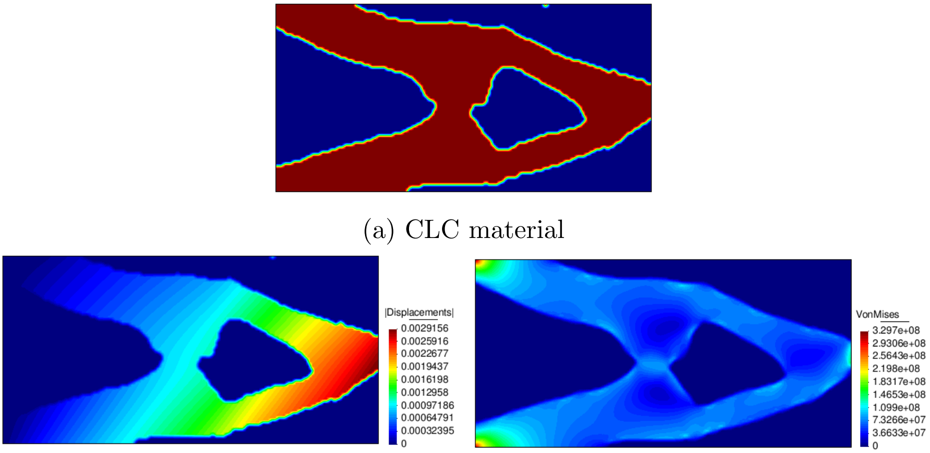


(a) CLC material

(b) CLC displacements

(c) CLC VonMises

Figure 7.6: CLC visual results with GA

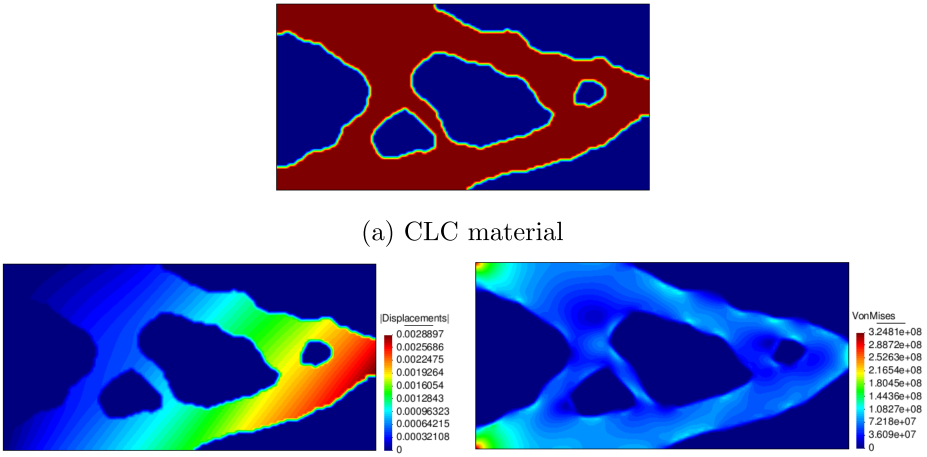


(a) CLC material

(b) CLC displacements

(c) CLC VonMises

Figure 7.7: CLC visual results with UMDA



(a) CLC material

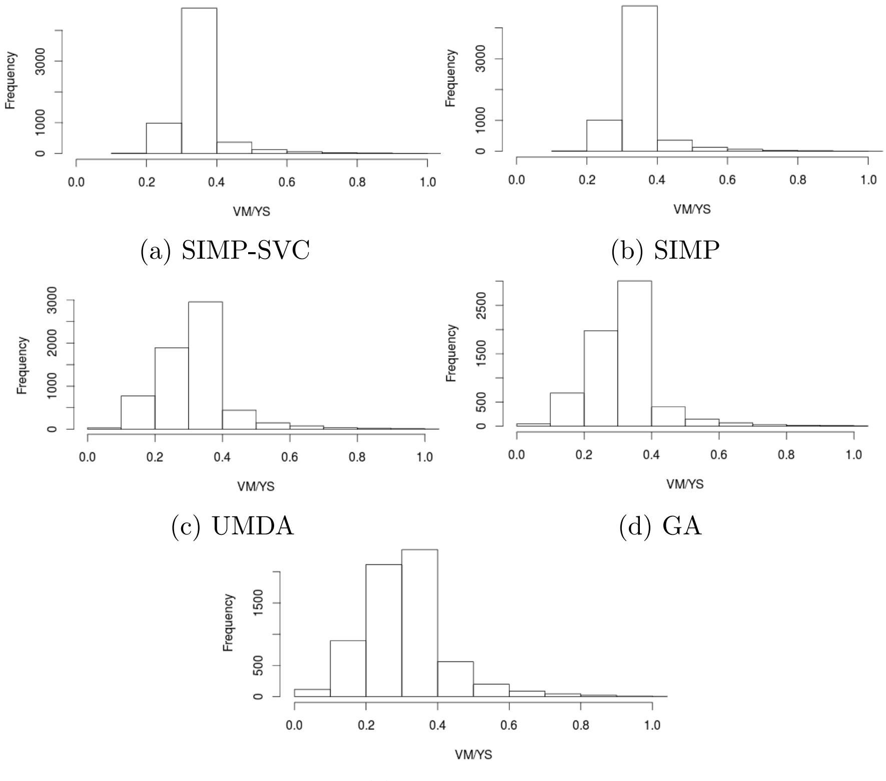
(b) CLC displacements

(c) CLC VonMises

Figure 7.8: CLC visual results with NSGA-VC

SIMP shows the most aesthetic shape according to the material distribution and the best minimization performance according to the compliance reached (shown in the compliance minimization columns of table 7.2). On the other hand, considering the iterations, SIMP does not converge within the maximal iterations, and SIMP-SVC does it in 42, saving at least the 71.34% of the iterations. The compliance reached is $2.3086e-1$ for SIMP and $2.3029e-1$ for SIMP-SVC, which do not represent a significant difference.

Figures 7.9(a), (b), (c), (d) and (e) show the histograms of efficiency for SIMP-SVC, SIMP, UMDA, GA and NSGA-VC, respectively.



(a) SIMP-SVC

(b) SIMP

(c) UMDA

(d) GA

(e) NSGA-VC

Figure 7.9: CLC: VM/YS elemental histogram

SIMP-SVC shows the most efficient performance, according to the histograms, and the average and standard deviation (shown in the compliance minimization columns of table 7.2), due to the average of elemental efficiencies is closer to 1 when using SIMP-SVC than any other.

Table 7.2 shows the details of the performance of each method. In the volume minimization columns, it is marked with blue the method that shows the best performance for volume minimization, and with red the method that shows the best performance on efficiency. In the compliance minimization columns it is marked with blue the method that shows the best performance for compliance minimization and with red the method that shows the best performance on efficiency.

	Volume min.		**Compliance min.**				
Data	**SIMP SC**	**NSGA SC**	**SIMP SVC**	**SIMP**	**UMDA**	**GA**	**NSGA VC**
ISF	7.4947e-1	-	7.4947e-1	7.4947e-1	-	-	-
NI	176	500	42	150	500	500	500
IV	2.0e0	2.0e0	2.0e0	2.0e0	2.0e0	2.0e0	2.0e0
FV	1.2714e0	1.3879e0	1.0e0	1.0e0	1.0e0	1.0e0	1.0e0
FVF	6.3574e-1	6.9395e-1	5.0e-1	5.0e-1	5.0e-1	5.0e-1	5.0e-1
$C(x)$	1.8473e2	1.6864e2	2.3086e2	2.3029e2	2.4076e2	2.3384e2	2.3748e2
ADX	-3.2311e-10	-2.6413e-6	-1.9895e-10	-6.7693e-10	1.4694e-5	-1.6610e-5	1.7415e-5
|ADX|	3.7892e-5	3.8368e-5	4.3923e-5	4.5094e-5	5.7588e-5	5.4477e-5	5.8972e-5
ADY	-1.9256e-2	-1.7570e-2	-2.4083e-2	-2.4024e-2	-2.5109e-2	-2.4392e-2	-2.4773e-2
|ADY|	1.9256e-2	1.7570e-2	2.4083e-2	2.4024e-2	2.5109e-2	2.4392e-2	2.4773e-2
A||DXY||	2.1400e-3	1.9527e-3	2.6764e-3	2.6698e-3	2.7907e-3	2.7110e-3	2.7534e-3
SF	9.9962e-1	9.9916e-1	1.2499e0	1.2435e0	1.4986e0	1.4185e0	1.4764e0
A(Eff)	2.2621e-1	2.2123e-1	3.4061e-1	3.3977e-1	3.1501e-1	3.1178e-1	3.0993e-1
SD(Eff)	6.9561e-2	8.6076e-2	7.2220e-2	7.2117e-2	1.1264e-1	1.0844e-1	1.2082e-1
MaxVM	2.1991e8	2.1981e8	2.7499e8	2.7357e8	3.2969e8	3.1208e8	3.2481e8

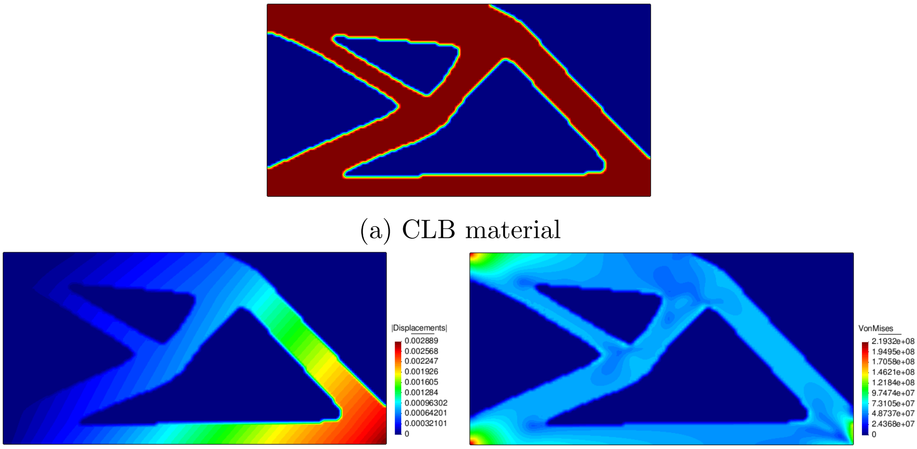
Table 7.2: CLC execution data

7.2 Cantilever with a Load at Bottom(CLB)

CLB test is described in chapter 4 in figure 4.1(b). First, we describe the results obtained for volume minimization and then, for compliance minimization.

7.2.1 Results for volume minimization methods

Figures 7.10 and 7.11 show results for the SIMP-SC and NSGA-SC respectively: (a) the material distribution on the design domain, (b) the normalized displacements over the structure, and c) the von Mises stresses over the structure.

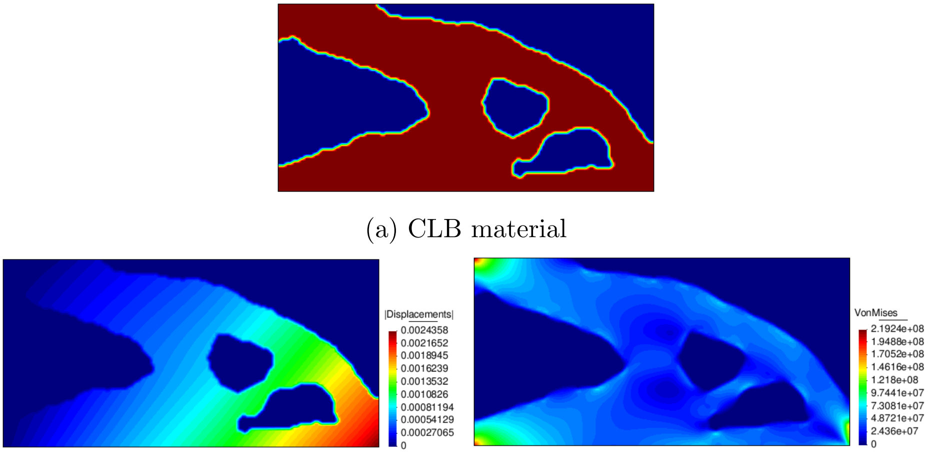


(a) CLB material

(b) CLB displacements

(c) CLB VonMises

Figure 7.10: CLB visual results with SIMP-SC



(a) CLB material

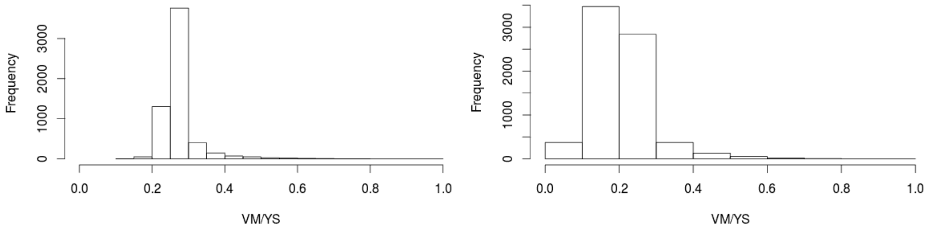
(b) CLB displacements

(c) CLB VonMises

Figure 7.11: CLB visual results with NSGA-SC

SIMP-SC shows the most aesthetic shape according to the material distribution and the best minimization performance according to the volume reached (shown in the volume minimization columns of table 7.3), where the volumes are $9.2578e-1$ and $1.1354e0$ for SIMP-SC and NSGA-SC respectively.

Figures 7.12(a) and (b) show the histograms of efficiency for SIMP-SC and NSGA-SC respectively.



(a) SIMP-SC

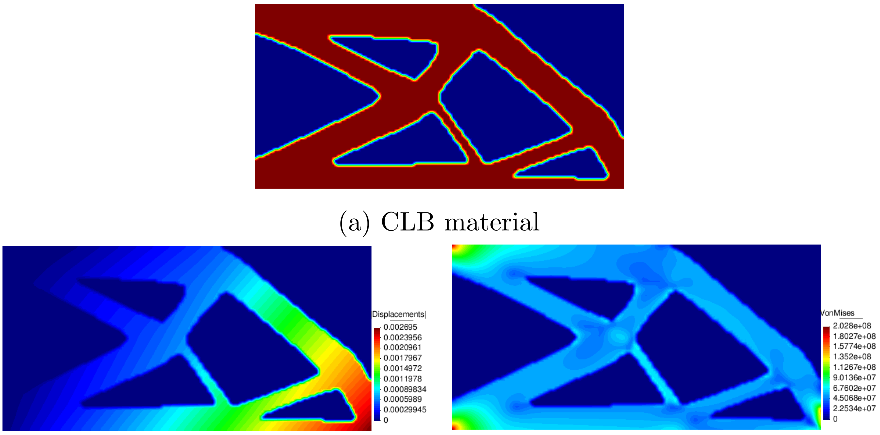
(b) NSGA-SC

Figure 7.12: CLB: VM/YS elemental histogram

SIMP-SC shows the most efficient performance, according to the histograms and the average and standard deviation (shown in the volume minimization columns of table 7.3), due to the average of elemental efficiencies is closer to 1 when using SIMP-SC than when using NSGA-SC.

7.2.2 Results for compliance minimization methods

Figures 7.13, 7.14, 7.15, 7.16 and 7.17 show results for the SIMP-SVC, SIMP, GA, UMDA and NSGA-VC, respectively: (a) the material distribution on the design domain, (b) the normalized displacements over the structure, and c) the von Mises stresses over the structure.

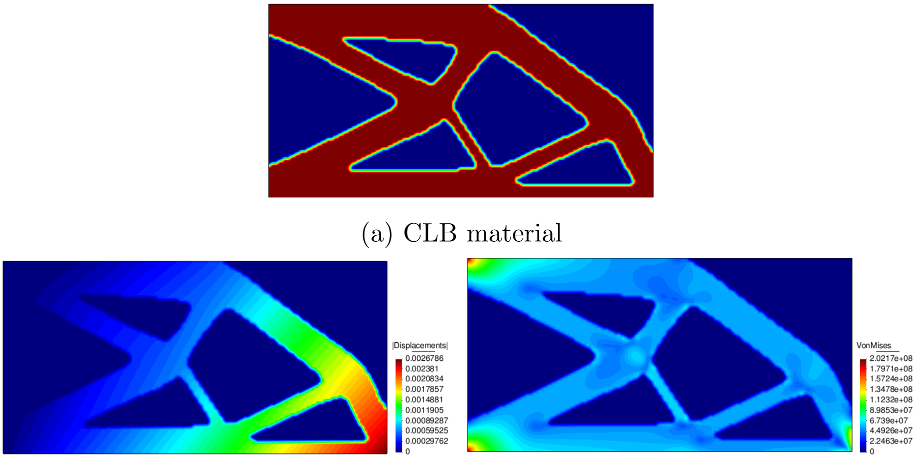


(a) CLB material

(b) CLB displacements

(c) CLB VonMises

Figure 7.13: CLB visual results with SIMP-SVC

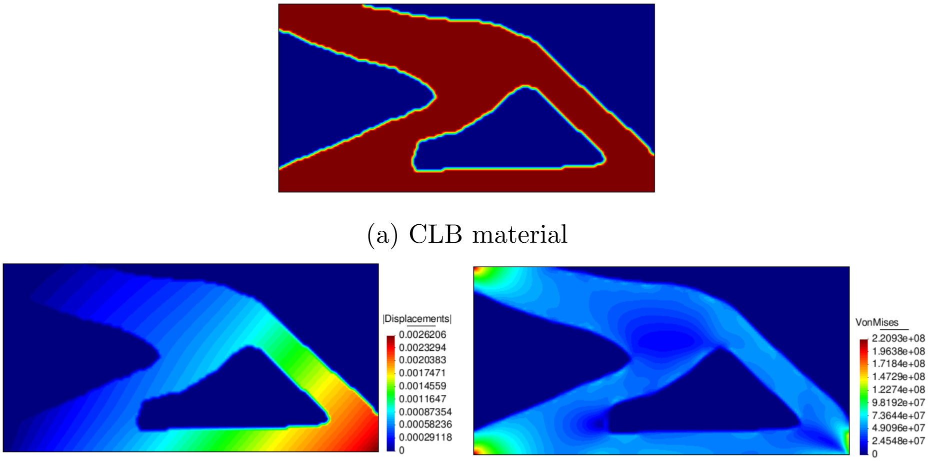


(a) CLB material

(b) CLB displacements

(c) CLB VonMises

Figure 7.14: CLB visual results with SIMP

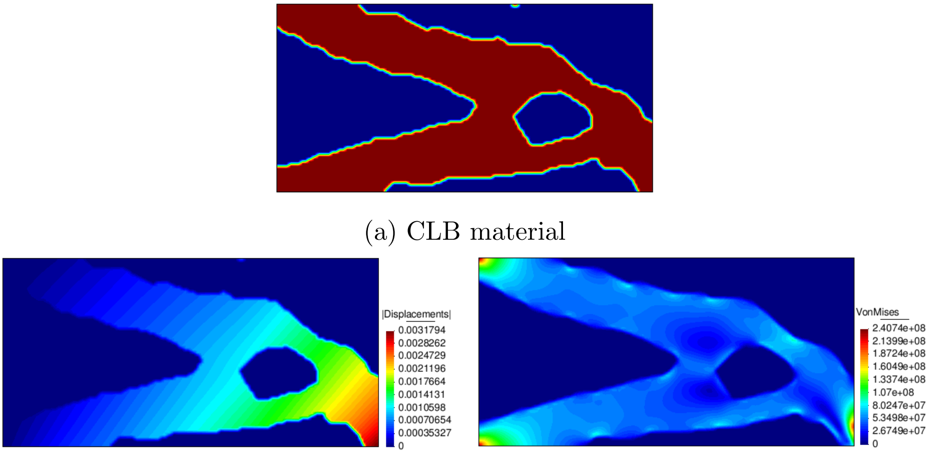


(a) CLB material

(b) CLB displacements

(c) CLB VonMises

Figure 7.15: CLB visual results with GA

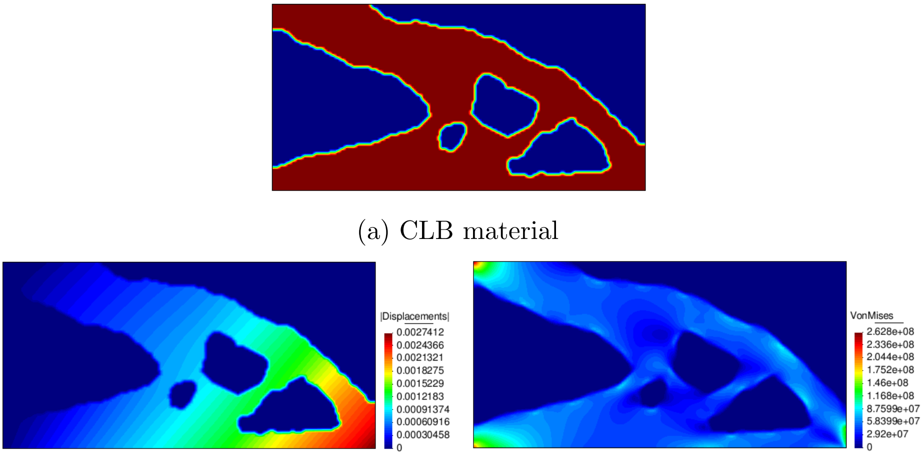


(a) CLB material

(b) CLB displacements

(c) CLB VonMises

Figure 7.16: CLB visual results with UMDA



(a) CLB material

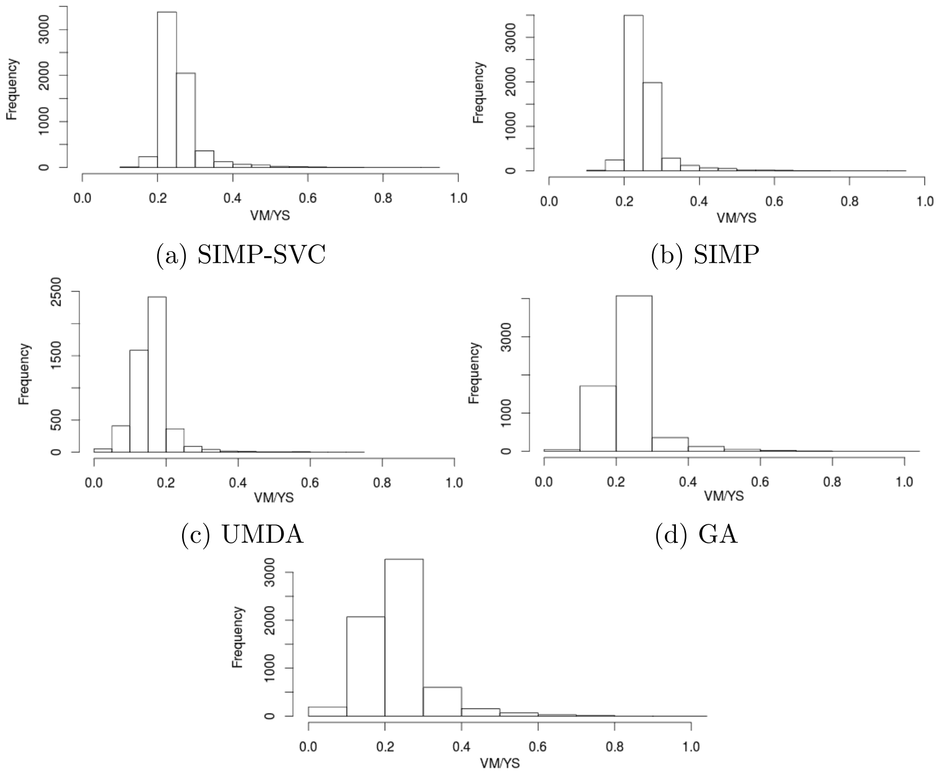
(b) CLB displacements

(c) CLB VonMises

Figure 7.17: CLB visual results with NSGA-VC

On one hand, SIMP-SVC and SIMP show the most aesthetic shape according to the material distribution, on the other hand, the GA shows the best minimization performance according to the compliance reached (shown in the compliance minimization columns of table 7.3). Note that SIMP does not converge within the maximal iterations, and SIMP-SVC does it in 39, saving at least the 74% of the iterations. The compliance reached is $1.2828e2$ for SIMP and $1.2911e2$ for SIMP-SVC, which do not represent a significant difference.

Figures 7.18(a), (b), (c), (d) and (e) show the histograms of efficiency for SIMP-SVC, SIMP, UMDA, GA and NSGA-VC, respectively.



(a) SIMP-SVC

(b) SIMP

(c) UMDA

(d) GA

(e) NSGA-VC

Figure 7.18: CLB: VM/YS elemental histogram

SIMP-SVC shows the most efficient performance, according to the histograms, and the average and standard deviation (shown in the compliance minimization columns of table 7.3), due to the average of elemental efficiencies is closer to 1 when using SIMP-SVC than any other.

Table 7.3 has the details of the performance of each method. In the volume minimization columns, it is marked with blue the method that shows the best performance for volume minimization, and with red the method that shows the best performance on efficiency. In the compliance minimization columns it is marked with blue the method that shows the best performance for compliance minimization and with red the method that shows the best performance on efficiency.

	Volume min.		**Compliance min.**				
Data	**SIMP SC**	**NSGA SC**	**SIMP SVC**	**SIMP**	**UMDA**	**GA**	**NSGA VC**
ISF	7.5264e-1	-	7.5264e-1	7.5264e-1	-	-	
NI	167	500	39	150	500	500	500
IV	2.0e0	2.0e0	2.0e0	2.0e0	2.0e0	2.0e0	2.0e0
FV	9.2578e-1	1.1354e0	1.0e0	1.0e0	1.0e0	1.0e0	1.0e0
FVF	4.6289e-1	5.6770e-1	5.0e-1	5.0e-1	5.0e-1	5.0e-1	5.0e-1
$C(x)$	1.3955e2	1.1602e2	1.2911e2	1.2828e2	1.4255e2	1.2499e2	1.3176e2
ADX	-5.6312e-4	-4.8702e-4	-5.3832e-4	-5.3654e-4	-7.4513e-4	-5.2506e-4	-5.3909e-4
|ADX|	5.6312e-4	4.8702e-4	5.3832e-4	5.3654e-4	7.4513e-4	5.2506e-4	5.3909e-4
ADY	-2.0173e-2	-1.6755e-2	-1.8660e-2	-1.8540e-2	-2.0598e-2	-1.8062e-2	-1.9037e-2
|ADY|	2.0173e-2	1.6755e-2	1.8660e-2	1.8540e-2	2.0598e-2	1.8062e-2	1.9037e-2
A|DXY|	2.311752e-3	1.9250e3	2.1427e-3	2.1293e-3	2.4087e-3	2.0752e-3	2.1835e-3
SF	9.9689e-1	9.9656e-1	9.2184e-1	9.1895e-1	1.0942e0	1.0042e0	1.1945e0
A(Eff)	2.7648e-1	2.0310e-1	2.5631e-1	2.5515e-1	2.4143e-1	2.2981e-1	2.3193e-1
SD(Eff	5.6441e-2	7.9881e-2	5.4746e-2	5.4179e-2	8.7452e-2	7.5448e-2	8.8541e-2
MaxVM	2.1931e8	2.1924e8	2.0280e8	2.0216e8	2.4074e8	2.2093e8	2.7279e8

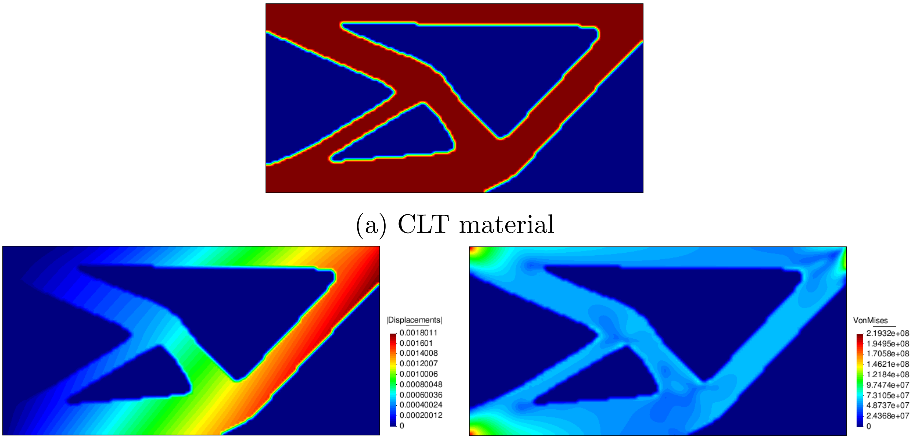
Table 7.3: CLB execution data

7.3 Cantilever with a Load at Top (CLT)

CLT test is described in chapter 4 in figure 4.1(c). First, we describe the results obtained for volume minimization and then, for compliance minimization.

7.3.1 Results for volume minimization methods

Figures 7.19 and 7.20 show results for the SIMP-SC and NSGA-SC respectively: (a) the material distribution on the design domain, (b) the normalized displacements over the structure, and c) the von Mises stresses over the structure.

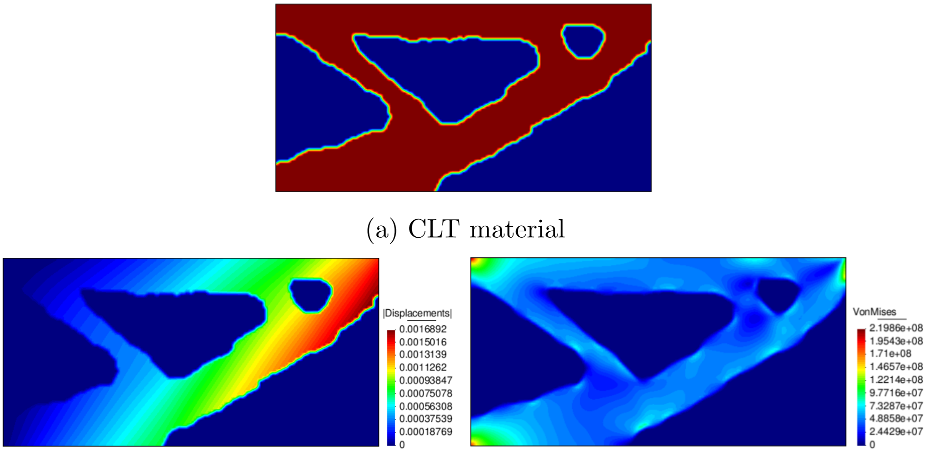


(a) CLT material

(b) CLT displacements

(c) CLT VonMises

Figure 7.19: CLT visual results with SIMP-SC



(a) CLT material

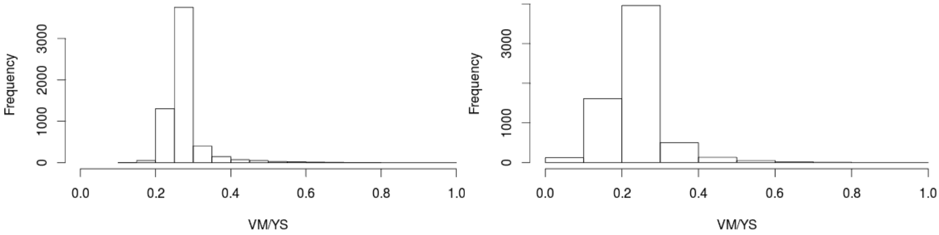
(b) CLT displacements

(c) CLT VonMises

Figure 7.20: CLT visual results with NSGA-SC

SIMP-SC shows the most aesthetic shape according to the material distribution and the best minimization performance according to the volume reached (shown in the volume minimization columns of table 7.4), where the volumes are $9.257e-1$ and $1.0e0$ for SIMP-SC and NSGA-SC respectively.

Figures 7.21(a) and (b) show the histograms of efficiency for SIMP-SC and NSGA-SC respectively.



(a) SIMP-SC

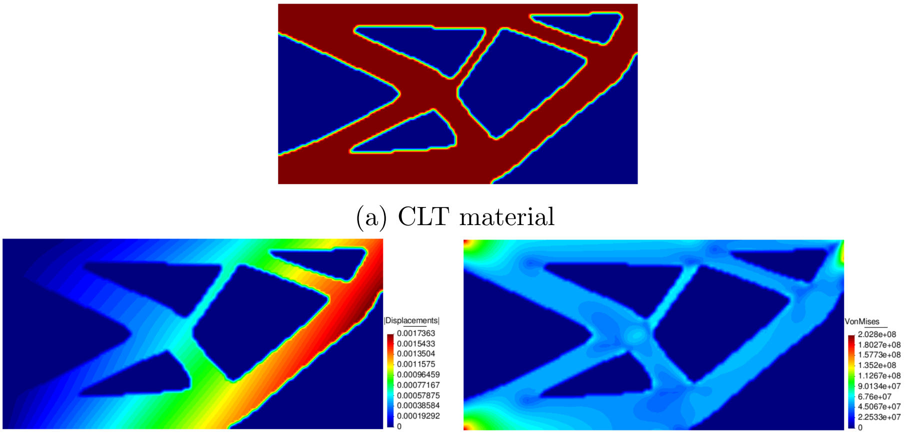
(b) NSGA-SC

Figure 7.21: CLT: VM/YS elemental histogram

SIMP-SC shows the most efficient performance, according to the histograms and the average and standard deviation (shown in the volume minimization columns of table 7.4), due to the average of elemental efficiencies is closer to 1 when using SIMP-SC than when using NSGA-SC.

7.3.2 Results for compliance minimization methods

Figures 7.22, 7.23, 7.24, 7.25 and 7.26 show results for the SIMP-SVC, SIMP, GA, UMDA and NSGA-VC, respectively: (a) the material distribution on the design domain, (b) the normalized displacements over the structure, and c) the von Mises stresses over the structure.

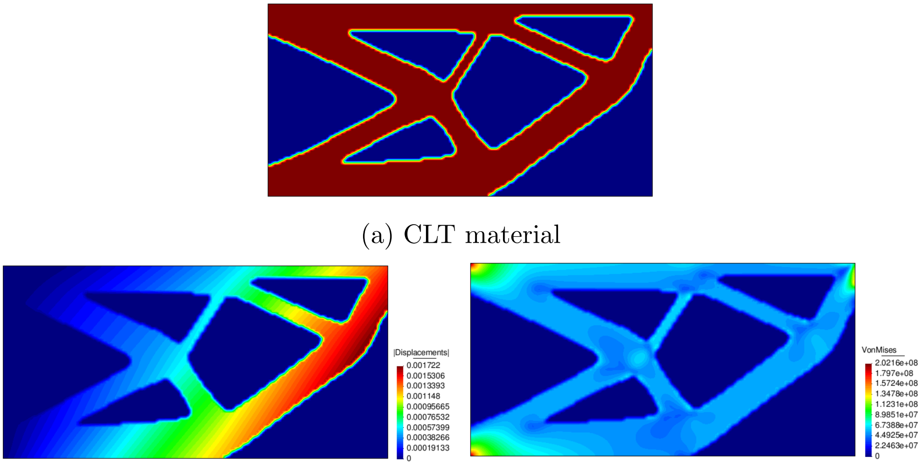


(a) CLT material

(b) CLT displacements

(c) CLT VonMises

Figure 7.22: CLT visual results with SIMP-SVC

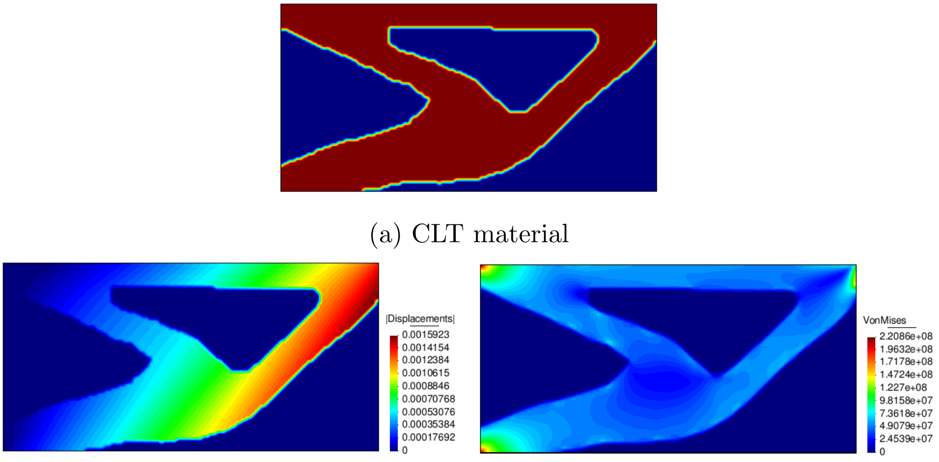


(a) CLT material

(b) CLT displacements

(c) CLT VonMises

Figure 7.23: CLT visual results with SIMP

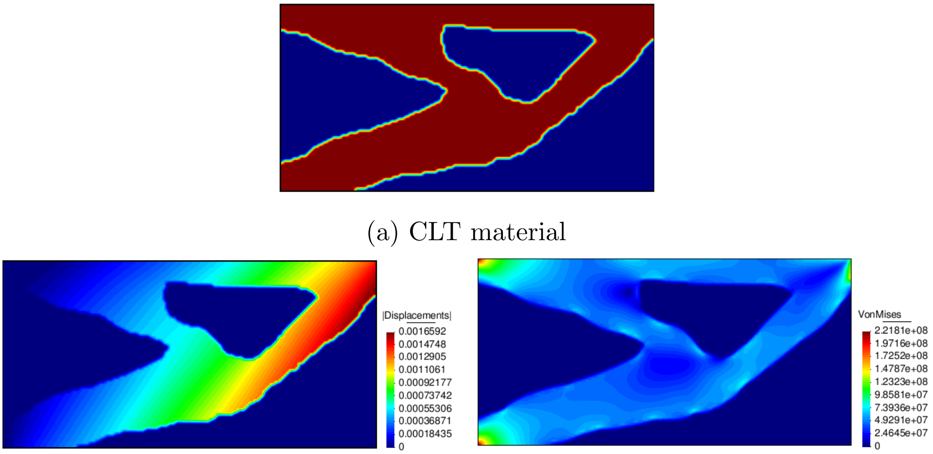


(a) CLT material

(b) CLT displacements

(c) CLT VonMises

Figure 7.24: CLT visual results with GA

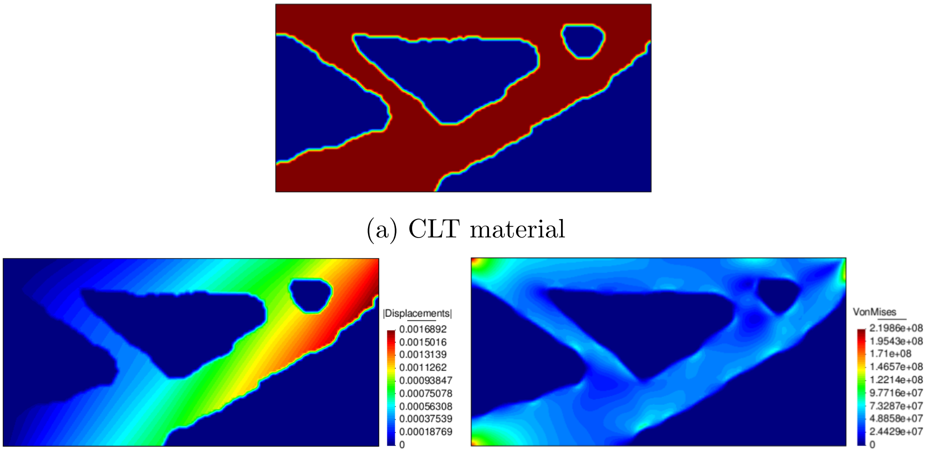


(a) CLT material

(b) CLT displacements

(c) CLT VonMises

Figure 7.25: CLT visual results with UMDA



(a) CLT material

(b) CLT displacements

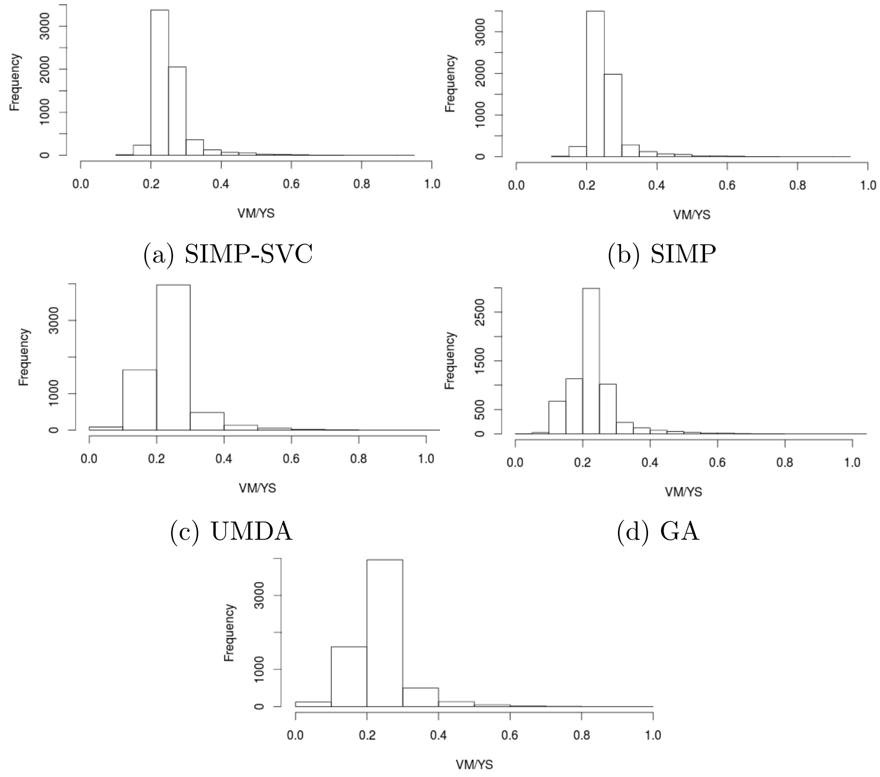
(c) CLT VonMises

Figure 7.26: CLT visual results with NSGA-VC

On one hand, SIMP-SVC and SIMP show the most aesthetic shape according to the material distribution, on the other hand, the GA shows the best minimization performance according to the compliance reached (shown in the compliance minimization columns of table 7.4).

Note that SIMP does not converge within the maximal iterations, and SIMP-SVC does it in 39, saving at least the 74% of the iterations. The compliance reached is $1.2828e2$ for SIMP and $1.2911e2$ for SIMP-SVC, which do not represent a significant difference.

Figures 7.27(a), (b), (c), (d) and (e) show the histograms of efficiency for SIMP-SVC, SIMP, UMDA, GA and NSGA-VC, respectively.



(a) SIMP-SVC

(b) SIMP

(c) UMDA

(d) GA

(e) NSGA-VC

Figure 7.27: CLT: VM/YS elemental histogram

SIMP-SVC shows the most efficient performance, according to the histograms, and the average and standard deviation (shown in the compliance minimization columns of table 7.4), due to the most of the values are closer to 1 when using SIMP-SVC than any other.

Table 7.4 has the details of the performance of each method. In the volume minimization columns, it is marked with blue the method that shows the best performance for volume minimization, and with red the method that shows the best performance on efficiency. In the compliance minimization columns it is marked with blue the method that shows the best performance for compliance minimization and with red the method that shows the best performance on efficiency.

	Volume min.		**Compliance min.**				
Data	**SIMP SC**	**NSGA SC**	**SIMP SVC**	**SIMP**	**UMDA**	**GA**	**NSGA VC**
ISF	7.5264e-1	-	7.5264e-1	7.5264e-1	-	-	
NI	167	500	39	150	500	500	500
IV	2.0e0	2.0e0	2.0e0	2.0e0	2.0e0	2.0e0	2.0e0
FV	9.257e-1	1.0e0	1.0e0	1.0e0	1.0e0	1.0e0	1.0e0
FVF	4.6289e-1	5.0e-1	5.0e-1	5.0e-1	5.0e-1	5.0e-1	5.0e-1
$C(x)$	1.3955e2	1.2803e2	1.2911e2	1.2828e2	1.2906e2	1.2390e2	1.2803e2
ADX	5.6312e-4	5.2836e-4	5.3832e-4	5.3656e-4	5.4327e-4	5.2492e-4	5.2836e-4
|ADX|	5.6312e-4	5.2836e-4	5.3832e-4	5.3656e-4	5.4327e-4	5.2492e-4	5.2836e-4
ADY	-2.0173e-2	-1.8504e-2	-1.8659e-2	-1.8540e-2	-1.8651e-2	-1.7906e-2	-1.8504e-2
|ADY|	2.0173e-2	1.8504e-2	1.8659e-2	1.8540e-2	1.8651e-2	1.7906e-2	1.8504e-2
A||DXY||	2.3117e-3	2.1236e-3	2.1426e-3	2.1293e-3	2.1431e-3	2.0584e-3	2.1236e-3
SF	9.9688e-1	9.9936e-1	9.2182e-1	91892e-1	1.0082e0	1.0038e0	9.9936e-1
A(Eff)	2.7649e-1	2.3371e-1	2.5630e-1	2.5513e-1	2.3295e-1	2.2942e-1	2.3371e-1
SD(Eff)	5.6441e-2	7.4658e-2	5.4744e-2	5.4175e-2	7.7602e-2	7.3949e-2	7.4658e-2
MaxVM	2.1931e8	2.1986e8	2.0280e8	2.0216e8	2.2180e8	2.2085e8	2.1986e8

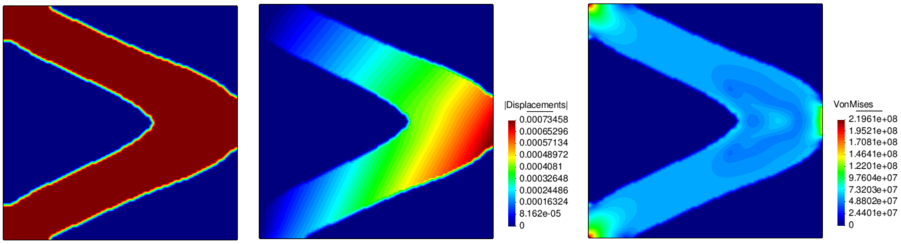
Table 7.4: CLT execution data

7.4 Short Cantilever with a Load at Center(SCLC)

SCLC test is described in chapter 4 in figure 4.2(a). First, we describe the results obtained for volume minimization and then, for compliance minimization.

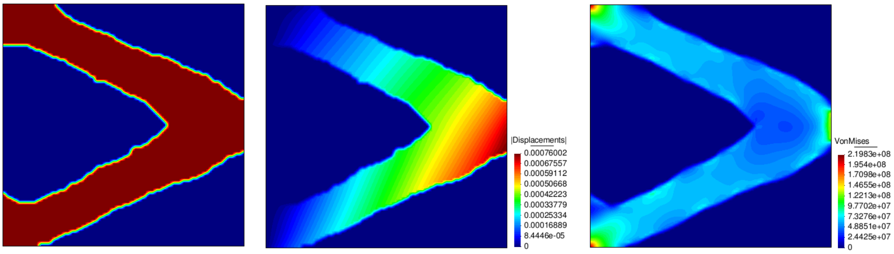
7.4.1 Results for volume minimization methods

Figures 7.28 and 7.29 show results for the SIMP-SC and NSGA-SC respectively: (a) the material distribution on the design domain, (b) the normalized displacements over the structure, and c) the von Mises stresses over the structure.



(a) SCLC material (b) SCLC displacements (c) SCLC VonMises

Figure 7.28: SCLC visual results with SIMP-SC

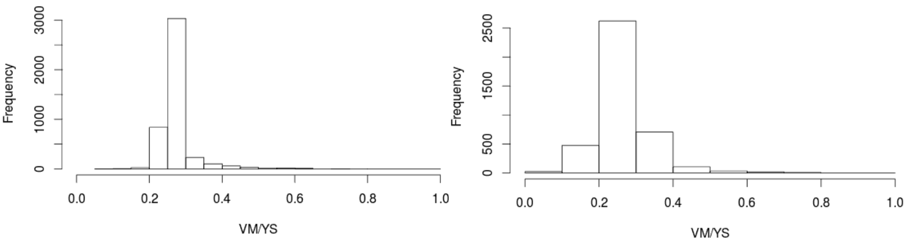


(a) SCLC material (b) SCLC displacements (c) SCLC VonMises

Figure 7.29: SCLC visual results with NSGA-SC

For this case, we consider that there is a high similarity between visual results of SIMP-SC and NSGA-SC according to the material distribution, nevertheless, the NSGA-SC shows the best minimization performance according to the volume reached (shown in the volume minimization columns of table 7.5), where the volumes are $4.3945e{-1}$ and $3.9890e{-1}$ for SIMP-SC and NSGA-SC respectively.

Figures 7.30(a) and (b) show the histograms of efficiency for SIMP-SC and NSGA-SC respectively.



(a) SIMP-SC

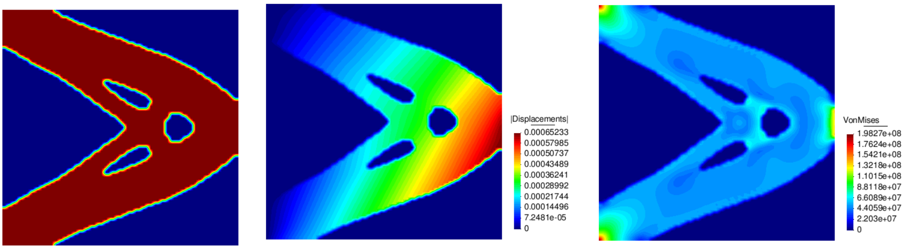
(b) NSGA-SC

Figure 7.30: SCLC: VM/YS elemental histogram

SIMP-SC shows the most efficient performance, according to the histograms and the average and standard deviation (shown in the volume minimization columns of table 7.5), due to the average of elemental efficiencies is closer to 1 when using SIMP-SC than when using NSGA-SC.

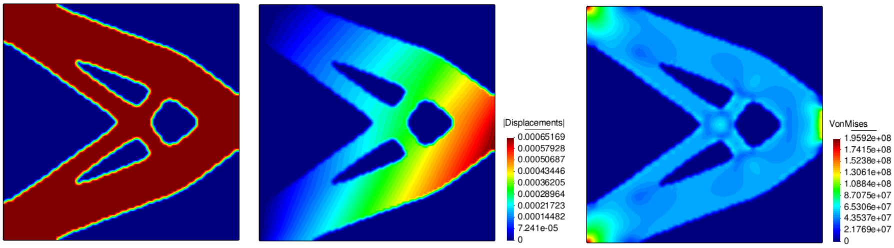
7.4.2 Results for compliance minimization methods

Figures 7.31, 7.32, 7.33, 7.34 and 7.35 show results for the SIMP-SVC, SIMP, GA, UMDA and NSGA-VC, respectively: (a) the material distribution on the design domain, (b) the normalized displacements over the structure, and c) the von Mises stresses over the structure.



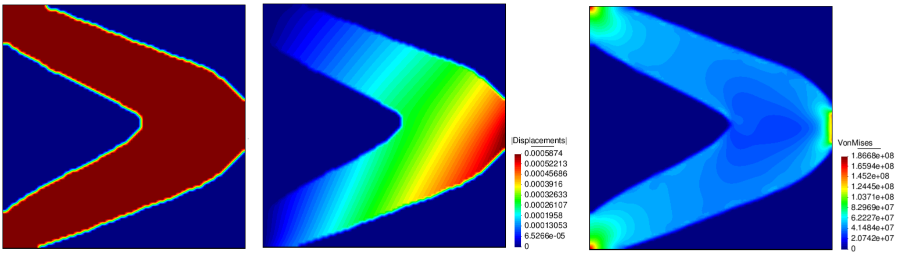
(a) SCLC material (b) SCLC displacements (c) SCLC VonMises

Figure 7.31: SCLC visual results with SIMP-SVC



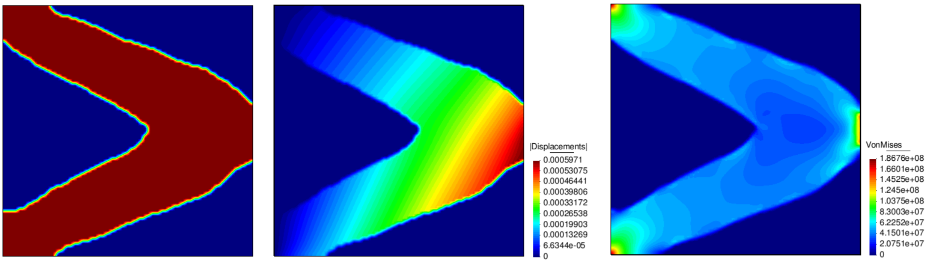
(a) SCLC material (b) SCLC displacements (c) SCLC VonMises

Figure 7.32: SCLC visual results with SIMP



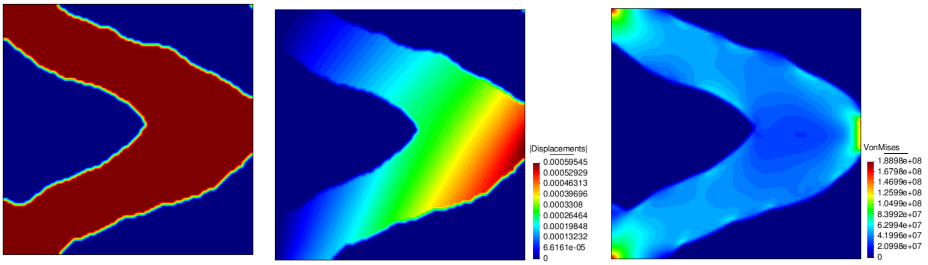
(a) SCLC material (b) SCLC displacements (c) SCLC VonMises

Figure 7.33: SCLC visual results with GA



(a) SCLC material (b) SCLC displacements (c) SCLC VonMises

Figure 7.34: SCLC visual results with UMDA



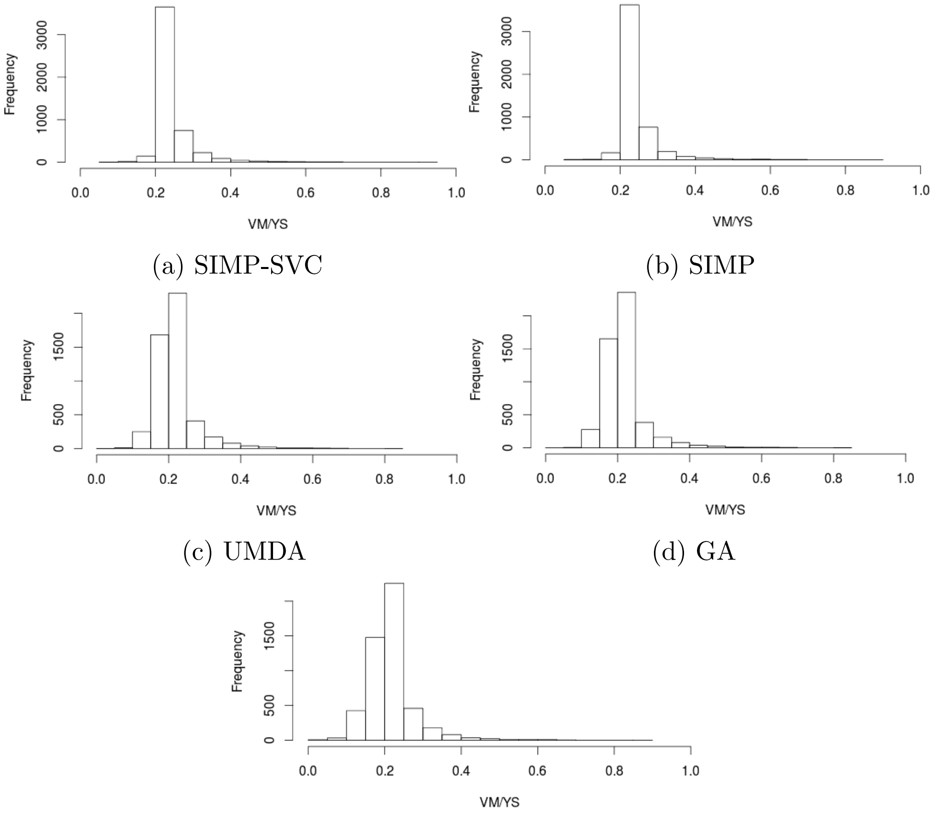
(a) SCLC material (b) SCLC displacements (c) SCLC VonMises

Figure 7.35: SCLC visual results with NSGA-VC

On one hand, SIMP-SVC and SIMP show the most aesthetic shape according to the material distribution, on the other hand, the GA shows the best minimization performance according to the compliance reached (shown in the compliance minimization columns of table 7.5).

Note that in table 7.5, SIMP-SVC reaches a better compliance than SIMP, even when SIMP has reached the maximal iterations, in addition, SIMP-SVC converges in 27 iterations, saving at least the 82% of the maximal number. The compliance reached is $5.6164e1$ for SIMP and $5.6125e1$ for SIMP-SVC, which do not represent a significant difference.

Figures 7.36(a), (b), (c), (d) and (e) show the histograms of efficiency for SIMP-SVC, SIMP, UMDA, GA and NSGA-VC, respectively.



(a) SIMP-SVC

(b) SIMP

(c) UMDA

(d) GA

(e) NSGA-VC

Figure 7.36: SCLC: VM/YS elemental histogram

SIMP shows the most efficient performance, according to the histograms, and the average and standard deviation (shown in the compliance minimization columns of table 7.5), due to the average of elemental efficiencies is closer to 1 when using SIMP than any other.

Table 7.5 has the details of the performance of each method. In the volume minimization columns, it is marked with blue the method that shows the best performance for volume minimization, and with red the method that shows the best performance on efficiency. In the compliance minimization columns it is marked with blue the method that shows the best performance for compliance minimization and with red the method that shows the best performance on efficiency.

	Volume min.		**Compliance min.**				
Data	**SIMP SC**	**NSGA SC**	**SIMP SVC**	**SIMP**	**UMDA**	**GA**	**NSGA VC**
ISF	7.5516e-1	-	7.5516e-1	7.5516e-1	-	-	
NI	138	500	27	150	500	500	500
IV	1.0e0	1.0e0	1.0e0	1.0e0	1.0e0	1.0e0	1.0e0
FV	4.3945e-1	3.9890e-1	5.0e-1	5.0e-1	5.0e-1	5.0e-1	5.0e-1
FVF	4.39453e-1	3.9890e-1	5.0e-1	5.0e-1	5.0e-1	5.0e-1	5.0e0
$C(x)$	6.2921e1	6.4176e1	5.6125e1	5.6164e1	5.1147e1	5.0920e1	5.1667e1
ADX	9.6287e-9	-2.4251e-6	-1.2925e-8	6.2024e-9	1.4412e-6	7.5318e-7	2.7494e-6
|ADX|	1.8152e-5	2.0347e-5	1.5652e-5	1.5125e-5	1.7288e-5	1.6953e-5	1.7714e-5
ADY	-7.6768e-3	-7.8315e-3	-6.8468e-3	-6.8517e-3	-6.2396e-3	-6.2119e-3	-6.3032e-3
|ADY|	7.6768e-3	7.8315e-3	6.8468e-3	6.8517e-3	6.2396e-3	6.2119e-3	6.3032e-3
A||DXY||	6.9821e-4	7.1234e-4	6.2271e-4	6.2313e-4	5.6759e-4	5.6506e-4	5.7338e-4
SF	9.9822e-1	9.9922e-1	9.0120e-1	8.9053e-1	8.4889e-1	8.4854e-1	8.5900e-1
A(Eff)	2.7610e-1	2.7195e-1	2.4709e-1	2.4739e-1	2.1598e-1	2.1569e-1	2.1682e-1
SD(Eff)	5.4350e-2	7.2235e-2	5.2463e-2	5.0792e-2	6.0822e-2	5.9772e-2	6.2438e-2
MaxVM	2.1960e8	2.1982e8	1.9826e8	1.9591e8	1.8675e8	1.8668e+8	1.8898e8

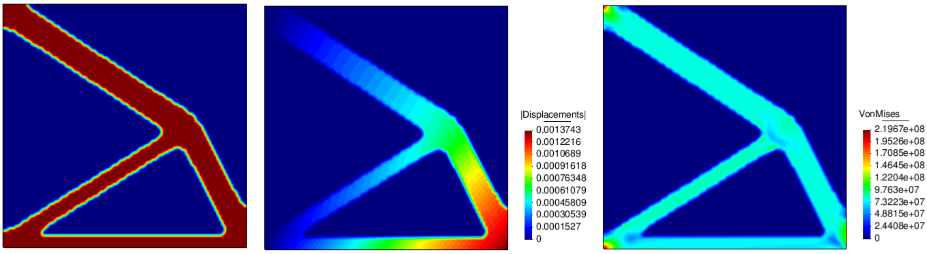
Table 7.5: SCLC execution data

7.5 Short Cantilever with a Load at Bottom(SCLB)

SCLB test is described in chapter 4 in figure 4.2(b). First, we describe the results obtained for volume minimization and then, for compliance minimization.

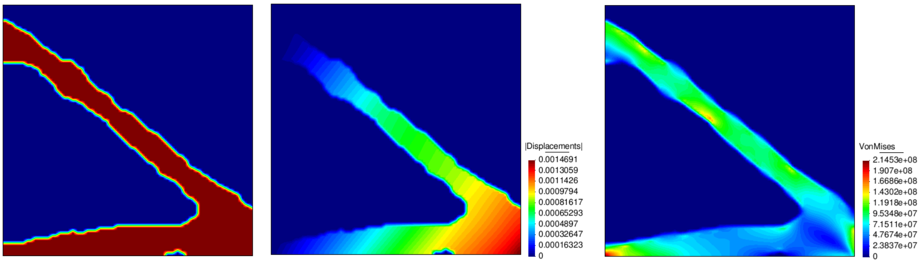
7.5.1 Results for volume minimization methods

Figures 7.37 and 7.38 show results for the SIMP-SC and NSGA-SC respectively: (a) the material distribution on the design domain, (b) the normalized displacements over the structure, and c) the von Mises stresses over the structure.



(a) SCLB material (b) SCLB displacements (c) SCLB VonMises

Figure 7.37: SCLB visual results with SIMP-SC

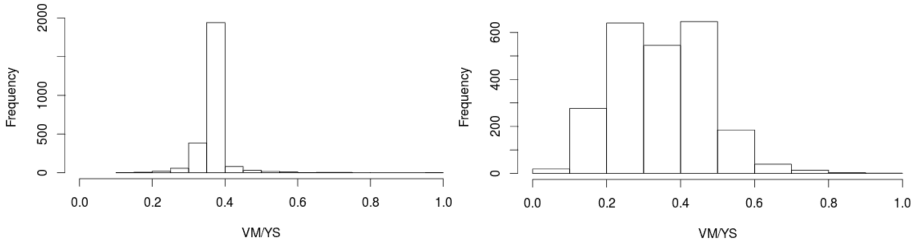


(a) SCLB material (b) SCLB displacements (c) SCLB VonMises

Figure 7.38: SCLB visual results with NSGA-SC

On one hand, SIMP-SC shows the most aesthetic shape according to the material distribution, on the other hand, NSGA-SC shows the best minimization performance according to the volume reached (shown in the volume minimization columns of table 7.6), where the volumes are $2.6367e-1$ and $2.3690e-1$ for SIMP-SC and NSGA-SC respectively.

Figures 7.39(a) and (b) show the histograms of efficiency for SIMP-SC and NSGA-SC respectively.



(a) SIMP-SC

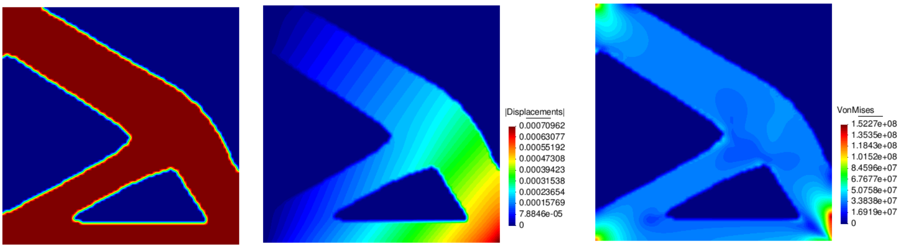
(b) NSGA-SC

Figure 7.39: SCLB: VM/YS elemental histogram

SIMP-SC shows the most efficient performance, according to the histograms and the average and standard deviation (shown in the volume minimization columns of table 7.6), due to the average of elemental efficiencies is closer to 1 when using SIMP-SC than when using NSGA-SC.

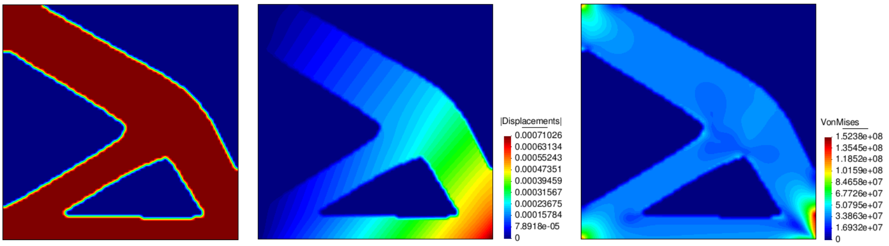
7.5.2 Results for compliance minimization methods

Figures 7.40, 7.41, 7.42, 7.43 and 7.44 show results for the SIMP-SVC, SIMP, GA, UMDA and NSGA-VC, respectively: (a) the material distribution on the design domain, (b) the normalized displacements over the structure, and c) the von Mises stresses over the structure.



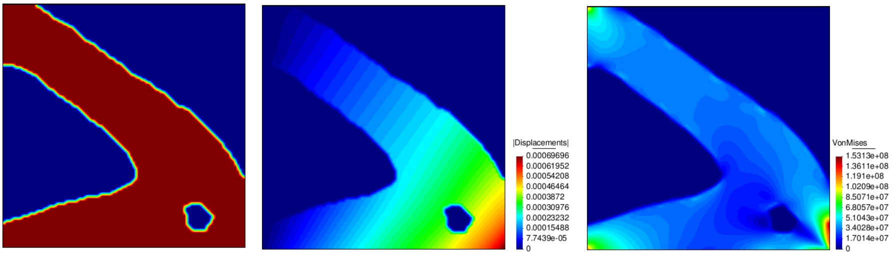
(a) SCLB material (b) SCLB displacements (c) SCLB VonMises

Figure 7.40: SCLB visual results with SIMP-SVC



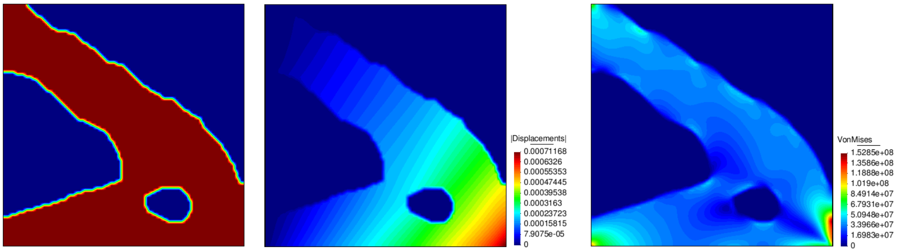
(a) SCLB material (b) SCLB displacements (c) SCLB VonMises

Figure 7.41: SCLB visual results with SIMP



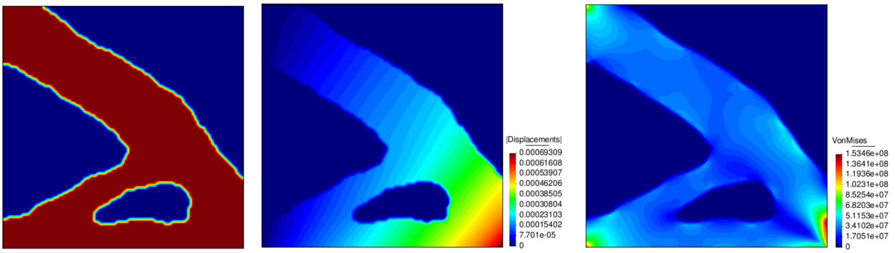
(a) SCLB material (b) SCLB displacements (c) SCLB VonMises

Figure 7.42: SCLB visual results with GA



(a) SCLB material (b) SCLB displacements (c) SCLB VonMises

Figure 7.43: SCLB visual results with UMDA



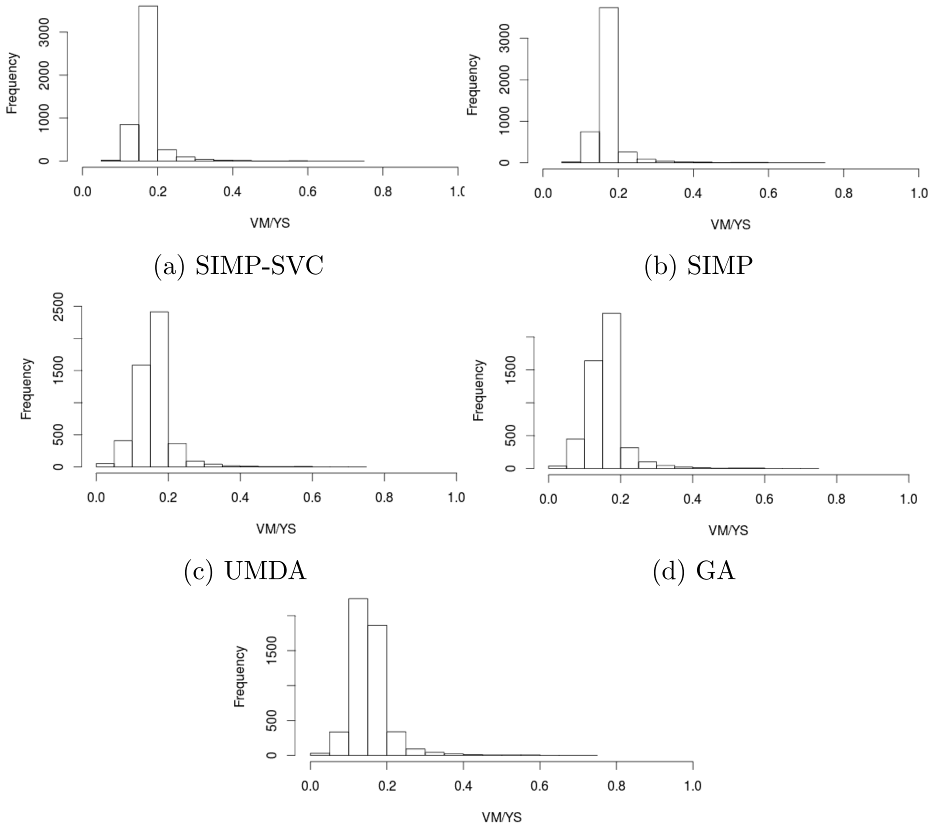
(a) SCLB material (b) SCLB displacements (c) SCLB VonMises

Figure 7.44: SCLB visual results with NSGA-VC

On one hand, SIMP-SVC and SIMP show the most aesthetic shape according to the material distribution, on the other hand, the NSGA-VC shows the best minimization performance according to the compliance reached (shown in the compliance minimization columns of table 7.6).

Note that in table 7.6, SIMP-SVC reaches a better compliance than SIMP, even when SIMP has reached the maximal iterations, in addition, SIMP-SVC converges in 23 iterations, saving at least the 84% of the maximal number. The compliance reached is $2.8305e1$ for SIMP and $2.8286e1$ for SIMP-SVC, which do not represent a significant difference.

Figures 7.45(a), (b), (c), (d) and (e) show the histograms of efficiency for SIMP-SVC, SIMP, UMDA, GA and NSGA-VC, respectively.



(a) SIMP-SVC

(b) SIMP

(c) UMDA

(d) GA

(e) NSGA-VC

Figure 7.45: SCLB: VM/YS elemental histogram

SIMP shows the most efficient performance, according to the histograms, and the average and standard deviation (shown in the compliance minimization columns of table 7.6), due to the average of elemental efficiencies is closer to 1 when using SIMP than any other.

Table 7.6 has the details of the performance of each method. In the volume minimization columns, it is marked with blue the method that shows the best performance for volume minimization, and with red the method that

shows the best performance on efficiency. In the compliance minimization columns it is marked with blue the method that shows the best performance for compliance minimization and with red the method that shows the best performance on efficiency.

	Volume min.		**Compliance min.**				
Data	**SIMP SC**	**NSGA SC**	**SIMP SVC**	**SIMP**	**UMDA**	**GA**	**NSGA VC**
ISF	7.4841e-1	-	7.4841e-1	7.4841e-1	-	-	-
NI	139	500	23	150	500	500	500
IV	1.0e0	1.0e0	1.0e0	1.0e0	1.0e0	1.0e0	1.0e0
FV	2.6367e-1	2.3690e-1	5.0e-1	5.0e-1	5.0e-1	5.0e-1	5.0e-1
FVF	2.6367e-1	2.3690e-1	5.0e-1	5.0e-1	5.0e-1	5.0e-1	5.0e-1
$C(x)$	5.7462e1	6.4536e1	2.8286e1	2.8305e1	2.8067e1	2.7639e1	2.7221e1
ADX	-3.2983e-4	-2.7966e-4	-1.6440e-4	-1.6480e-4	-1.6595e-4	-1.6005e-4	-1.6249e-4
|ADX|	3.2983e-4	2.7966e-4	1.6440e-4	1.6480e-4	1.6595e-4	1.6005e-4	1.6249e-4
ADY	-1.0880e-2	-1.2220e-2	-5.3506e-3	-5.3544e-3	-5.3105e-3	-5.2286e-3	-5.1503e-3
|ADY|	1.0880e-2	1.2220e-2	5.3506e-3	5.3544e-3	5.3105e-3	5.2286e-3	5.1503e-3
A||DXY||	1.0429e-3	1.1463e-3	5.1400e-4	5.1445e-4	5.1120e-4	5.0222e-4	4.9630e-4
SF	9.9849e-1	9.7514e-1	7.2492e-1	7.2553e-1	7.2830e-1	7.2981e-1	7.3156e-1
A(Eff)	3.7022e-1	3.4602e-1	1.6864e-1	1.6906e-1	1.5734e-1	1.5552e-1	1.5501e-1
SD(Eff)	4.4009e-2	1.2499e-1	4.3037e-2	4.2259e-2	5.2652e-2	5.3567e-2	5.2052e-2
MaxVM	2.1966e8	2.1453e8	1.5948e8	1.5931e8	1.6022e8	1.6056e8	1.6094e8

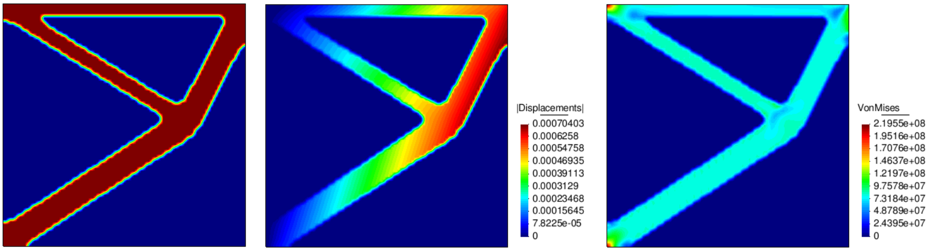
Table 7.6: SCLB execution data

7.6 Short Cantilever with a Load at Top(SCLT)

SCLT test is described in chapter 4 in figure 4.2(c). First, we describe the results obtained for volume minimization and then, for compliance minimization.

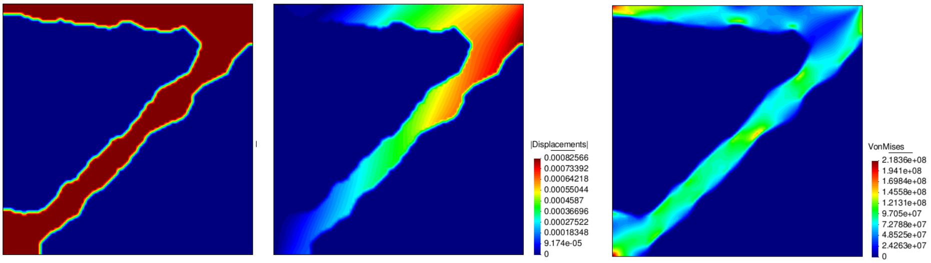
7.6.1 Results for volume minimization methods

Figures 7.46 and 7.47 show results for the SIMP-SC and NSGA-SC respectively: (a) the material distribution on the design domain, (b) the normalized displacements over the structure, and c) the von Mises stresses over the structure.



(a) SCLT material (b) SCLT displacements (c) SCLT VonMises

Figure 7.46: SCLT visual results with SIMP-SC

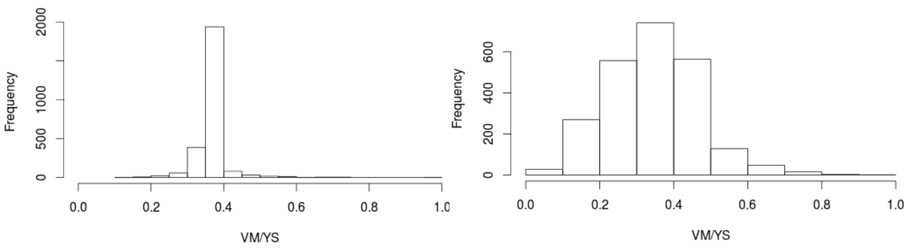


(a) SCLT material (b) SCLT displacements (c) SCLT VonMises

Figure 7.47: SCLT visual results with NSGA-SC

On one hand, SIMP-SC shows the most aesthetic shape according to the material distribution, on the other hand, NSGA-SC shows the best minimization performance according to the volume reached (shown in the volume minimization columns of table 7.7), where the volumes are $2.6367e-1$ and $2.3520e-1$ for SIMP-SC and NSGA-SC respectively.

Figures 7.48(a) and (b) show the histograms of efficiency for SIMP-SC and NSGA-SC respectively.



(a) SIMP-SC

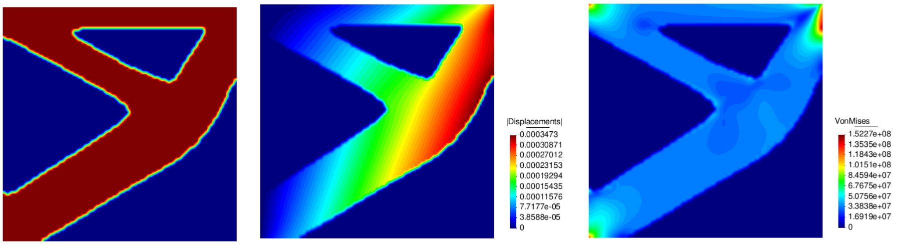
(b) NSGA-SC

Figure 7.48: SCLT: VM/YS elemental histogram

SIMP-SC shows the most efficient performance, according to the histograms and the average and standard deviation (shown in the volume minimization columns of table 7.7), due to the average of elemental efficiencies is closer to 1 when using SIMP-SC than when using NSGA-SC.

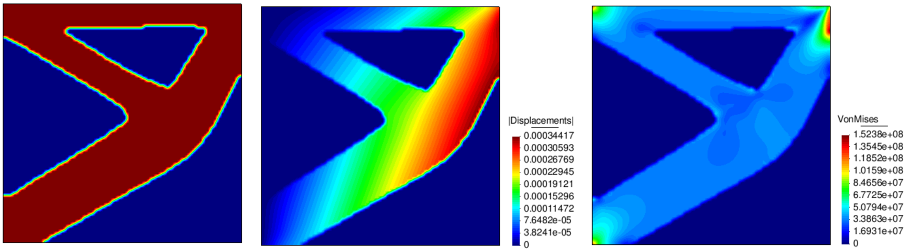
7.6.2 Results for compliance minimization methods

Figures 7.49, 7.50, 7.51, 7.52 and 7.53 show results for the SIMP-SVC, SIMP, GA, UMDA and NSGA-VC, respectively: (a) the material distribution on the design domain, (b) the normalized displacements over the structure, and c) the von Mises stresses over the structure.



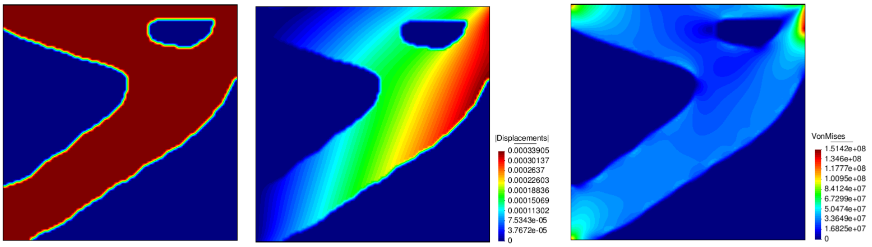
(a) SCLT material (b) SCLT displacements (c) SCLT VonMises

Figure 7.49: SCLT visual results with SIMP-SVC



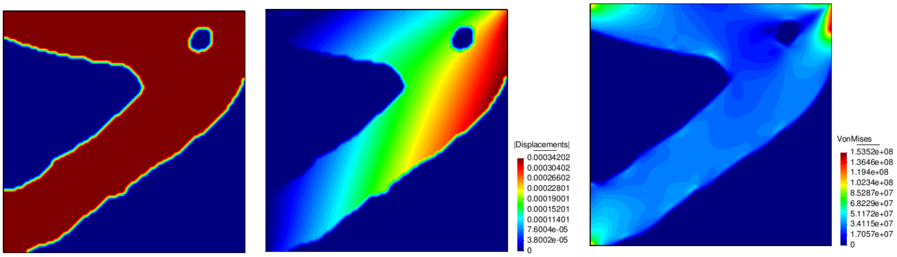
(a) SCLT material (b) SCLT displacements (c) SCLT VonMises

Figure 7.50: SCLT visual results with SIMP



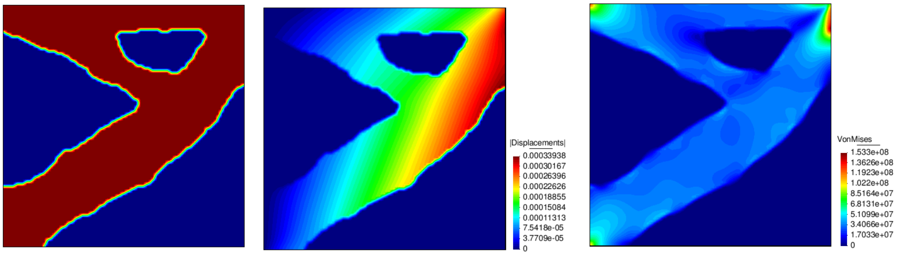
(a) SCLT material (b) SCLT displacements (c) SCLT VonMises

Figure 7.51: SCLT visual results with GA



(a) SCLT material (b) SCLT displacements (c) SCLT VonMises

Figure 7.52: SCLT visual results with UMDA



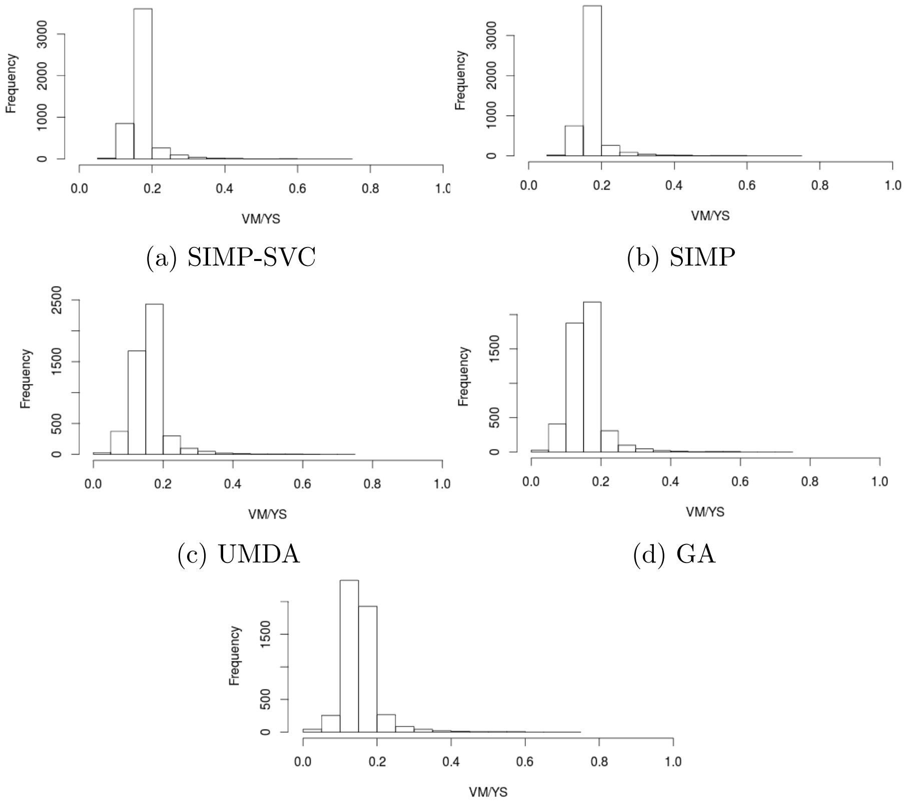
(a) SCLT material (b) SCLT displacements (c) SCLT VonMises

Figure 7.53: SCLT visual results with NSGA-VC

On one hand, SIMP-SVC and SIMP show the most aesthetic shape according to the material distribution, on the other hand, the NSGA-VC shows the best minimization performance according to the compliance reached (shown in the compliance minimization columns of table 7.7).

Note that in table 7.7, SIMP-SVC reaches a better compliance than SIMP, even when SIMP has reached the maximal iterations, in addition, SIMP-SVC converges in 23 iterations, saving at least the 84% of the maximal number. The compliance reached is $2.8304e1$ for SIMP and $2.8285e1$ for SIMP-SVC, which do not represent a significant difference.

Figures 7.54(a), (b), (c), (d) and (e) show the histograms of efficiency for SIMP-SVC, SIMP, UMDA, GA and NSGA-VC, respectively.



(a) SIMP-SVC

(b) SIMP

(c) UMDA

(d) GA

(e) NSGA-VC

Figure 7.54: SCLT: VM/YS elemental histogram

SIMP shows the most efficient performance, according to the histograms, and the average and standard deviation (shown in the compliance minimization columns of table 7.7), due to the average of elemental efficiencies is closer to 1 when using SIMP than any other.

Table 7.7 has the details of the performance of each method. In the volume minimization columns, it is marked with blue the method that shows the best performance for volume minimization, and with red the method that shows the best performance on efficiency. In the compliance minimization columns it is marked with blue the method that shows the best performance for compliance minimization and with red the method that shows the best performance on efficiency.

	Volume min.		**Compliance min.**				
Data	**SIMP SC**	**NSGA SC**	**SIMP SVC**	**SIMP**	**UMDA**	**GA**	**NSGA VC**
ISF	7.4841e-1	-	7.4841e-1	7.4841e-1	-	-	
NI	139	500	23	150	500	500	500
IV	1.0e0	1.0e0	1.0e0	1.0e0	1.0e0	1.0e0	1.0e0
FV	2.6367e-1	2.3520e-1	5.0e-1	5.0e-1	5.0e-1	5.0e-1	5.0e-1
FVF	2.6367e-1	2.3520e-1	5.0e-1	5.0e-1	5.0e-1	5.0e-1	5.0e-1
$C(x)$	5.7451e1	6.3057e1	2.8285e1	2.8304e1	2.7781e1	2.7244e1	2.7168e1
ADX	3.2972e-4	3.1303e-4	1.6439e-4	1.6482e-4	1.5896e-4	1.6223e-4	1.5580e-4
|ADX|	3.2972e-4	3.1303e-4	1.6439e-4	1.6482e-4	1.5896e-4	1.6223e-4	1.5580e-4
ADY	-1.0878e-2	-1.1940e-2	-5.3504e-3	-5.3543e-3	-5.2554e-3	-5.1539e-3	-5.1398e-3
|ADY|	1.0878e-2	1.1940e-2	5.3504e-3	5.3543e-3	5.2554e-3	5.1539e-3	5.1398e-3
A||DXY||	1.0427e0-3	1.1303e-3	5.1398e-4	5.1445e-4	5.0418e-4	4.9647e-4	4.9314e-4
SF	9.9795e-1	9.9256e-1	7.2490e-1	7.2551e-1	7.3191e-1	7.2161e-1	7.3043e-1
A(Eff)	3.7014e-1	3.4263e-1	1.6863e-1	1.6905e-1	1.5656e-1	1.5444e-1	1.5524e-1
SD(Eff)	4.3981e-2	1.2334e-1	4.3037e-2	4.2255e-2	5.1862e-2	5.2562e-2	5.0707e-2
MaxVM	2.1955e8	2.1836e8	1.5947e8	1.5961e8	1.6102e8	1.5875e8	1.6069e8

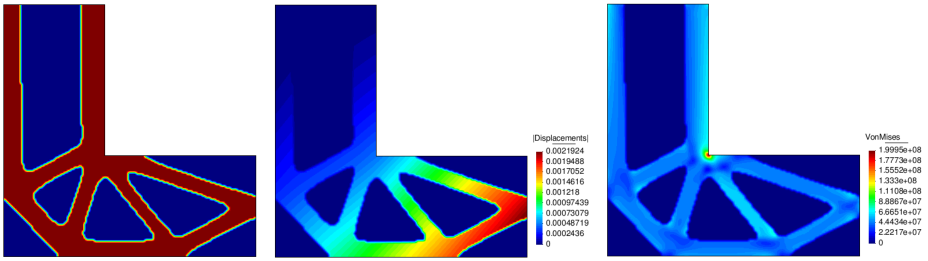
Table 7.7: SCLT execution data

7.7 LShape with a Load at Center(LLC)

LLC test is described in chapter 4 in figure 4.3(a). First, we describe the results obtained for volume minimization and then, for compliance minimization.

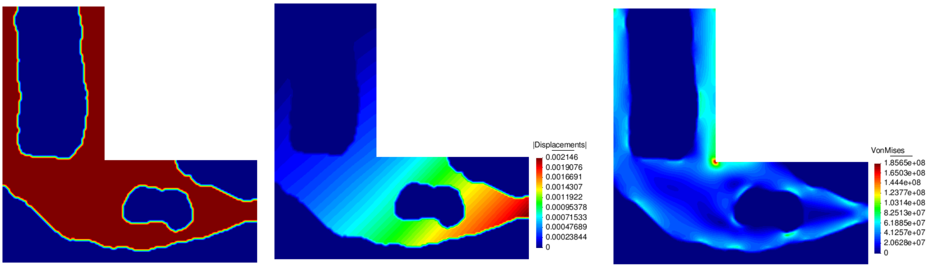
7.7.1 Results for volume minimization methods

Figures 7.55 and 7.56 show results for the SIMP-SC and NSGA-SC respectively: (a) the material distribution on the design domain, (b) the normalized displacements over the structure, and c) the von Mises stresses over the structure.



(a) LLC material (b) LLC displacements (c) LLC VonMises

Figure 7.55: LLC visual results with SIMP-SC



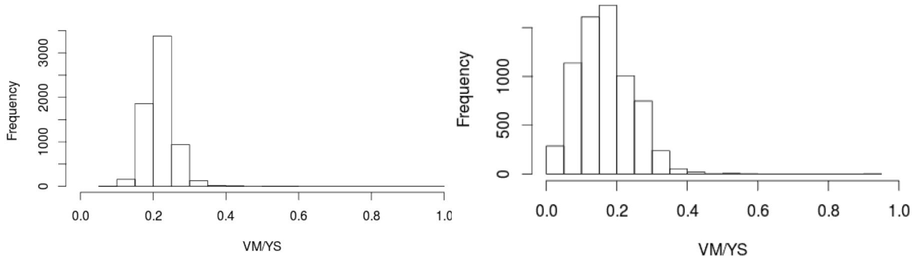
(a) LLC material (b) LLC displacements (c) LLC VonMises

Figure 7.56: LLC visual results with NSGA-SC

SIMP-SC shows the most aesthetic shape according to the material distribution and the best minimization performance according to the volume reached (shown in the volume minimization columns of table 7.8), where the volumes are $2.9125e - 1$ and $3.0417e - 1$ for SIMP-SC and NSGA-SC respectively.

Note that, SIMP-SC reaches a volume/SF of 2.9125e-1/9.9802e-1 and NSGA-SC reaches a volume/SF of 3.0417e-1/9.3353e-1. This relation shows a small change in the volume, but a high difference in the security factor.

Figures 7.57(a) and (b) show the histograms of efficiency for SIMP-SC and NSGA-SC respectively.



(a) SIMP-SC

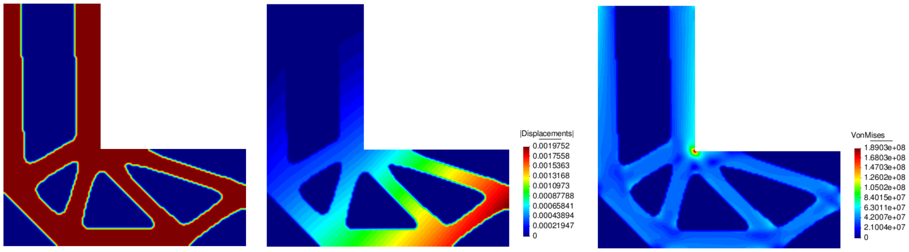
(b) NSGA-SC

Figure 7.57: LLC: VM/YS elemental histogram

SIMP-SC shows the most efficient performance, according to the histograms and the average and standard deviation (shown in the volume minimization columns of table 7.8), due to the average of elemental efficiencies is closer to 1 when using SIMP-SC than when using NSGA-SC.

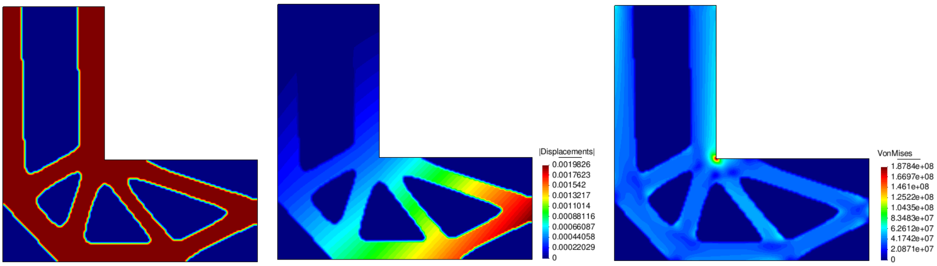
7.7.2 Results for compliance minimization methods

Figures 7.58, 7.59, 7.60, 7.61 and 7.62 show results for the SIMP-SVC, SIMP, GA, UMDA and NSGA-VC, respectively: (a) the material distribution on the design domain, (b) the normalized displacements over the structure, and c) the von Mises stresses over the structure.



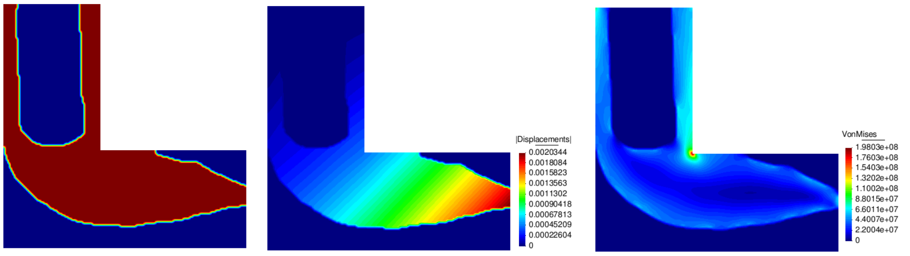
(a) LLC material (b) LLC displacements (c) LLC VonMises

Figure 7.58: LLC visual results with SIMP-SVC



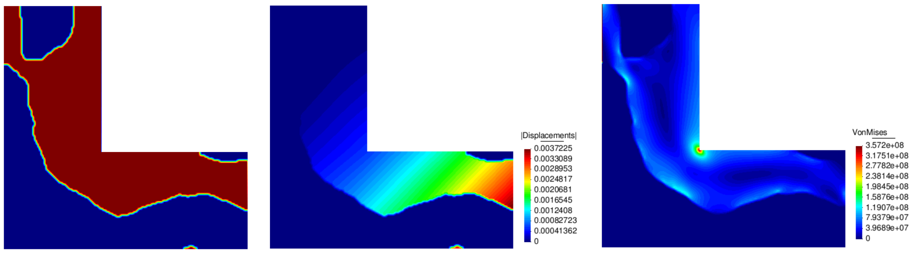
(a) LLC material (b) LLC displacements (c) LLC VonMises

Figure 7.59: LLC visual results with SIMP



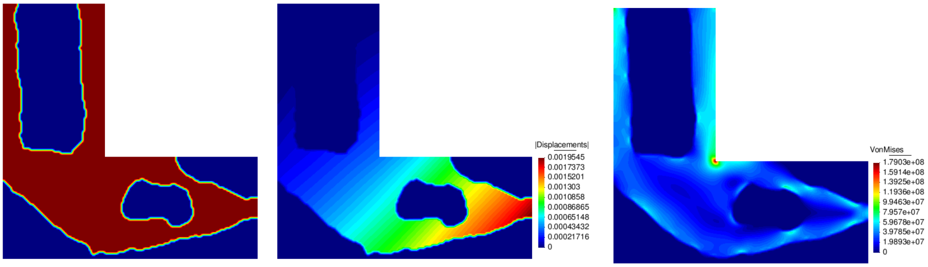
(a) LLC material (b) LLC displacements (c) LLC VonMises

Figure 7.60: LLC visual results with GA



(a) LLC material (b) LLC displacements (c) LLC VonMises

Figure 7.61: LLC visual results with UMDA



(a) LLC material (b) LLC displacements (c) LLC VonMises

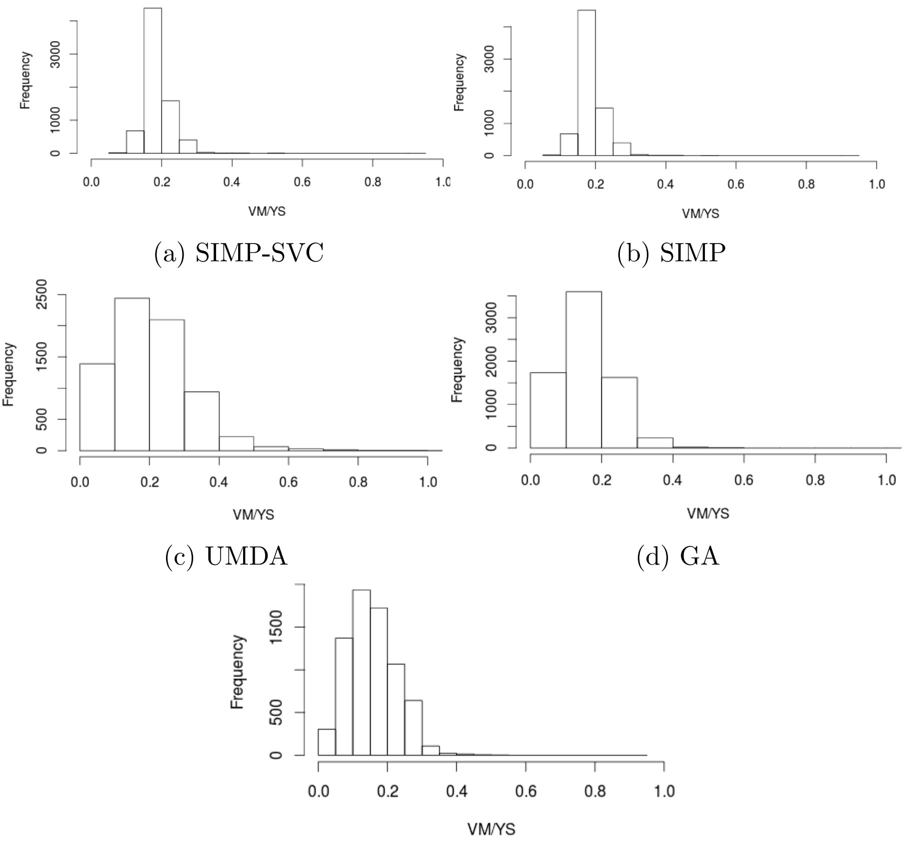
Figure 7.62: LLC visual results with NSGA-VC

SIMP shows the most aesthetic shape according to the material distribution and the best minimization performance according to the compliance reached (shown in the compliance minimization columns of table 7.8).

Note that SIMP does not converge within the maximal iterations, and SIMP-SVC does it in 30, saving at least the 80% of the iterations. The compliance reached is $2.0794e1$ for SIMP and $2.0906e1$ for SIMP-SVC, which do not represent a significant difference.

Note that NSGA-VC reaches a compliance of $C(x) = 2.0805e1$, which is better than SIMP-SVC and very similar to SIMP, nevertheless its material distribution shows not an aesthetic shape.

Figures 7.63(a), (b), (c), (d) and (e) show the histograms of efficiency for SIMP-SVC, SIMP, UMDA, GA and NSGA-VC, respectively.



(a) SIMP-SVC

(b) SIMP

(c) UMDA

(d) GA

(e) NSGA-VC

Figure 7.63: LLC: VM/YS elemental histogram

SIMP-SVC shows the most efficient performance, according to the histograms, and the average and standard deviation (shown in the compliance minimization columns of table 7.8), due to the average of elemental efficiencies is closer to 1 when using SIMP-SVC than any other.

Table 7.8 has the details of the performance of each method. In the volume minimization columns, it is marked with blue the method that shows the best performance for volume minimization, and with red the method that

shows the best performance on efficiency. In the compliance minimization columns it is marked with blue the method that shows the best performance for compliance minimization and with red the method that shows the best performance on efficiency.

	Volume min.		**Compliance min.**				
Data	**SIMP SC**	**NSGA SC**	**SIMP SVC**	**SIMP**	**UMDA**	**GA**	**NSGA VC**
ISF	7.5569e-1	-	7.5569e-1	7.5569e-1	-	-	-
NI	170	500	30	150	500	500	500
IV	6.4e-1	6.4e-1	6.4e-1	6.4e-1	6.4e-1	6.4e-1	6.4e-1
FV	2.9125e-1	3.0417e-1	3.2e-1	3.2e-1	3.2e-1	3.2e-1	3.2e-1
FVF	4.5508e-1	4.7527e-1	5.0e-1	5.0e-1	5.0e-1	5.0e-1	
$C(x)$	2.3231e1	2.2499e1	2.0906e1	2.0794e1	3.8191e1	2.1641e1	2.0805e1
ADX	-5.8670e-4	-5.6973e-4	-5.1391e-4	-5.1657e-4	-1.0323e-3	-5.4948e-4	-4.9724e-4
|ADX|	5.8670e-4	5.6973e-4	5.1391e-4	5.1657e-4	1.0323e-3	5.4948e-4	4.9724e-4
ADY	-1.0810e-2	-1.0469e-2	-9.7257e-3	-9.6748e-3	-1.7769e-2	-1.0063e-2	-9.6798e-3
|ADY|	1.0810e-2	1.0469e-2	9.7257e-3	9.6748e-3	1.7769e-2	1.0063e-2	9.6798e-3
A||DXY||	1.6459e-3	1.6007e-3	1.4816e-3	1.4757e-3	2.7410e-3	1.5394e-3	1.4698e-3
SF	9.9802e-1	9.3353e-1	9.4391e-1	9.3930e-1	1.7872e0	1.0029e0	9.0179e-1
A(Eff)	2.1641e-1	1.6680e-1	1.8954e-1	1.8868e-1	2.0392e-1	1.5910e-1	1.5673e-1
SD(Eff	4.2650e-2	7.7830e-2	3.9668e-2	3.9945e-2	1.1849e-1	7.3881e-2	7.2532e-2
MaxVM	2.1956e8	2.0537e8	2.0766e8	2.0664e8	3.9320e8	2.2065e8	1.9839e8

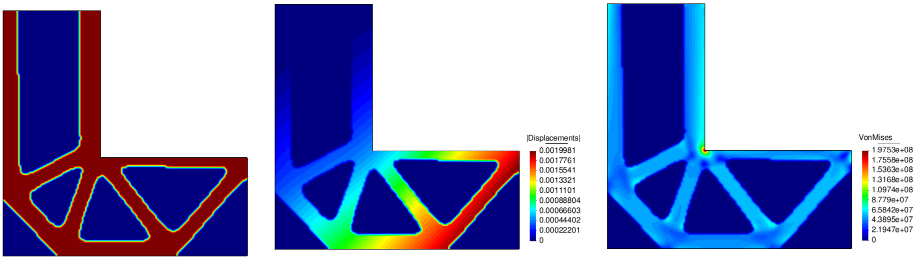
Table 7.8: LLC execution data

7.8 LShape with a Load at Top(LLT)

LLC test is described in chapter 4 in figure 4.3(b). First, we describe the results obtained for volume minimization and then, for compliance minimization.

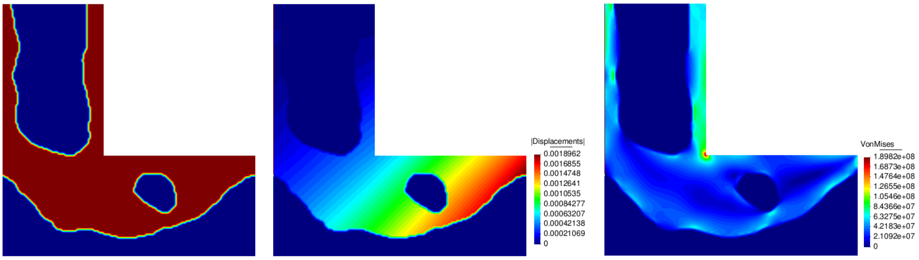
7.8.1 Results for volume minimization methods

Figures 7.64 and 7.65 show results for the SIMP-SC and NSGA-SC respectively: (a) the material distribution on the design domain, (b) the normalized displacements over the structure, and c) the von Mises stresses over the structure.



(a) LLT material (b) LLT displacements (c) LLT VonMises

Figure 7.64: LLT visual results with SIMP-SC

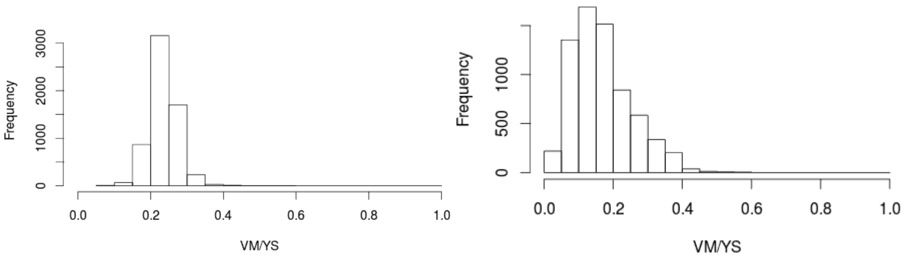


(a) LLT material (b) LLT displacements (c) LLT VonMises

Figure 7.65: LLC visual results with NSGA-SC

SIMP-SC shows the most aesthetic shape according to the material distribution and the best minimization performance according to the volume reached (shown in the volume minimization columns of table 7.9), where the volumes are $2.7246e-1$ and $3.0271e-1$ for SIMP-SC and NSGA-SC respectively.

Figures 7.66(a) and (b) show the histograms of efficiency for SIMP-SC and NSGA-SC respectively.



(a) SIMP-SC

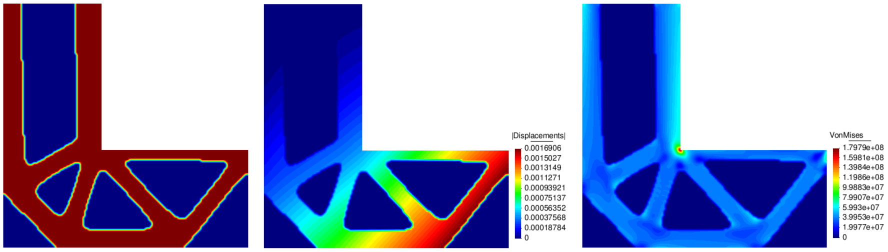
(b) NSGA-SC

Figure 7.66: LLT: VM/YS elemental histogram

SIMP-SC shows the most efficient performance, according to the histograms and the average and standard deviation (shown in the volume minimization columns of table 7.9), due to the average of elemental efficiencies is closer to 1 when using SIMP-SC than when using NSGA-SC.

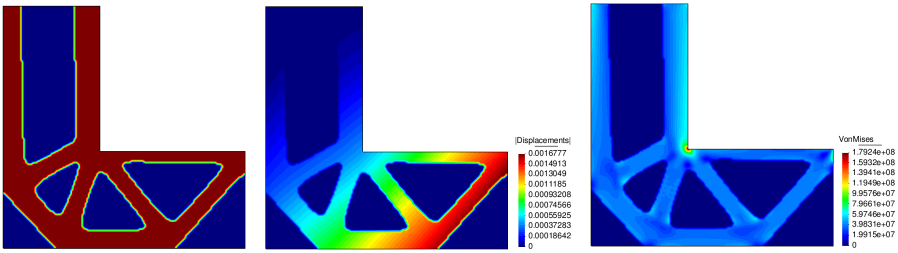
7.8.2 Results for compliance minimization methods

Figures 7.67, 7.68, 7.69, 7.70 and 7.71 show results for the SIMP-SVC, SIMP, GA, UMDA and NSGA-VC, respectively: (a) the material distribution on the design domain, (b) the normalized displacements over the structure, and c) the von Mises stresses over the structure.



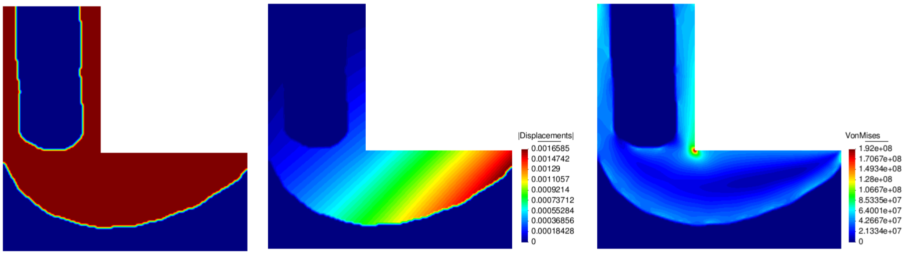
(a) LLT material (b) LLT displacements (c) LLT VonMises

Figure 7.67: LLT visual results with SIMP-SVC



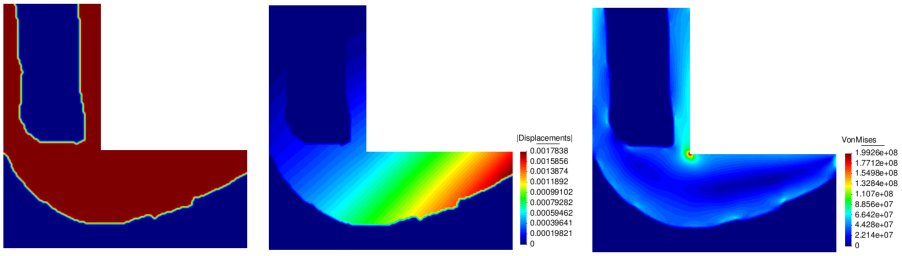
(a) LLT material (b) LLT displacements (c) LLT VonMises

Figure 7.68: LLT visual results with SIMP



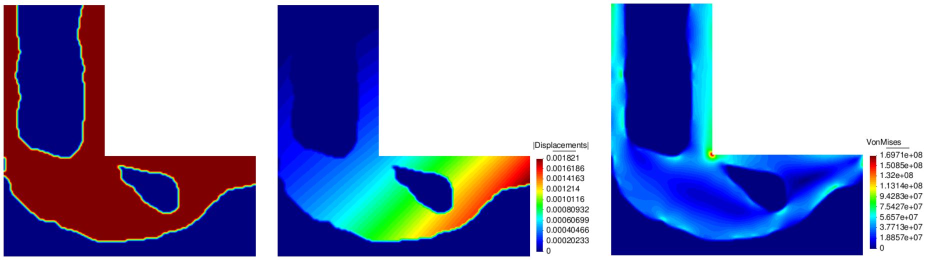
(a) LLT material (b) LLT displacements (c) LLT VonMises

Figure 7.69: LLT visual results with GA



(a) LLT material (b) LLT displacements (c) LLT VonMises

Figure 7.70: LLT visual results with UMDA



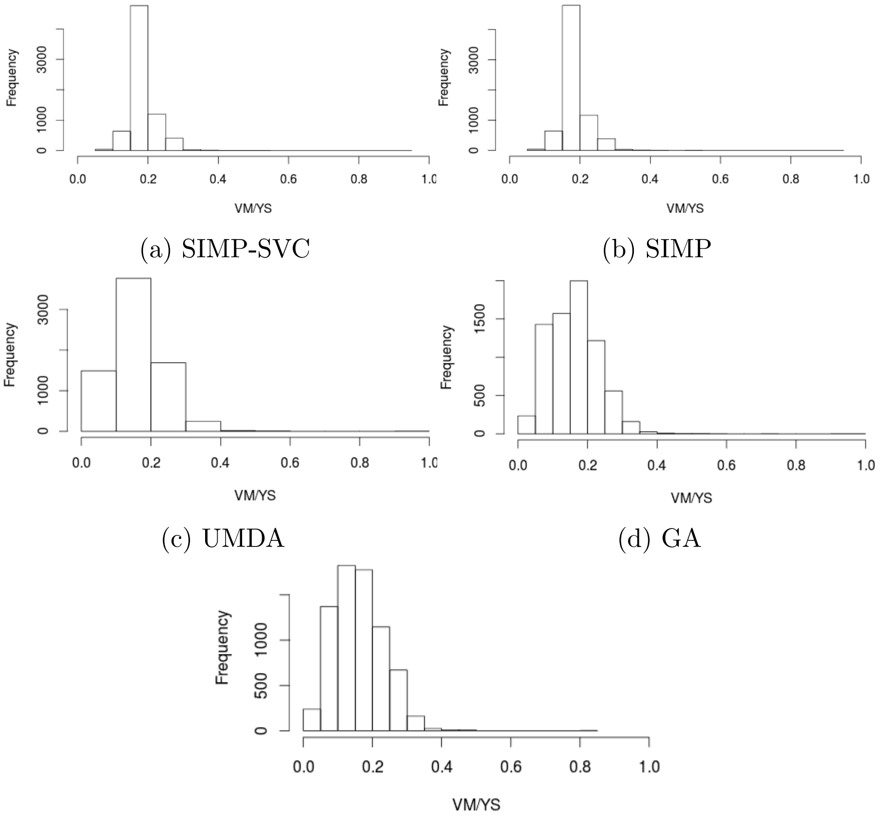
(a) LLT material (b) LLT displacements (c) LLT VonMises

Figure 7.71: LLT visual results with NSGA-VC

SIMP-SVC shows the most aesthetic shape according to the material distribution and the best minimization performance according to the compliance reached (shown in the compliance minimization columns of table 7.9), even with a lower number of iterations than SIMP.

Note that, SIMP does not converge within the maximal iterations, and SIMP-SVC does it in 28, saving at least the 81.33% of the iterations.

Figures 7.72(a), (b), (c), (d) and (e) show the histograms of efficiency for SIMP-SVC, SIMP, UMDA, GA and NSGA-VC, respectively.



(a) SIMP-SVC

(b) SIMP

(c) UMDA

(d) GA

(e) NSGA-VC

Figure 7.72: LLT: VM/YS elemental histogram

SIMP-SVC shows the most efficient performance, according to the histograms, and the average and standard deviation (shown in the compliance minimization columns of table 7.9), due to the average of elemental efficiencies is closer to 1 when using SIMP-SVC than any other.

Table 7.9 has the details of the performance of each method. In the volume minimization columns, it is marked with blue the method that shows the best performance for volume minimization, and with red the method that

shows the best performance on efficiency. In the compliance minimization columns it is marked with blue the method that shows the best performance for compliance minimization and with red the method that shows the best performance on efficiency.

	Volume min.		**Compliance min.**				
Data	**SIMP SC**	**NSGA SC**	**SIMP SVC**	**SIMP**	**UMDA**	**GA**	**NSGA VC**
ISF	7.5550e-1	-	7.5550e-1	7.5550e-1	-	-	-
NI	183	500	28	150	500	500	500
IV	6.4e-1	6.4e-1	6.4e-1	6.4e-1	6.4e-1	6.4e-1	6.4e-1
FV	2.7246e-1	3.0271e-1	3.2e-1	3.2e-1	3.2e-1	3.2e-1	3.2e-1
FVF	4.25724e-1	4.7298e-1	5.0e-1	5.0e-1	5.0e-1	5.0e-1	5.0e-1
$C(x)$	2.5389e1	2.3852e1	2.0191e1	2.1044e1	2.2254e1	2.1912e1	2.1542e1
ADX	-1.8755e-4	-1.7816e-4	-1.6784e-4	-1.6760e-4	-1.5077e-4	-1.1424e-4	-1.9548e-4
|ADX|	1.8755e-4	1.7816e-4	1.6784e-4	1.6760e-4	1.5077e-4	1.1424e-4	1.9548e-4
ADY	-1.1819e-2	-1.1094e-2	-9.8113e-3	-9.7886e-3	-1.0350e-2	-1.0190e-2	-1.0020e-2
|ADY|	1.1819e-2	1.1094e-2	9.8113e-3	9.7886e-3	1.0350e-2	1.0190e-2	1.0020e-2
A||DXY||	1.6992e-3	1.5955e-3	1.4119e-3	1.4087e-3	1.4868e-3	1.4608e-3	1.4453e-3
SF	9.9995e-1	9.7756e-1	9.0702e-1	9.0651e-1	9.9858e-1	9.6628e-1	8.4851e-1
A(Eff)	2.3528e-1	1.6896e-1	1.8814e-1	1.8762e-1	1.6328e-1	1.6156e-1	1.5991e-1
SD(Eff)	4.1240e-2	8.7014e-2	3.8693e-2	3.8581e-2	7.1339e-2	7.1437e-2	7.2082e-2
MaxVM	2.1998e8	2.1506e8	1.9954e8	1.9943e8	2.1968e8	2.1258e8	1.8667e8

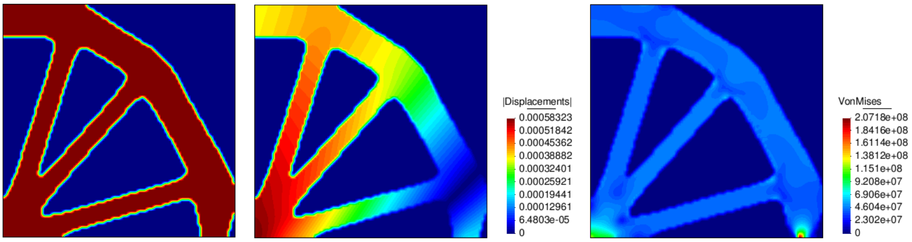
Table 7.9: LLT execution data

7.9 One Load Michell(OLM)

OLM test is described in chapter 4 in figure 4.4(a). First, we describe the results obtained for volume minimization and then, for compliance minimization.

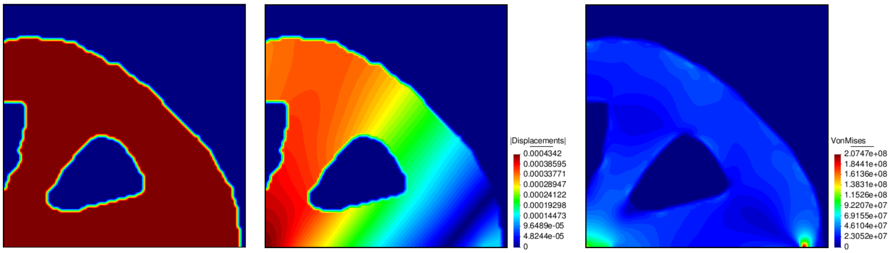
7.9.1 Results for volume minimization methods

Figures 7.73 and 7.74 show results for the SIMP-SC and NSGA-SC respectively: (a) the material distribution on the design domain, (b) the normalized displacements over the structure, and c) the von Mises stresses over the structure.



(a) OLM material (b) OLM displacements (c) OLM VonMises

Figure 7.73: OLM visual results with SIMP-SC

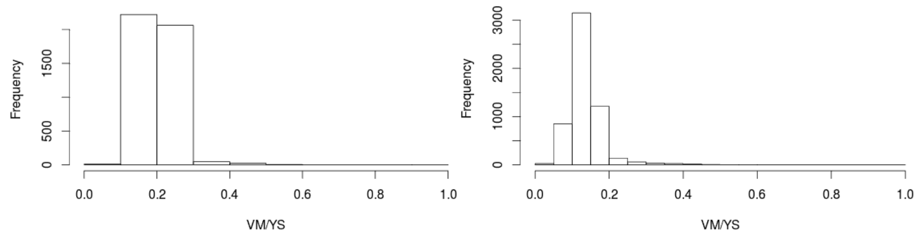


(a) OLM material (b) OLM displacements (c) OLM VonMises

Figure 7.74: OLM visual results with NSGA-SC

SIMP-SC shows the most aesthetic shape according to the material distribution and the best minimization performance according to the volume reached (shown in the volume minimization columns of table 7.10), where the volumes are $4.4531e-1$ and $5.5080e-1$ for SIMP-SC and NSGA-SC respectively.

Figures 7.75(a) and (b) show the histograms of efficiency for SIMP-SC and NSGA-SC respectively.



(a) SIMP-SC

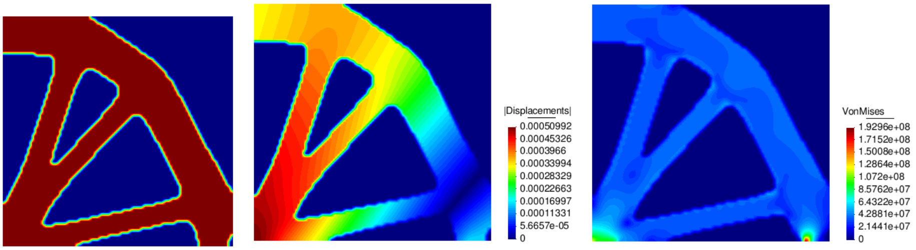
(b) NSGA-SC

Figure 7.75: OLM: VM/YS elemental histogram

SIMP-SC shows the most efficient performance, according to the histograms and the average and standard deviation (shown in the volume minimization columns of table 7.10), due to the average of elemental efficiencies is closer to 1 when using SIMP-SC than when using NSGA-SC.

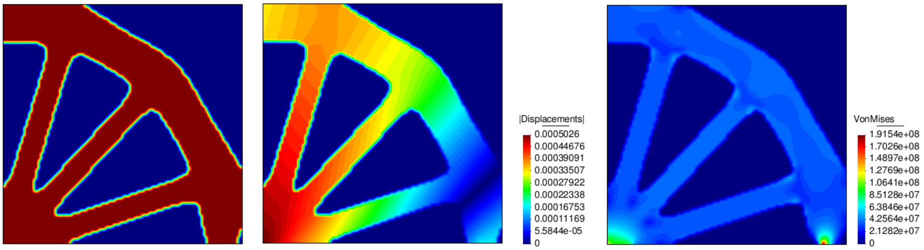
7.9.2 Results for compliance minimization methods

Figures 7.76, 7.77, 7.78, 7.79 and 7.80 show results for the SIMP-SVC, SIMP, GA, UMDA and NSGA-VC, respectively: (a) the material distribution on the design domain, (b) the normalized displacements over the structure, and c) the von Mises stresses over the structure.



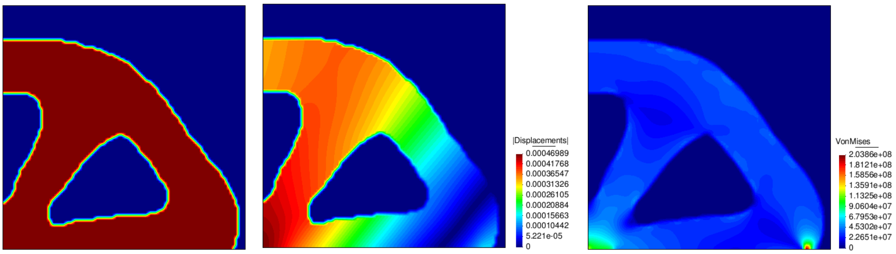
(a) OLM material (b) OLM displacements (c) OLM VonMises

Figure 7.76: OLM visual results with SIMP-SVC



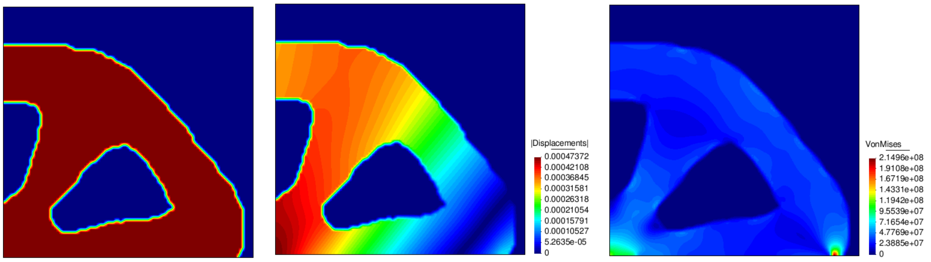
(a) OLM material (b) OLM displacements (c) OLM VonMises

Figure 7.77: OLM visual results with SIMP



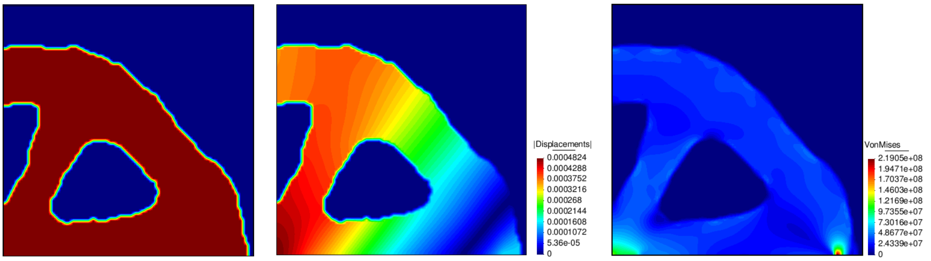
(a) OLM material (b) OLM displacements (c) OLM VonMises

Figure 7.78: OLM visual results with GA



(a) OLM material (b) OLM displacements (c) OLM VonMises

Figure 7.79: OLM visual results with UMDA



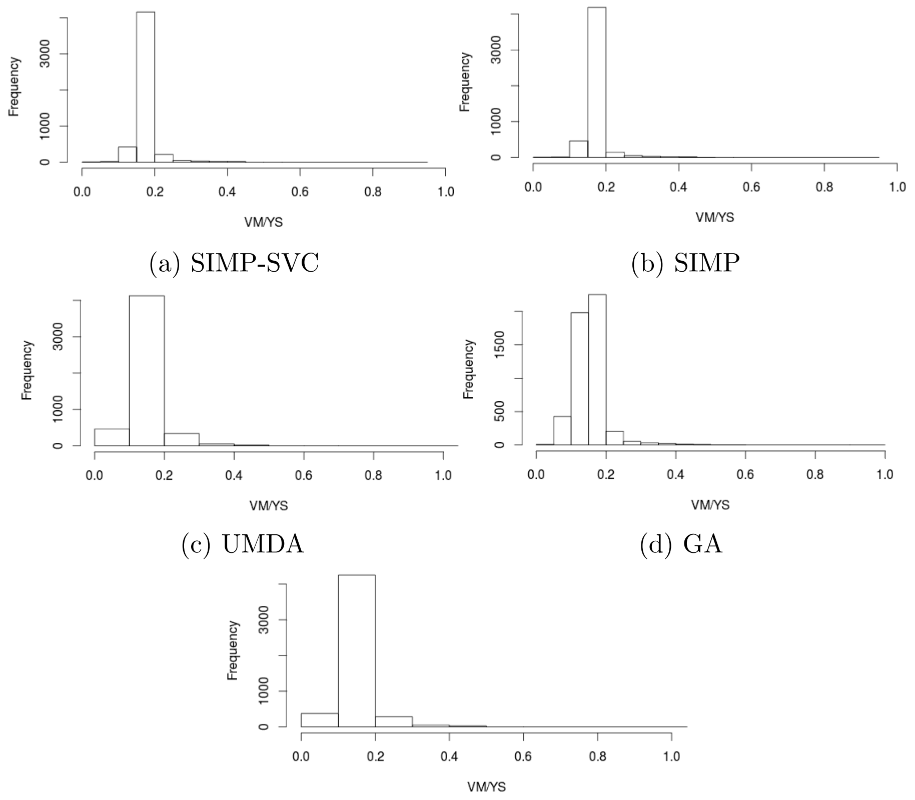
(a) OLM material (b) OLM displacements (c) OLM VonMises

Figure 7.80: OLM visual results with NSGA-VC

On one hand, SIMP-SVC and SIMP show the most aesthetic shape according to the material distribution, on the other hand, the GA shows the best minimization performance according to the compliance reached (shown in the compliance minimization columns of table 7.10).

Note that SIMP does not converge within the maximal iterations, and SIMP-SVC does it in 30, saving at least the 80% of the iterations. The compliance reached is $2.7777e1$ for SIMP and $2.8184e1$ for SIMP-SVC, which do not represent a significant difference.

Figures 7.81(a), (b), (c), (d) and (e) show the histograms of efficiency for SIMP-SVC, SIMP, UMDA, GA and NSGA-VC, respectively.



(a) SIMP-SVC

(b) SIMP

(c) UMDA

(d) GA

(e) NSGA-VC

Figure 7.81: OLM: VM/YS elemental histogram

SIMP-SVC shows the most efficient performance, according to the histograms, and the average and standard deviation (shown in the compliance minimization columns of table 7.10), due to the average of elemental efficiencies is closer to 1 when using SIMP-SVC than any other.

Table 7.10 has the details of the performance of each method. In the volume minimization columns, it is marked with blue the method that shows the best performance for volume minimization, and with red the method that shows the best performance on efficiency. In the compliance mini-

mization columns it is marked with blue the method that shows the best performance for compliance minimization and with red the method that shows the best performance on efficiency.

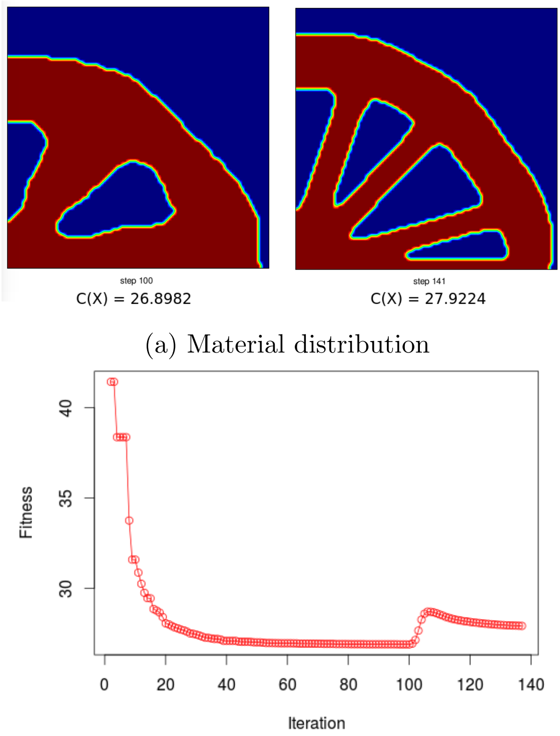
	Volume min.		Compliance min.				
Data	**SIMP SC**	**NSGA SC**	**SIMP SVC**	**SIMP**	**UMDA**	**GA**	**NSGA VC**
ISF	7.5248e-1	-	7.5248e-1	7.5248e-1	-	-	-
NI	138	500	30	150	500	500	500
IV	1.0e0	1.0e0	1.0e0	1.0e0	1.0e0	1.0e0	1.0e0
FV	4.4531e-1	5.5080e-1	5.0e-1	5.0e-1	5.0e-1	5.0e-1	5.0e-1
FVF	4.4531e-1	5.5080e-1	5.0e-1	5.0e-1	5.0e-1	5.0e-1	5.0e-1
$C(x)$	3.2247e1	2.4009e1	2.8184e1	2.7777e1	2.6185e1	2.5972e1	2.6684e1
ADX	1.9085e-5	1.7469e-5	1.7803e-5	1.7974e-5	1.8856e-5	1.8363e-5	1.8796e-5
|ADX|	1.9085e-5	1.7469e-5	1.7803e-5	1.7974e-5	1.8856e-5	1.8363e-5	1.8796e-5
ADY	-6.3126e-3	-4.6949e-3	-5.5149e-3	-5.4353e-3	-5.1226e-3	-5.0812e-3	-5.2204e-3
|ADY|	6.3126e-3	4.6949e-3	5.5149e-3	5.4353e-3	5.1226e-3	5.0812e-3	5.2204e-3
A||DXY||	5.7431e-4	4.2731e-4	5.0180e-4	4.9457e-4	4.6623e-4	4.6244e-4	4.7511e-4
SF	9.9963e-1	9.9694e-1	9.3160e-1	9.2516e-1	1.0324e0	9.7989e-1	1.0508e0
A(Eff)	2.0239e-1	1.3719e-1	1.7432e-1	1.7361e-1	1.5120e-1	1.5170e-1	1.5271e-1
SD(Eff)	3.7517e-2	4.8971e-2	3.6663e-2	3.6530e-2	5.1058e-2	4.7925e-2	5.0999e-2
MaxVM	2.1992e8	2.1926e8	2.0495e8	2.0353e8	2.2714e8	2.1557e8	2.3118e8

Table 7.10: OLM execution data

For this tests, an extra experiment is executed. It is an hybrid (or memetic) algorithm that mixes global and local optimization, this means, the solution from an evolutionary algorithm is the input for a local optimization algorithm: SIMP-SVC. Based on tests executed, SIMP-SVC delivers aesthetic results, and the GA delivers better results in objective function. The purpose of this experiment is to get results mixing the aesthetic and the objective function from SIMP-SVC and GA respectively.

Figure 7.82(a) shows (in the left side) the result obtained with the genetic algorithm until 100 iterations. Note that the compliance reached is $C(x) = 26.8982$. Then, starting from this solution, SIMP-SVC is executed. The convergence for SIMP-SVC is at 41 iterations and the compliance reached is $C(x) = 27.9224$.

Figure 7.82(b) shows a graphic with the compliance evolution of the hybrid algorithm. Note that, starting from the iteration 100 (end of the evolutionary algorithm, and beginning of the SIMP-SVC), the compliance increases, thus, it gets worse.



(a) Material distribution

(b) Graphic of compliance behavior

Figure 7.82: Hybrid algorithm results

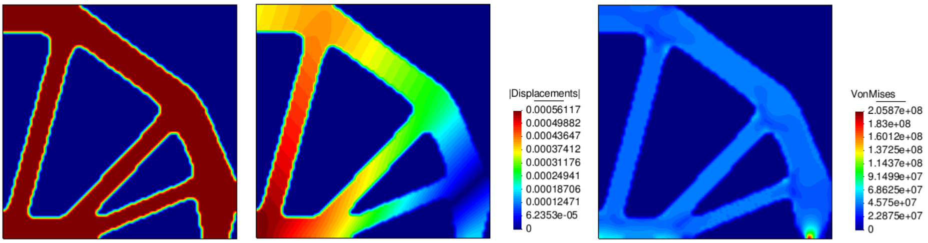
The reason of this behavior is due to SIMP-SVC is updating the design variables based on the heuristics(filtering mainly), not in the mathematical-optimization approach. In other words, SIMP-SVC is updating the design variables to reach an aesthetic shape, no matter the objective if the objective function is increasing.

7.10 Two Equal Loads Michell(TELM)

TELM test is described in chapter 4 in figure 4.4(b). First, we describe the results obtained for volume minimization and then, for compliance minimization.

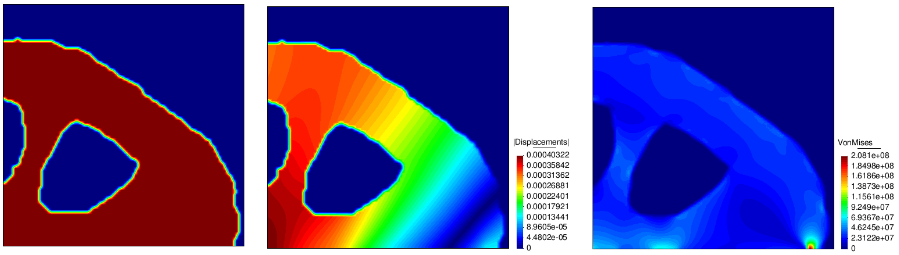
7.10.1 Results for volume minimization methods

Figures 7.83 and 7.84 show results for the SIMP-SC and NSGA-SC respectively: (a) the material distribution on the design domain, (b) the normalized displacements over the structure, and c) the von Mises stresses over the structure.



(a) TELM material (b) TELM displacements (c) TELM VonMises

Figure 7.83: TELM visual results with SIMP-SC

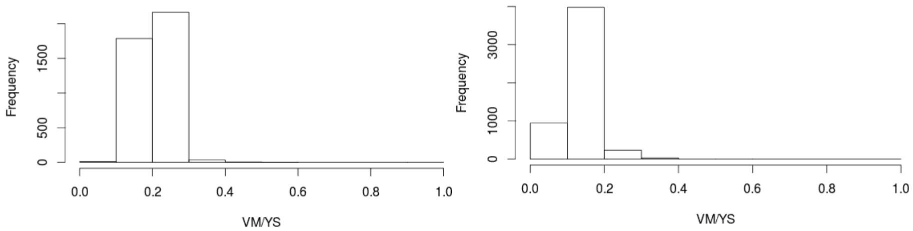


(a) TELM material (b) TELM displacements (c) TELM VonMises

Figure 7.84: TELM visual results with NSGA-SC

SIMP-SC shows the most aesthetic shape according to the material distribution and the best minimization performance according to the volume reached (shown in the volume minimization columns of table 7.11), where the volumes are $4.0820e-1$ and $5.1870e-1$ for SIMP-SC and NSGA-SC respectively.

Figures 7.85(a) and (b) show the histograms of efficiency for SIMP-SC and NSGA-SC respectively.



(a) SIMP-SC

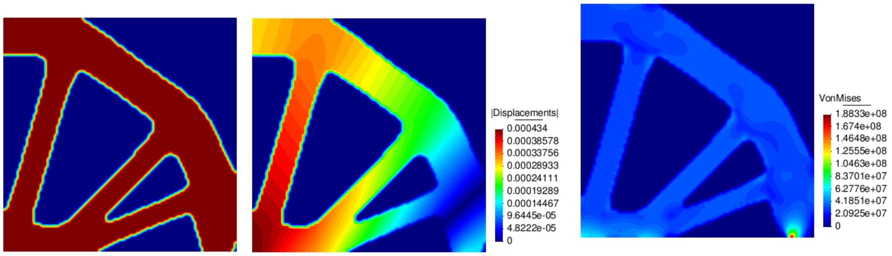
(b) NSGA-SC

Figure 7.85: TELM: VM/YS elemental histogram

SIMP-SC shows the most efficient performance, according to the histograms and the average and standard deviation (shown in the volume minimization columns of table 7.11), due to the average of elemental efficiencies is closer to 1 when using SIMP-SC than when using NSGA-SC.

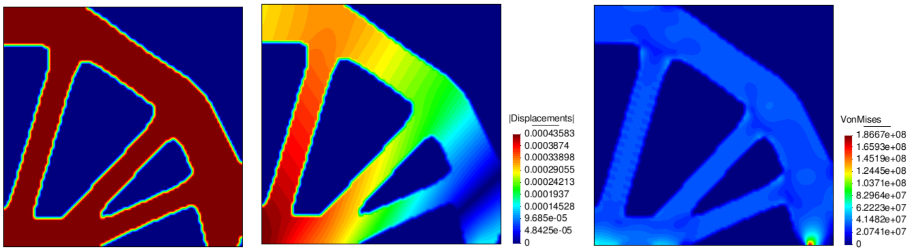
7.10.2 Results for compliance minimization methods

Figures 7.86, 7.87, 7.88, 7.89 and 7.90 show results for the SIMP-SVC, SIMP, GA, UMDA and NSGA-VC, respectively: (a) the material distribution on the design domain, (b) the normalized displacements over the structure, and c) the von Mises stresses over the structure.



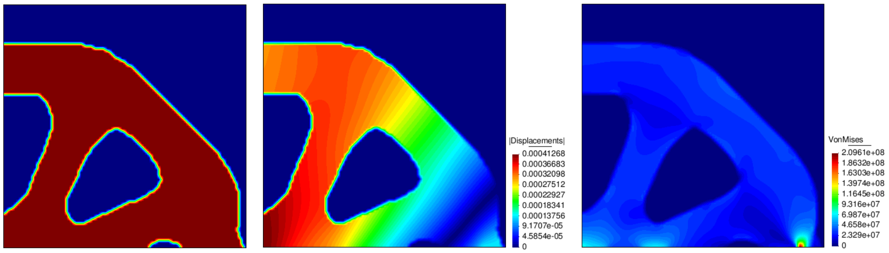
(a) TELM material (b) TELM displacements (c) TELM VonMises

Figure 7.86: TELM visual results with SIMP-SVC



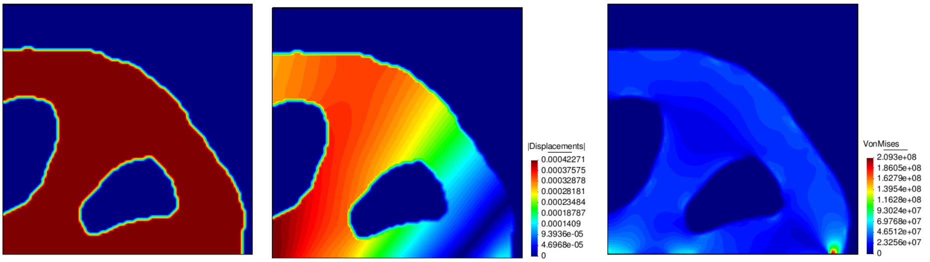
(a) TELM material (b) TELM displacements (c) TELM VonMises

Figure 7.87: TELM visual results with SIMP



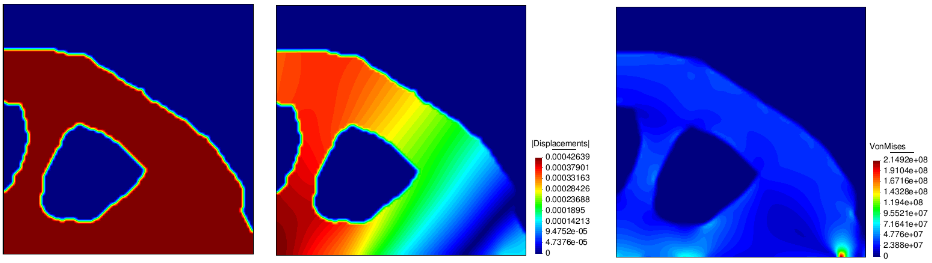
(a) TELM material (b) TELM displacements (c) TELM VonMises

Figure 7.88: TELM visual results with GA



(a) TELM material (b) TELM displacements (c) TELM VonMises

Figure 7.89: TELM visual results with UMDA



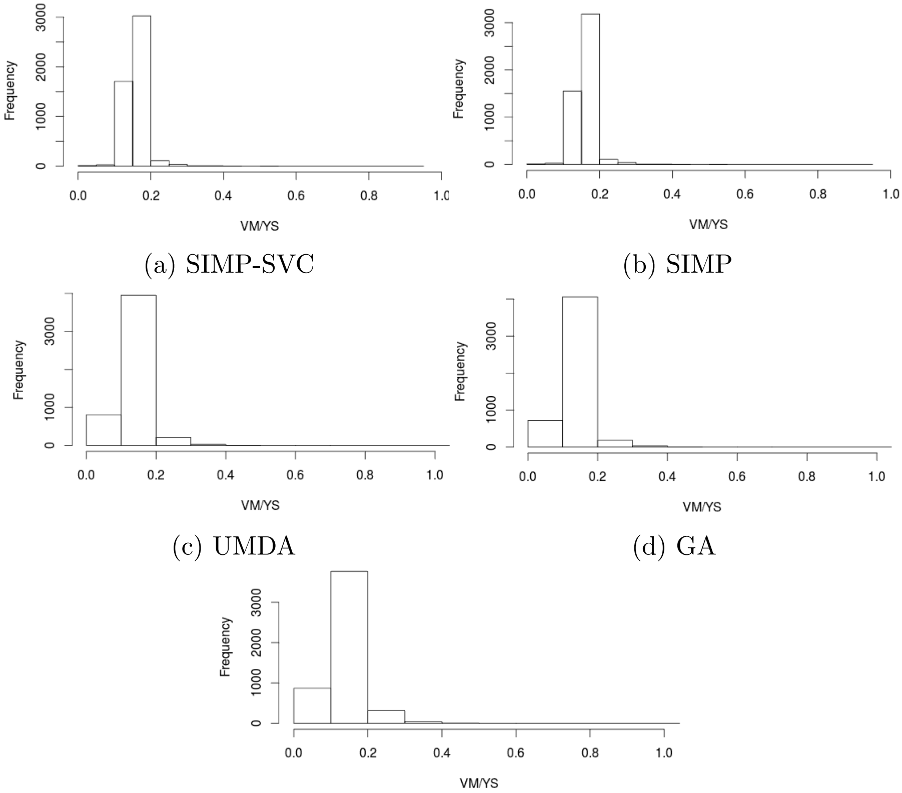
(a) TELM material (b) TELM displacements (c) TELM VonMises

Figure 7.90: TELM visual results with NSGA-VC

On one hand, SIMP-SVC and SIMP show the most aesthetic shape according to the material distribution, on the other hand, the GA shows the best minimization performance according to the compliance reached (shown in the compliance minimization columns of table 7.11).

Note that in table 7.11, SIMP-SVC reaches a better compliance than SIMP, even when SIMP has reached the maximal iterations, in addition, SIMP-SVC converges in 29 iterations, saving at least the 80.66% of the maximal number.

Figures 7.91(a), (b), (c), (d) and (e) show the histograms of efficiency for SIMP-SVC, SIMP, UMDA, GA and NSGA-VC, respectively.



(a) SIMP-SVC

(b) SIMP

(c) UMDA

(d) GA

(e) NSGA-VC

Figure 7.91: TELM: VM/YS elemental histogram

SIMP shows the most efficient performance, according to the histograms, and the average and standard deviation (shown in the compliance minimization columns of table 7.11), due to the average of elemental efficiencies is closer to 1 when using SIMP than any other.

Table 7.11 has the details of the performance of each method. In the volume minimization columns, it is marked with blue the method that shows the best performance for volume minimization, and with red the method that shows the best performance on efficiency. In the compliance mini-

mization columns it is marked with blue the method that shows the best performance for compliance minimization and with red the method that shows the best performance on efficiency.

	Volume min.		**Compliance min.**				
Data	**SIMP SC**	**NSGA SC**	**SIMP SVC**	**SIMP**	**UMDA**	**GA**	**NSGA VC**
ISF	7.4867e-1	-	7.4867e-1	7.4867e-1	-	-	-
NI	156	500	29	150	500	500	500
IV	1.0e0	1.0e0	1.0e0	1.0e0	1.0e0	1.0e0	1.0e0
FV	4.0820e-1	5.1870e-1	5.0e-1	5.0e-1	5.0e-1	5.0e-1	5.0e-1
FVF	4.0820e-1	5.1870e-1	5.0e-1	5.0e-1	5.0e-1	5.0e-1	5.0e-1
$C(x)$	2.9077e1	2.1260e1	2.2515e1	2.2538e1	2.1912e1	2.1469e1	2.2294e1
ADX	4.6020e-5	4.0206e-5	3.8176e-5	3.8311e-5	3.9072e-5	4.0445e-5	4.3158e-5
|ADX|	4.6020e-5	4.0206e-5	3.8176e-5	3.8311e-5	3.9072e-5	4.0445e-5	4.3158e-5
ADY	-1.1384e-2	-8.3140e-3	-8.8085e-3	-8.8171e-3	-8.5697e-3	-8.3973e-3	-8.7200e-3
|ADY|	1.1384e-2	8.3140e-3	8.8085e-3	8.8171e-3	8.5697e-3	8.3973e-3	8.7200e-3
A||DXY||	5.2082e-4	3.8145e-4	4.0333e-4	4.0367e-4	3.9254e-4	3.8506e-4	4.0029e-4
SF	9.9362e-1	9.9988e-1	9.0963e-1	9.0184e-1	1.0059e0	1.0082e0	1.0330e0
A(Eff)	2.0362e-1	1.3365e-1	1.5617e-1	1.5654e-1	1.3865e-1	1.3757e-1	1.3877e-1
SD(Eff)	3.2775e-2	4.5230e-2	2.9895e-2	2.9579e-2	4.4272e-2	4.3766e-2	4.8552e-2
MaxVM	2.1815e8	2.1997e8	2.0012e8	1.9840e8	2.2130e8	2.2181e8	2.2726e8

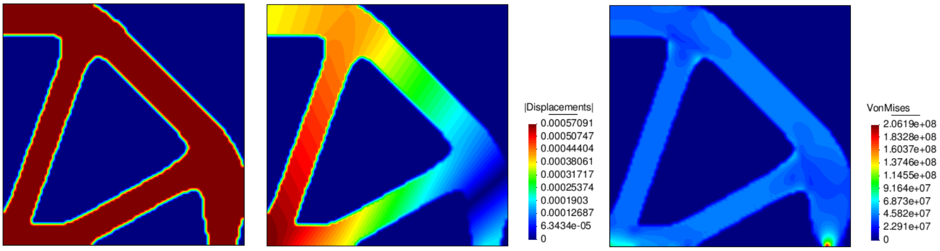
Table 7.11: TELM execution data

7.11 Two Different Loads Michell(TDLM)

TDLM test is described in chapter 4 in figure 4.4(c). First, we describe the results obtained for volume minimization and then, for compliance minimization.

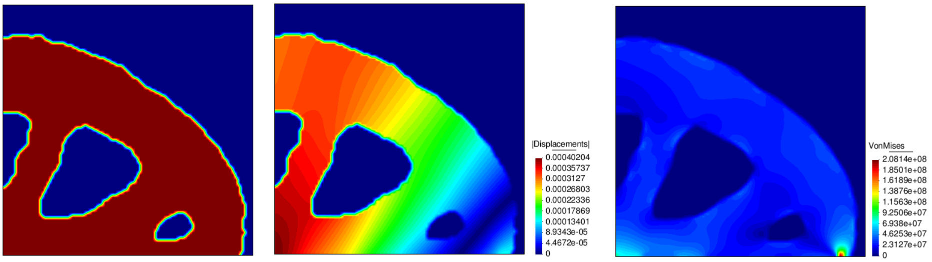
7.11.1 Results for volume minimization methods

Figures 7.92 and 7.93 show results for the SIMP-SC and NSGA-SC respectively: (a) the material distribution on the design domain, (b) the normalized displacements over the structure, and c) the von Mises stresses over the structure.



(a) TDLM material (b) TDLM displacements (c) TDLM VonMises

Figure 7.92: TDLM visual results with SIMP-SC

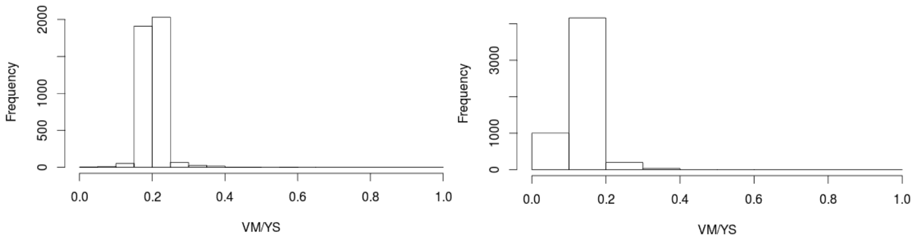


(a) TDLM material (b) TDLM displacements (c) TDLM VonMises

Figure 7.93: TDLM visual results with NSGA-SC

SIMP-SC shows the most aesthetic shape according to the material distribution and the best minimization performance according to the volume reached (shown in the volume minimization columns of table 7.12), where the volumes are $4.1601e - 1$ and $5.4000e - 1$ for SIMP-SC and NSGA-SC respectively.

Figures 7.94(a) and (b) show the histograms of efficiency for SIMP-SC and NSGA-SC respectively.



(a) SIMP-SC

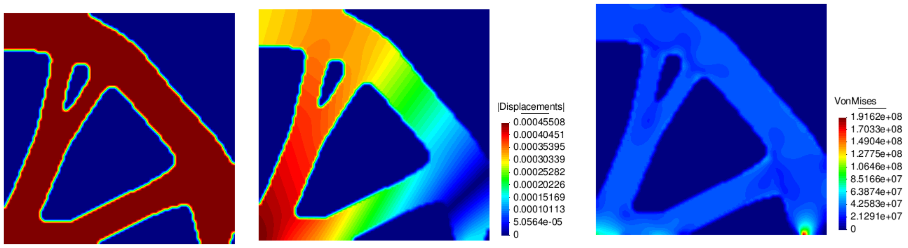
(b) NSGA-SC

Figure 7.94: TDLM VM/YS elemental histogram

SIMP-SC shows the most efficient performance, according to the histograms and the average and standard deviation (shown in the volume minimization columns of table 7.12), due to the average of elemental efficiencies is closer to 1 when using SIMP-SC than when using NSGA-SC.

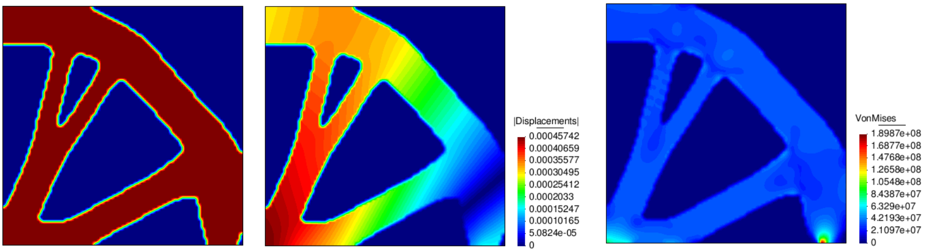
7.11.2 Results for compliance minimization methods

Figures 7.95, 7.96, 7.97, 7.98 and 7.99 show results for the SIMP-SVC, SIMP, GA, UMDA and NSGA-VC, respectively: (a) the material distribution on the design domain, (b) the normalized displacements over the structure, and c) the von Mises stresses over the structure.



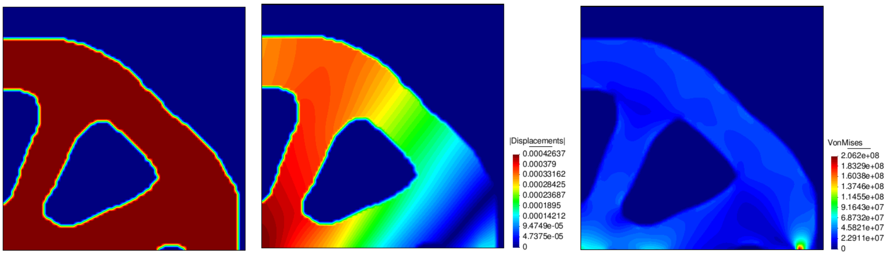
(a) TDLM material (b) TDLM displacements (c) TDLM VonMises

Figure 7.95: TDLM visual results with SIMP-SVC



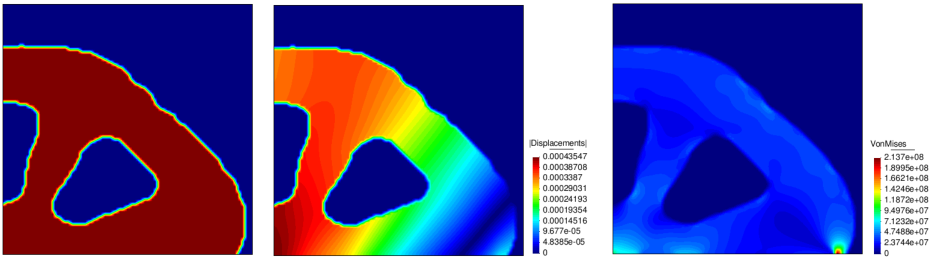
(a) TDLM material (b) TDLM displacements (c) TDLM VonMises

Figure 7.96: TDLM visual results with SIMP



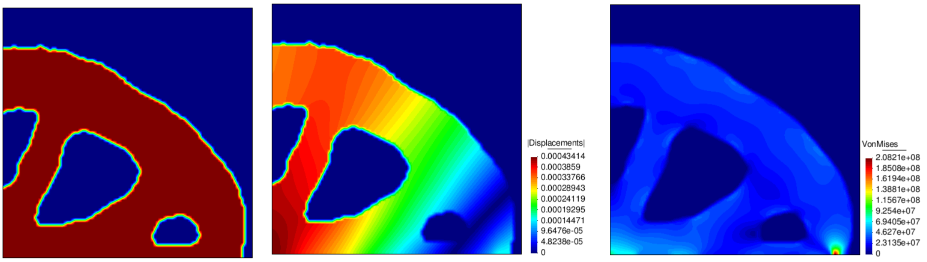
(a) TDLM material (b) TDLM displacements (c) TDLM VonMises

Figure 7.97: TDLM visual results with GA



(a) TDLM material (b) TDLM displacements (c) TDLM VonMises

Figure 7.98: TDLM visual results with UMDA



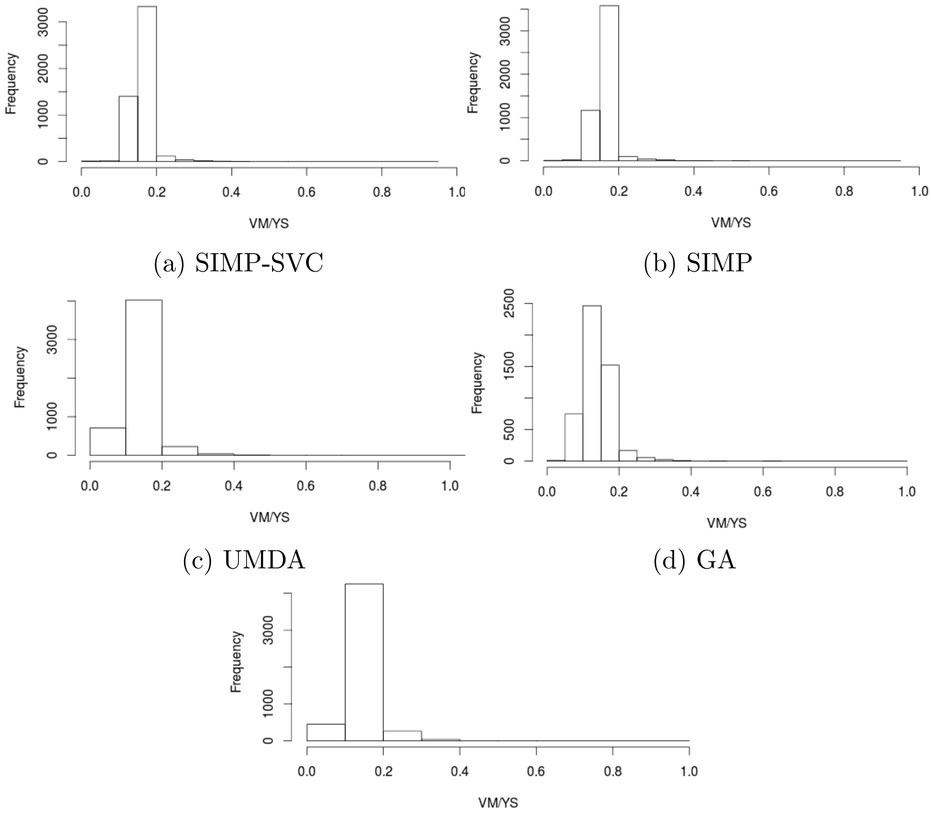
(a) TDLM material (b) TDLM displacements (c) TDLM VonMises

Figure 7.99: TDLM visual results with NSGA-VC

On one hand, SIMP-SVC and SIMP show the most aesthetic shape according to the material distribution, on the other hand, the GA shows the best minimization performance according to the compliance reached (shown in the compliance minimization columns of table 7.12).

Note that in table 7.12, SIMP-SVC reaches a better compliance than SIMP, even when SIMP has reached the maximal iterations, in addition, SIMP-SVC converges in 29 iterations, saving at least the 80.66% of the maximal number.

Figures 7.100(a), (b), (c), (d) and (e) show the histograms of efficiency for SIMP-SVC, SIMP, UMDA, GA and NSGA-VC, respectively.



(a) SIMP-SVC

(b) SIMP

(c) UMDA

(d) GA

(e) NSGA-VC

Figure 7.100: TDLM VM/YS elemental histogram

SIMP shows the most efficient performance, according to the histograms, and the average and standard deviation (shown in the compliance minimization columns of table 7.12), due to the average of elemental efficiencies is closer to 1 when using SIMP than any other.

Table 7.12 has the details of the performance of each method. In the volume minimization columns, it is marked with blue the method that shows the best performance for volume minimization, and with red the method that shows the best performance on efficiency. In the compliance minimization columns it is marked with blue the method that shows the best performance for compliance minimization and with red the method that shows the best performance on efficiency.

	Volume min.		**Compliance min.**				
Data	**SIMP SC**	**NSGA SC**	**SIMP SVC**	**SIMP**	**UMDA**	**GA**	**NSGA VC**
ISF	7.4729e-1	-	7.4729e-1	7.4729e-1	-	-	-
NI	130	500	29	150	500	500	500
IV	1.0e0	1.0e0	1.0e0	1.0e0	1.0e0	1.0e0	1.0e0
FV	4.1601e-1	5.4000e-1	5.0e-1	5.0e-1	5.0e-1	5.0e-1	5.0e-1
FVF	4.1601e-1	5.4000e-1	5.0e-1	5.0e-1	5.0e-1	5.0e-1	5.0e-1
$C(x)$	2.9675e1	2.0983e1	2.3782e1	2.3804e1	2.2795e1	2.2359e1	2.2697e1
ADX	4.6116e-5	4.1457e-5	3.9801e-5	3.9140e-4	4.3113e-5	4.2634e-5	4.4686e-5
|ADX|	4.6116e-5	4.1457e-5	3.9801e-5	3.9140e-4	4.3113e-5	4.2634e-5	4.4686e-5
ADY	-1.1274e-2	-7.9782e-3	-9.0578e-3	-9.0486e-3	-8.6897e-3	-8.5276e-3	-8.6388e-3
|ADY|	1.1274e-2	7.9782e-3	9.0578e-3	9.0486e-3	8.6897e-3	8.5276e-3	8.6388e-3
A||DXY||	5.1584e-4	3.6648e-4	4.1483e-4	4.1426e-4	3.9874e-4	3.9145e-4	3.9682e-4
SF	9.9502e-1	9.9937e-1	9.2524e-1	9.1705e-1	1.0278e0	9.9151e-1	9.9881e-1
A(Eff)	1.9987e-1	1.3013e-1	1.5972e-1	1.6052e-1	1.4162e-1	1.4051e-1	1.4148e-1
SD(Eff)	3.4099e-2	4.3549e-2	3.3142e-2	3.0831e-2	4.5740e-2	4.4637e-2	4.3935e-2
MaxVM	2.1890e8	2.1986e8	2.0355e8	2.0175e8	2.2612e8	2.1813e8	2.1973e8

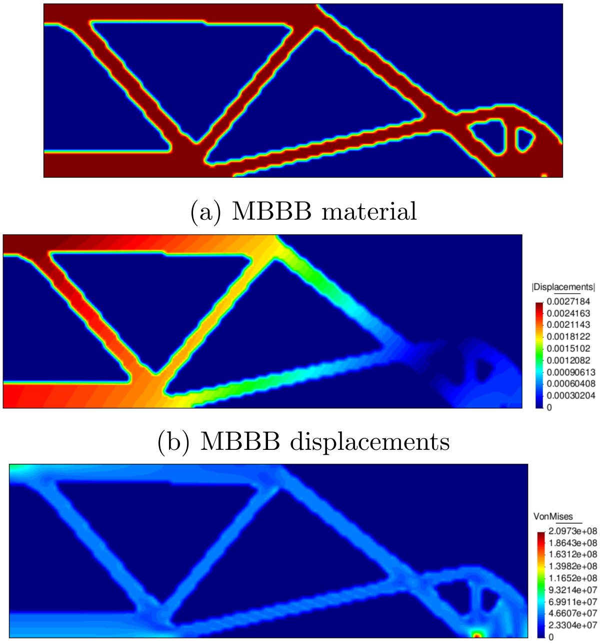
Table 7.12: TDLM execution data

7.12 MBBB

MBBB test is described in chapter 4 in figure **??**. First, we describe the results obtained for volume minimization and then, for compliance minimization.

7.12.1 Results for volume minimization methods

Figures 7.101 and 7.102 show results for the SIMP-SC and NSGA-SC respectively: (a) the material distribution on the design domain, (b) the normalized displacements over the structure, and c) the von Mises stresses over the structure.

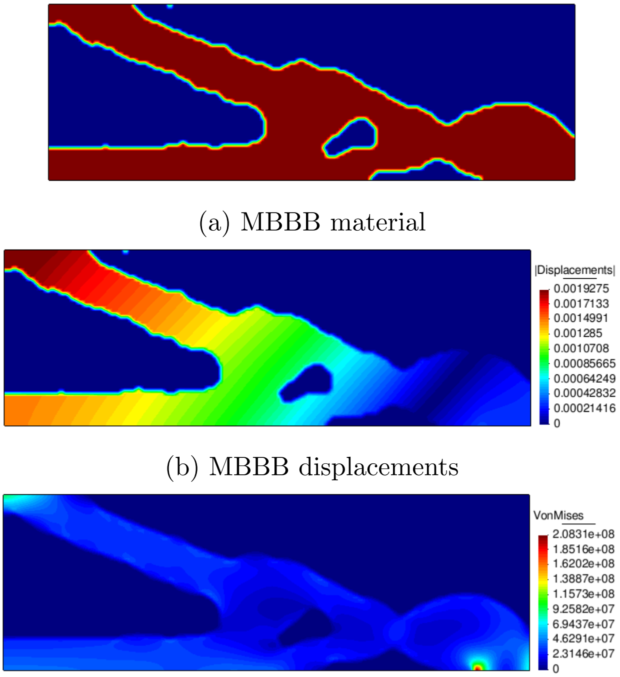


(a) MBBB material

(b) MBBB displacements

(c) MBBB VonMises

Figure 7.101: MBBB visual results with SIMP-SC



(a) MBBB material

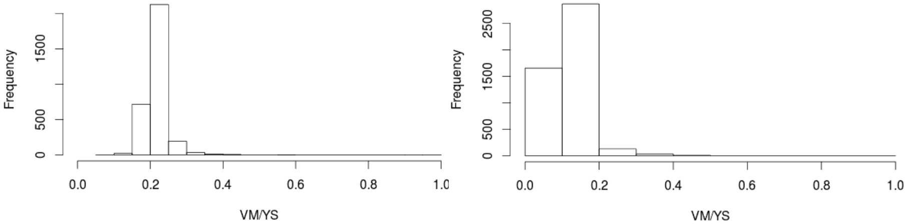
(b) MBBB displacements

(c) MBBB VonMises

Figure 7.102: MBBB visual results with NSGA-SC

SIMP-SC shows the most aesthetic shape according to the material distribution and the best minimization performance according to the volume reached (shown in the volume minimization columns of table 7.13), where the volumes are $8.7304e-1$ and $1.3075e0$ for SIMP-SC and NSGA-SC respectively.

Figures 7.103(a) and (b) show the histograms of efficiency for SIMP-SC and NSGA-SC respectively.



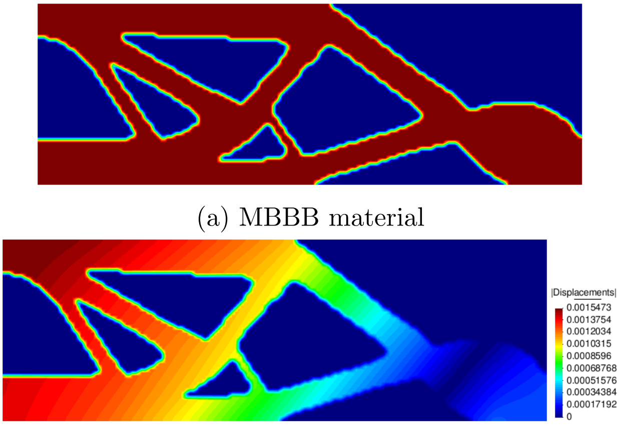
(a) SIMP-SC

(b) NSGA-SC

Figure 7.103: MBBB: VM/YS elemental histogram

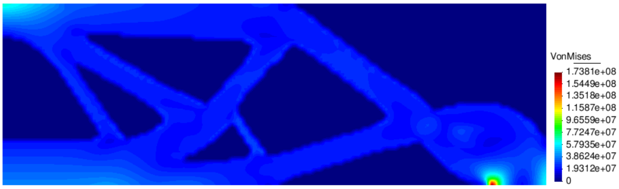
7.12.2 Results for compliance minimization methods

Figures 7.104, 7.105, 7.115, 7.116 and 7.108 show results for the SIMP-SVC, SIMP, GA, UMDA and NSGA-VC, respectively: (a) the material distribution on the design domain, (b) the normalized displacements over the structure, and c) the von Mises stresses over the structure.



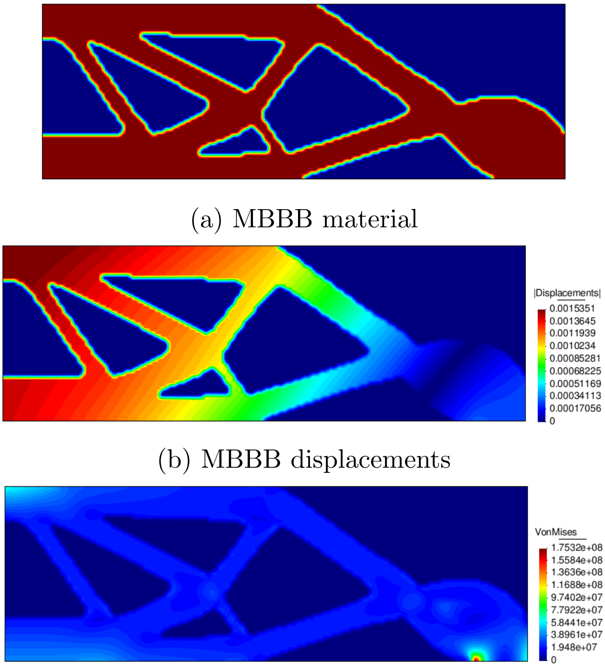
(a) MBBB material

(b) MBBB displacements



(c) MBBB VonMises

Figure 7.104: MBBB visual results with SIMP-SVC

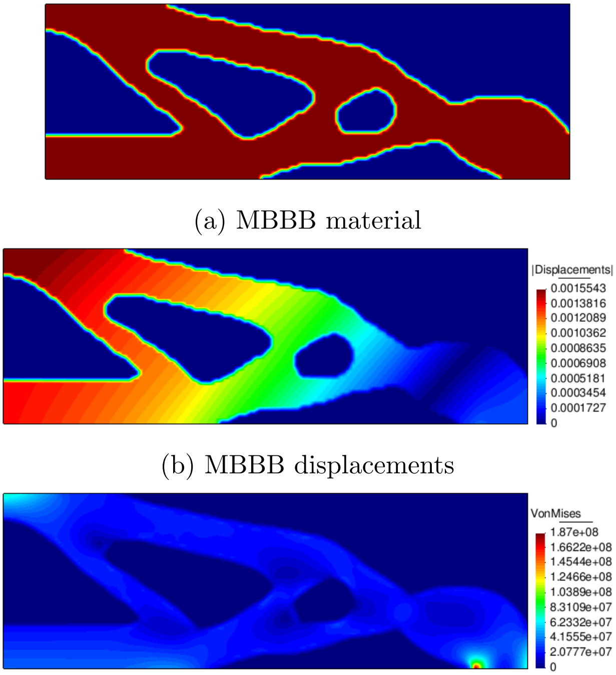


(a) MBBB material

(b) MBBB displacements

(c) MBBB VonMises

Figure 7.105: MBBB visual results with SIMP

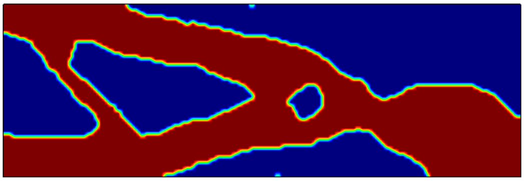


(a) MBBB material

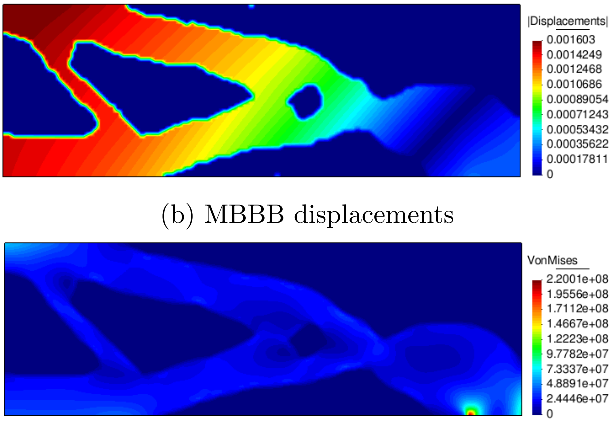
(b) MBBB displacements

(c) MBBB VonMises

Figure 7.106: MBBB visual results with GA



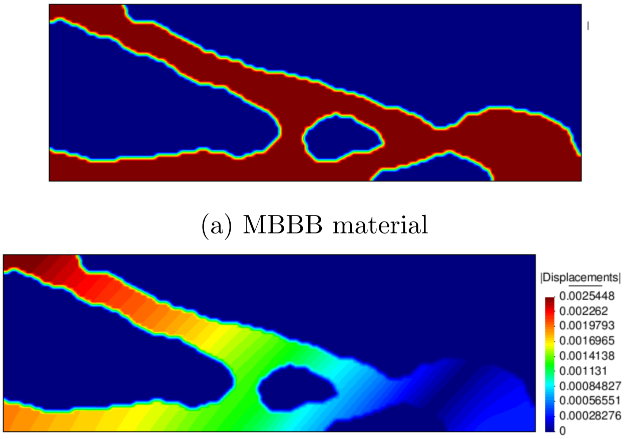
(a) MBBB material



(b) MBBB displacements

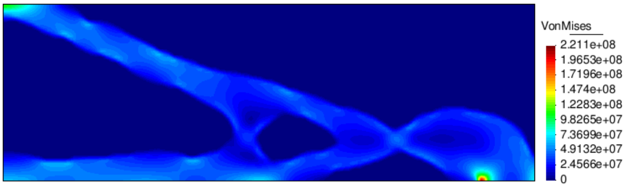
(c) MBBB VonMises

Figure 7.107: MBBB visual results with UMDA



(a) MBBB material

(b) MBBB displacements



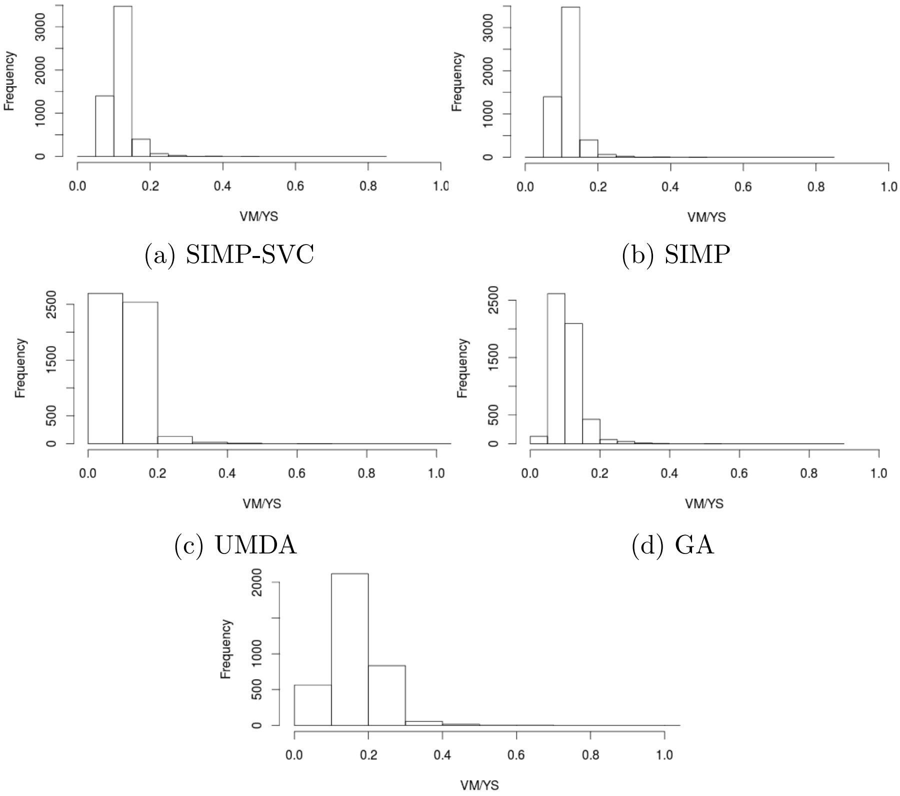
(c) MBBB VonMises

Figure 7.108: MBBB visual results with NSGA-VC

SIMP shows the most aesthetic shape according to the material distribution and the best minimization performance according to the compliance reached (shown in the compliance minimization columns of table 7.13).

Note that SIMP does not converge within the maximal iterations, and SIMP-SVC does it in 32, saving at least the 78.66% of the iterations. The compliance reached is $4.1311e1$ for SIMP and $4.1618e1$ for SIMP-SVC, which do not represent a significant difference.

Figures 7.109(a), (b), (c), (d) and (e) show the histograms of efficiency for SIMP-SVC, SIMP, UMDA, GA and NSGA-VC, respectively.



(a) SIMP-SVC

(b) SIMP

(c) UMDA

(d) GA

(e) NSGA-VC

Figure 7.109: MBBB: VM/YS elemental histogram

SIMP and SIMP-SVC show the most efficient performance, according to the histograms, and the average and standard deviation (shown in the compliance minimization columns of table 7.13), due to the average of elemental efficiencies is closer to 1 when using SIMP-SVC than any other.

Table 7.13 has the details of the performance of each method. In the volume minimization columns, it is marked with blue the method that shows the best performance for volume minimization, and with red the method that shows the best performance on efficiency. In the compliance mini-

mization columns it is marked with blue the method that shows the best performance for compliance minimization and with red the method that shows the best performance on efficiency.

	Volume min.		**Compliance min.**				
Data	**SIMP SC**	**NSGA SC**	**SIMP SVC**	**SIMP**	**UMDA**	**GA**	**NSGA VC**
ISF	7.4724e-1	-	7.4724e-1	7.4724e-1	-	-	-
NI	174	500	32	150	500	500	500
IV	3.0e0	3.0e0	3.0e0	3.0e0	3.0e0	3.0e0	
FV	8.7304e-1	1.3075e0	1.5e0	1.5e0	1.5e0	1.5e0	1.5e0
FVF	2.9101e-1	4.3583e-1	5.0e-1	5.0e-1	5.0e-1	5.0e-1	5.0e-1
$C(x)$	7.2730e1	1.1300e1	4.1618e1	4.1311e1	4.3148e1	4.1633e1	6.7223e1
ADX	-5.6241e-5	5.5125e-5	-4.0913e-5	-3.9833e-5	-4.0331e-5	-4.5024e-5	-6.6861e-5
|ADX|	5.6241e-5	5.5125e-5	4.0913e-5	3.9833e-5	4.0331e-5	4.5024e-5	6.6861e-5
ADY	-5.0429e-2	-3.5388e-2	-2.8525e-2	-2.8313e-2	-2.9599e-2	-2.8562e-2	-4.6623e-2
|ADY|	5.0429e-2	3.5388e-2	2.8525e-2	2.8313e-2	2.9599e-2	2.8562e-2	4.6623e-2
A||DXY||	-2.65497e-3	1.8636e-3	1.5020e-3	1.4909e-3	1.5585e-3	1.5041e-3	2.4550e-3
SF	9.9933e-1	9.9976e-1	8.3442e-1	8.4178e-1	1.0569e0	8.979624e-1	1.0596e0
A(Eff)	2.1763e-1	1.2439e-1	1.1495e-1	1.1495e-1	1.0566e-1	1.0571e-1	1.6350e-1
SD(Eff)	3.6821e-2	5.2451e-2	3.3244e-2	3.3244e-2	4.8247e-2	4.2546e-2	6.6697e-2
MaxVM	2.1985e8	2.1994e8	1.8357e8	1.8519e8	2.3253e8	1.9755e8	2.3311e8

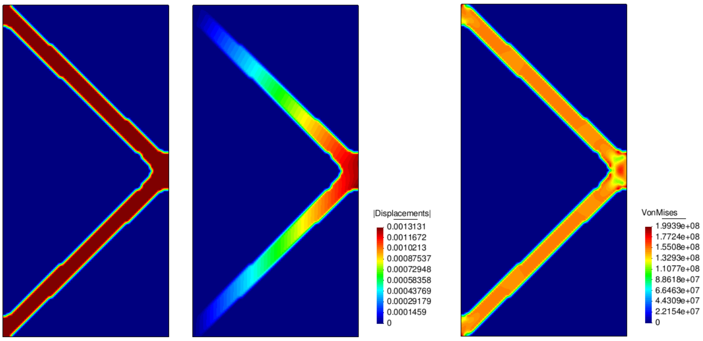
Table 7.13: MBBB execution data

7.13 Two Bars

Two bars test is described in chapter 4 in figure 4.6. First, we describe the results obtained for volume minimization and then, for compliance minimization.

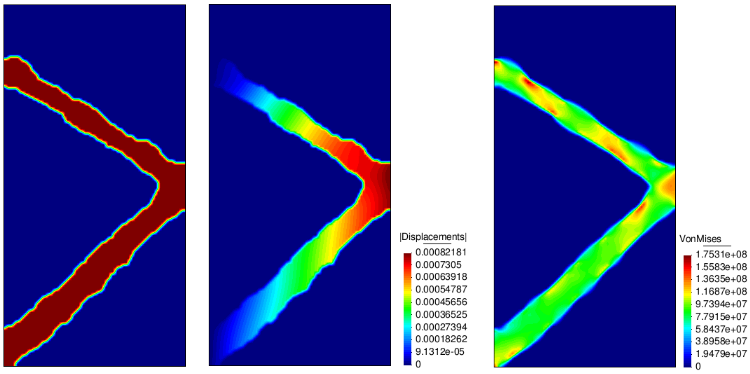
7.13.1 Results for volume minimization methods

Figures 7.110 and 7.111 show results for the SIMP-SC and NSGA-SC respectively: (a) the material distribution on the design domain, (b) the normalized displacements over the structure, and c) the von Mises stresses over the structure.



(a) Two bars material (b) Two bars displacements (c) Two bars Von-Mises

Figure 7.110: Two Bars visual results with SIMP-SC

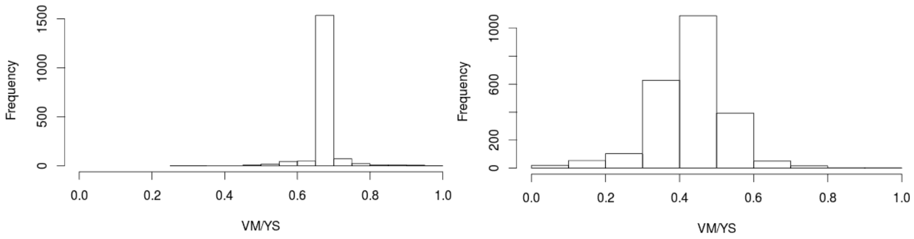


(a) TwoBars material (b) TwoBars displacements (c) TwoBars VonMises

Figure 7.111: Two bars visual results with NSGA-SC

SIMP-SC shows the most aesthetic shape according to the material distribution and the best minimization performance according to the volume reached (shown in the volume minimization columns of table 7.14), where the volumes are $2.880e-1$ and $3.6796e-1$ for SIMP-SC and NSGA-SC respectively.

Figures 7.112(a) and (b) show the histograms of efficiency for SIMP-SC and NSGA-SC respectively.



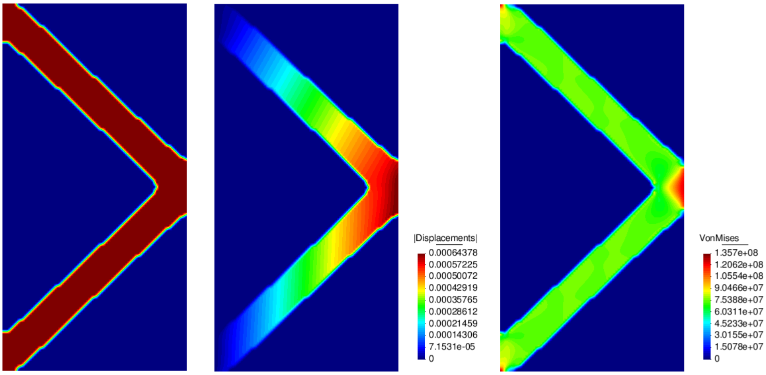
(a) SIMP-SC (b) NSGA-SC

Figure 7.112: Two bars: VM/YS elemental histogram

SIMP-SC shows the most efficient performance, according to the histograms and the average and standard deviation (shown in the volume minimization columns of table 7.14), due to the average of elemental efficiencies is closer to 1 when using SIMP-SC than when using NSGA-SC.

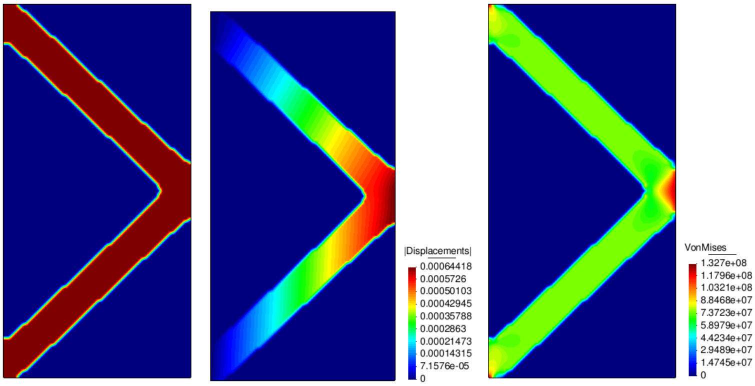
7.13.2 Results for compliance minimization methods

Figures 7.113, 7.114, 7.115, 7.116 and 7.117 show results for the SIMP-SVC, SIMP, GA, UMDA and NSGA-VC, respectively: (a) the material distribution on the design domain, (b) the normalized displacements over the structure, and c) the von Mises stresses over the structure.



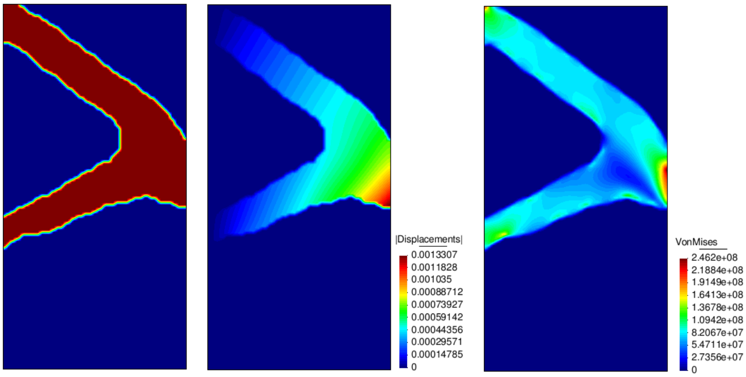
(a) TwoBars material (b) TwoBars displacements (c) TwoBars VonMises

Figure 7.113: Two Bars visual results with SIMP-SVC



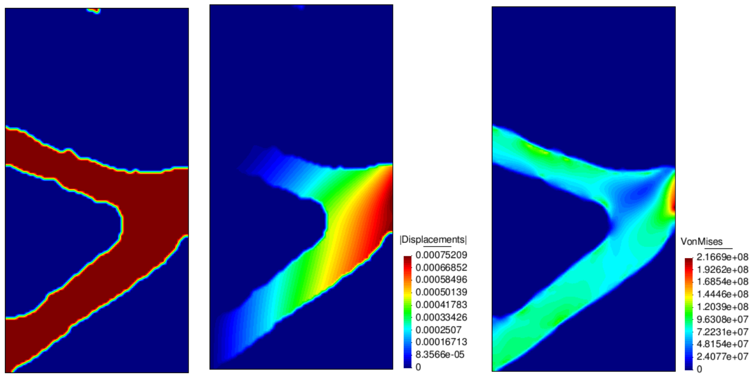
(a) TwoBars material (b) TwoBars displacements (c) TwoBars VonMises

Figure 7.114: Two Bars visual results with SIMP



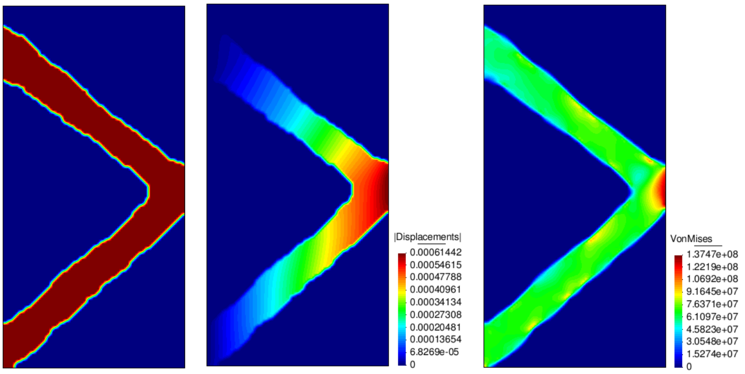
(a) TwoBars material (b) TwoBars displacements (c) TwoBars VonMises

Figure 7.115: TwoBars visual results with GA



(a) TwoBars material (b) TwoBars displacements (c) TwoBars VonMises

Figure 7.116: TwoBars visual results with UMDA

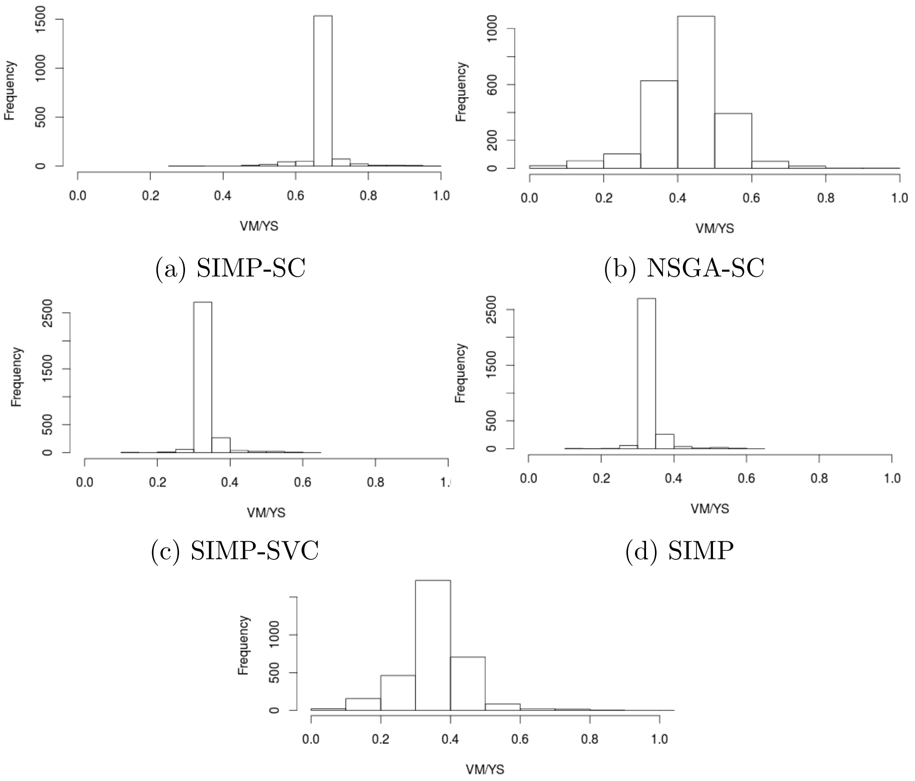


(a) TwoBars material (b) TwoBars displacements (c) TwoBars VonMises

Figure 7.117: Two bars visual results with NSGA-VC

For this case we consider there is a high similarity between visual results of SIMP, SIMP-SC and NSGA-VC according to the material distribution, nevertheless, the NSGA-VC shows the best minimization performance according to the volume reached (shown in the volume minimization columns of table 7.14).

Figures 7.118(a), (b), (c), (d) and (e) show the histograms of efficiency for SIMP-SVC, SIMP, UMDA, GA and NSGA-VC, respectively.



(a) SIMP-SC

(b) NSGA-SC

(c) SIMP-SVC

(d) SIMP

(e) UMDA

Figure 7.118: Two bars: VM/YS elemental histogram

SIMP shows the most efficient performance, according to the histograms, and the average and standard deviation (shown in the compliance minimization columns of table 7.14), due to the average of elemental efficiencies is closer to 1 when using SIMP than any other.

Table 7.14 has the details of the performance of each method. In the volume minimization columns, it is marked with blue the method that shows the best performance for volume minimization, and with red the method that shows the best performance on efficiency. In the compliance minimization columns it is marked with blue the method that shows the best performance for compliance minimization and with red the method that shows the best performance on efficiency.

	Volume min.		**Compliance min.**				
Data	**SIMP SC**	**NSGA SC**	**SIMP SVC**	**SIMP**	**UMDA**	**GA**	**NSGA VC**
ISF	7.4920e-1	-	7.4920e-1	7.4920e-1	-	-	-
NI	192	500	68	96	500	500	500
IV	2.0e0	2.0e0	2.0e0	2.0e0	2.0e0	2.0e0	2.0e0
FV	2.880e-1	3.6796e-1	5.0e-1	5.0e-1	5.0e-1	5.0e-1	5.0e-1
FVF	1.4404e-1	1.8398e-1	2.5e-1	2.5e-1	2.5e-1	2.5e-1	2.5e-1
$C(x)$	1.9446e2	1.3529e2	9.8653e1	9.8640e1	1.3486e2	1.3934e2	9.2952e1
ADX	-3.9891e-10	6.8456e-5	-4.2028e-11	-6.1112e-11	1.6984e-4	-1.9331e-4	-6.9929e-6
|ADX|	1.0968e-5	6.8456e-5	4.4010e-6	4.5801e-6	1.6984e-4	1.9331e-4	1.5632e-5
ADY	-2.1587e-2	-1.5018e-2	-1.0945e-2	-1.0943e-2	-1.4970e-2	-1.5466e-2	-1.0317e-2
|ADY|	2.1587e-2	1.5018e-2	1.0945e-2	1.0943e-2	1.4970e-2	1.5466e-2	1.0317e-2
A||DXY||	1.2699e-3	8.8608e-4	6.4386e-4	6.4378e-4	8.9939-4	9.3379e-4	6.0714e-4
SF	9.9857e-1	9.9920e-1	6.1681e-1	6.0319e-1	8.9939e-4	1.1683e0	6.3975e-1
A(Eff)	6.8345e-1	4.2792e-1	3.4487e-1	3.4487e-1	3.5718e-1	3.5895e-1	3.0893e-1
SD(Eff)	4.3589e-2	9.8310e-2	3.0273e-2	2.9974e-2	9.1786e-2	1.0523e-1	5.7233e-2
MaxVM	2.1968e8	2.1982e8	1.3569e8	1.3270e8	2.2657e+8	2.5703e8	1.4074e8

Table 7.14: Two Bars execution data

7.14 Results summary

The table 7.15 contains a numerical comparative of the methods SIMP and SIMP-SVC according to the computational cost saved and the objective function reached. The columns show the following: 1) the name of the test, 2) the percentage of iterations of SIMP saved by SIMP-SVC, 3) the difference of the compliance between SIMP and SIMP-SVC, where the cells marked with * represent that the SIMP-SVC reaches a better compliance than SIMP, even with a lower number of iterations, 4) the percentage that the difference value in third column represents of the total compliance reached by SIMP, in order to demonstrate that the difference is not significant.

Test	**SIMP-SVC** **% iterations**	**C(x)** **difference**	**% difference**
CLC	71.34%	0.57	0.2475%
CLB	74%	0.83	0.6470%
CLT	74%	0.83	0.6470%
SCLC	82%	0.039*	0.0694%
CLB	84%	0.019*	0.0671%
SCLT	84%	0.019*	0.0671%
LLC	80%	0.11	0.5291%
LLT	81.33%	0.853*	4.0534%
OLM	80%	0.414	1.4904%
TELM	80.66%	0.023*	0.1020%
TDLM	80.66%	0.018*	0.0756%
MBBB	78.66%	0.307	0.7431%
TBars	29.17%	0.013	0.01317%

Table 7.15: Numerical comparative of SIMP and SIMP-SVC

The following is a list of every test and the method with the better performance for volume and compliance problems, according to the objective function obtained. As mentioned in this chapter, SIMP based methods has a better aesthetic solution for almost all tests, so, in this results analysis it is only considered objective functions results.

	Best performance method	
Test	**Volume minimization**	**Compliance minimization**
CLC	SIMP-SC	SIMP
CLB	SIMP-SC	GA
CLT	SIMP-SC	GA
SCLC	NSGA-SC	GA
SCLB	NSGA-SC	NSGA-VC
SCLT	NSGA-SC	NSGA-VC
LLC	SIMP-SC	SIMP
LLT	SIMP-SC	SIMP-SVC
OLM	SIMP-SC	GA
TELM	SIMP-SC	GA
TDLM	SIMP-SC	GA
MBBB	SIMP-SC	SIMP
Two Bars	SIMP-SC	NSGA-VC

Chapter 8

Conclusions and future work

8.1 Conclusions

First, we describe the observations and conclusions about local and global optimization methods. Then, we describe observations and conclusions about the results obtained from all the case studies and algorithms.

8.1.1 About the SIMP and SIMP-based methods

- The SIMP is a simple and efficient topology optimization algorithm for compliance minimization with a volume constraint. Its implementation is not computationally expensive, due to there is only one structure evaluation per iteration, and it requires at most hundreds of iterations, which represents less than 5% of the evolutionary algorithms cost.

- The SIMP methodology gives as solution a structure with an aesthetic shape due to the heuristic approach (principally the filtering step). Comparing the visual results, the SIMP based methods frequently are more aesthetic than evolutionary algorithms. In many cases, SIMP is capable to reach objective functions values close to those delivered by global methods in a considerable lower execution time.

- The SIMP principles are based on the non-consideration of the structural self-weight. It is an inconvenience due to it does not represent a real-world problem. No matter, results in this work consider self-weight.

- The SIMP does not reach convergence in many cases and it stops until the maximal iterations are reached, on the other hand, SIMP-SVC detects the stagnations of the objective function, and there are very small changes in the structure through the iterations to determine the convergence. SIMP-SVC delivers competitive results for all the tests using the same convergence parameters, these results are better than the original SIMP in 6/13 tests and similar in the rest of the cases. The reduction of iterations is an average of 75.3707% for 13 tests.

- The SIMP-SC shows a better performance than evolutionary algorithms on visual results and volume minimization in 10/13 tests.

8.1.2 About evolutionary algorithms

- The proposed individual representation based in control points reduces the dimension size an average of 98.771% in the 13 tests, and it makes suitable(along with parallel computing) the use of evolutionary algorithms for the solution of topology optimization problems, for solving compliance and volume problems. This technique can be used for mono and multi-objective algorithms because of it is used to calculate a fitness(or fitnesses) with a lower number of variables, independently of the optimization method.

- The UMDA generates the worst results according to the visual solutions, compliance or efficiency. Nevertheless, the probability distribution calculated during its execution can help to reduce the design domain, this is to say, when learning the probability distribution, the probability of existing in some elements is set to zero, this reduces the search domain.

- The GA reaches a better objective function that others algorithms in 6 of 13 tests for compliance minimization. The reason of this is the capability of GA for exploring solutions with shapes that SIMP based methods can not explore.

- NSGA-II has an interesting performance in some tests for compliance and volume problems. Despite this algorithm is employed in a simple way (it does not consider constraints neither the use of filtering techniques), it delivers the best solution in 3 of 13 tests for compliance minimization and 3 of 13 tests for volume minimization. In contrast with the other algorithms, it is used to optimize more than one objective functions.

8.2 Future work

- Propose a 3d numerical benchmark for topology optimization, solve it with the SIMP, SIMP-SVC and SIMP-SC and report the results.

- Based on the SIMP analysis, generate a local optimization algorithm for the solution of volume problems, adding the stress constraint directly to the augmented objective function, and develop the variable update equation considering the stress constraint. There are methods that work for volume problems as evolutionary structural optimization, no matter, those methods do not have a mathematical basis that ensure, at least, local optimality.

- For compliance problems solved by evolutionary algorithms, a fraction of the population according to the vv parameter is selected. This population fraction is subject to a volume fitting procedure, which consists in modifying the control points heights until reaching the volume constraint. This causes selecting only a small fraction of the population and in consequence, a high selection pressure. A proposal is to modify the control points for each individual to fullfil the volume constraint, this way, we can increase the number of selected individuals, as consequence, reducing the selection pressure, and increasing the exploration of the search space, and the use of the information of those individuals which are not selected, but that are evaluated.

- Apply a filter with the purpose of getting smoother structures. It is possible to reduce the stress concentrators by smoothing the shape borders, in other words, smooth borders could improve the structure performance delivering lower objective function values and aesthetic structures.

- Note that, in figures of the Results chapter, the contours of von Mises stresses are smoother than the material distribution contours, which present abrupt changes on the border. For example the contour of figure 7.35(c) is smoother than those of figure 7.35(a). Based on this observation, we could propose a post-process that improve the appearance of the results.

- Continue exploring other ways of representation for individuals in order to reduce the dimension of the problem for topology optimization.

Bibliography

[1] Femt. `http://personal.cimat.mx:8181/~miguelvargas/FEMT/`. Accessed: 2016-08-02.

[2] Gid. `http://www.gidhome.com/`. Accessed: 2016-08-03.

[3] Kanpur genetic algorithms laboratory. `http://www.iitk.ac.in/kangal/codes.shtml`. Accessed: 2016-08-09.

[4] Gregoire Allaire, C. Dapogny, and P. Frey. Shape optimization with a level set based mesh evolution method. *Computer Methods in Applied Mechanics and Engineering*, 282:22–53, 2014.

[5] Grégoire Allaire, Charles Dapogny, and Pascal Frey. Shape optimization with a level set based mesh evolution method. *Computer Methods in Applied Mechanics and Engineering*, 282:22–53, 2014.

[6] Gregoire Allaire and Francois Jouve. A level-set method for vibration and multiple loads structural optimization. *Computer Methods in Applied Mechanics and Engineering*, 194:3269–3290, 2005.

[7] Grégoire Allaire and François Jouve. A level-set method for vibration and multiple loads structural optimization. *Computer methods in applied mechanics and engineering*, 194(30):3269–3290, 2005.

[8] Gregoire Allaire, Francois Jouve, and Anca-Maria Toader. A level-set method for shape optimization. *Comptes Rendu Mathematique*, 334:1125–1130, 2002.

[9] Grégoire Allaire, François Jouve, and Anca-Maria Toader. A level-set method for shape optimization. *Comptes Rendus Mathematique*, 334(12):1125–1130, 2002.

[10] Gregoire Allaire, Francois Jouve, and Anca-Maria Toader. Structural optimization using sensitivity analysis and a level-set method. *Journal of Computational Physics*, 194:363–393, 2004.

[11] Grégoire Allaire, François Jouve, and Anca-Maria Toader. Structural optimization using sensitivity analysis and a level-set method. *Journal of computational physics*, 194(1):363–393, 2004.

[12] S. Amstutz, A. A. Novotny, and E. A. de Souza Neto. Topological derivative-based topology optimization of structures subject to drucker-prager stress constraints. *Computer Methods in Applied Mechanics and Engineering*, 233-236:123–136, 2012.

[13] Samuel Amstutz and Heiko Andra. A new algorithm for topology optimization using a level-set method. *Journal of Computational Physics*, 216:573–588, 2006.

[14] Samuel Amstutz and Antonio A Novotny. Topological optimization of structures subject to von mises stress constraints. *Structural and Multidisciplinary Optimization*, 41(3):407–420, 2010.

[15] R. Balamurugan, C.V. Ramakrishnan, and N. Singh. Performance evaluation of a two stage adaptive genetic algorithm (tsaga) in structural topology optimization. *Applied Soft Computing*, 8:1607–1624, 2008.

[16] R Balamurugan, CV Ramakrishnan, and Nidur Singh. Performance evaluation of a two stage adaptive genetic algorithm (tsaga) in structural topology optimization. *Applied Soft Computing*, 8(4):1607–1624, 2008.

[17] M. P. Bendsoe. Optimal shape design as a material distribution problem. *Structural optimization*, 1:193–202, 1989.

[18] M. P. Bendsoe and N. Kikuchi. Generating optimal topologies in structural design using a homogenization method. *Computer Methods Applied Mechanics and Engineering*, 71:197–224, 1988.

[19] M. P. Bendsoe and O. Sigmund. Material interpolation schemes in topology optimization. *Archive of Applied Mechanics*, 69:635–654, 1999.

[20] Martin P Bendsøe. Optimal shape design as a material distribution problem. *Structural optimization*, 1(4):193–202, 1989.

[21] Martin P Bendsøe and Ole Sigmund. Material interpolation schemes in topology optimization. *Archive of applied mechanics*, 69(9-10):635–654, 1999.

[22] T. Borrvall. Topology optimization of elastic continua using restriction. *Archives of Computational Methods in Engineering*, 8:351–385, 2001.

[23] S Botello, H Esqueda, F Gómez, M Moreles, and E Onate. Módulo de aplicaciones del método de los elementos finitos mefi 1.0, chapter manual teórico. *CIMAT & CIMNE*, pages 6–69, 2004.

[24] Blaise Bourdin and Antonin Chambolle. Design-dependent loads in topology optimization. *ESAIM: Control, Optimisation and Calculus of Variations*, 9:19–48, 2003.

[25] T. E. Bruns and D. A. Tortorelli. An element removal and reintroduction strategy for the topology optimization of structures and compliant mechanism. *International Journal for Numerical Methods in Engineering*, 57:1413–1430, 2003.

[26] Sujin Bureerat and Jumlong Limtragool. Performance enhancement of evolutionary search for structural topology optimisation. *Finite Elements in Analysis and Design*, 42:547–566, 2006.

[27] Sujin Bureerat and Jumlong Limtragool. Performance enhancement of evolutionary search for structural topology optimisation. *Finite Elements in Analysis and Design*, 42(6):547–566, 2006.

[28] Sujin Bureerat and Jumlong Limtragool. Structural topology optimisation using simulated annealing with multiresolution design variables. *Finite Elements in Analysis and Design*, 44:738–747, 2008.

[29] Sujin Bureerat and Jumlong Limtragool. Structural topology optimisation using simulated annealing with multiresolution design variables. *Finite Elements in Analysis and Design*, 44(12):738–747, 2008.

[30] Shouyu Cai and Weihong Zhang. Stress constrained topology optimization with free-form design domains. *Computer Methods in Applied Mechanics and Engineering*, 289:267–290, 2015.

[31] Shouyu Cai, Weihong Zhang, Jihong Zhu, and Tong Gao. Stress constrained shape and topology optimization with fixed mesh: A b-spline finite cell method combined with level-set function. *Computer Methods in Applied Mechanics and Engineering*, 278:361–387, 2014.

[32] Miguel Carrasco, Benjamin Ivorra, and Angel Manuel Ramos. Stochastic topology design optimization for continuous elastic materials. *Computer Methods in Applied Mechanics and Engineering*, 289:131–154, 2015.

[33] R. Cazacu and L. Grama. Overview of structural optimization methods for plane and solid structures. *Annals of the University of Oradea*, 2014.

[34] Jean Cea, Stephane Garreau, Philippe Guillaume, and Mohamed Masmoudi. The shape and topological optimizations connection. *Computer Methods in Applied Mechanics and Engineering*, 188:713–726, 2000.

[35] Rohit Chandra. *Parallel programming in OpenMP*. Morgan Kaufmann, 2001.

[36] Bing-Chung Chen and Noboru Kikuchi. Topology optimization with design-dependent loads. *Finite Elements in Analysis and Design*, 37:57–70, 2001.

[37] Shikui Chen, Michael Yu Wang, and Ai Qun Liu. Shape feature control in structural topology optimization. *Computer-Aided Design*, 40:951–962, 2008.

[38] C Chow and C Liu. Approximating discrete probability distributions with dependence trees. *IEEE transactions on Information Theory*, 14(3):462–467, 1968.

[39] Vincent A. Cicirello and Robert Cernera. Profiling the distance characteristics of mutation operators for permutation-based genetic algorithms. *International Florida Artificial Intelligence Research Society Conference*, 26:46–51, 2013.

[40] Jeremy S De Bonet, Charles L Isbell, Paul Viola, et al. Mimic: Finding optima by estimating probability densities. *Advances in neural information processing systems*, pages 424–430, 1997.

[41] J. D. Deaton and R. V. Grandhi. A survey of structural and multidisciplinary continuum topology optimization: post 2000. *Structural and Multidisciplinary Optimization*, 49:1–38, 2014.

[42] Joshua D Deaton and Ramana V Grandhi. A survey of structural and multidisciplinary continuum topology optimization: post 2000. *Structural and Multidisciplinary Optimization*, 49(1):1–38, 2014.

[43] Kalyanmoy Deb. Multi-objective optimisation using evolutionary algorithms: an introduction. In *Multi-objective evolutionary optimisation for product design and manufacturing*, pages 3–34. Springer, 2011.

[44] Kalyanmoy Deb and Ram B Agrawal. Simulated binary crossover for continuous search space. *Complex Systems*, 9(3):1–15, 1994.

[45] Kalyanmoy Deb and Samir Agrawal. A niched-penalty approach for constraint handling in genetic algorithms. In *Artificial Neural Nets and Genetic Algorithms*, pages 235–243. Springer, 1999.

[46] Kalyanmoy Deb, Samir Agrawal, Amrit Pratap, and Tanaka Meyarivan. A fast elitist non-dominated sorting genetic algorithm for

multi-objective optimization: Nsga-ii. In *International Conference on Parallel Problem Solving From Nature*, pages 849–858. Springer, 2000.

[47] Kalyanmoy Deb and Debayan Deb. Analysing mutation schemes for real-parameter genetic algorithms. *International Journal of Artificial Intelligence and Soft Computing*, 4(1):1–28, 2014.

[48] Kalyanmoy Deb and Amarendra Kumar. Real-coded genetic algorithms with simulated binary crossover: Studies on multimodal and multiobjective problems. *Complex Systems*, 9:431–454, 1995.

[49] Luca Dede, Micheal J. Borden, and Thomas J. R. Hughes. Isogeometric analysis for topology optimization with a phase field model. *Archives of Computational Methods in Engineering*, 19:427–465, 2012.

[50] H. A. Eschenauer, V. V. Kobelev, and A. Schumacher. Bubble method for topology and shape optimization of structures. *Structural Optimization*, 8:42–51, 1994.

[51] Hans Eschenauer, Niels Olhoff, and Walter Schnell. *Applied structural mechanics: fundamentals of elasticity, load-bearing structures, structural optimization: including exercises*. Springer Science & Business Media, 1997.

[52] José Miguel Vargas Félix. *Cálculo de Estructuras Utilizando Elemento Finito con Cómputo en Paralelo*. PhD thesis, Centro de Investigación en Matemáticas A.C. Guanajuato, Mexico, 11 2010. phD Thesis.

[53] N. P. Garcia-Lopez, M. Sanchez-Silva, A. L. Medaglia, and A. Chateauneuf. An improved robust topology optimization approach using multiobjective evolutionary algorithms. *Computers and Structures*, 125:1–10, 2013.

[54] William Gropp, Ewing Lusk, and Anthony Skjellum. *Using MPI: portable parallel programming with the message-passing interface*, volume 1. MIT press, 1999.

[55] James K. Guest and Lindsey C. Smith Genut. Reducing dimensionality in topology optimization using adaptive design variable fields. *International Journal for Numerical Methods in Engineering*, 81:1019–1045, 2010.

[56] Xu Guo, Wei Sheng Zhang, Michael Yu Wang, and Peng Wei. Stress-related topology optimization via level set approach. *Computer Methods in Applied Mechanics and Engineering*, 200:3439–3452, 2011.

[57] Xu Guo, Weisheng Zhang, and Li Zhang. Robust structural topology optimization considering boundary uncertainties. *Computer Methods in Applied Mechanics and Engineering*, 253:356–368, 2013.

[58] Xu Guo, Weisheng Zhang, and Whenliang Zhong. Explicit feature control in structural topology optimization via level set method. *Computer Methods in Applied Mechanics and Engineering*, 272:354–378, 2014.

[59] Deepti Gupta and Shabina Ghafir. An overview of methods maintaining diversity in genetic algorithms. *International Journal of Emerging Technology and Advanced Engineering 5*, 2:56–60, 2012.

[60] Wilfried Hansel, Andre Treptow, Wilfried Becker, and Bernd Freisleben. A heuristic and a genetic topology optimization algorithm for weight-minimal laminate structures. *Composite Structures*, 58:287–294, 2002.

[61] Wilfried Hansel, André Treptow, Wilfried Becker, and Bernd Freisleben. A heuristic and a genetic topology optimization algorithm for weight-minimal laminate structures. *Composite Structures*, 58(2):287–294, 2002.

[62] B Hassani and E Hinton. A review of homogenization and topology opimization ii—analytical and numerical solution of homogenization equations. *Computers & structures*, 69(6):719–738, 1998.

[63] Behrooz Hassani and Ernest Hinton. A review of homogenization and topology optimization i—homogenization theory for media with periodic structure. *Computers & Structures*, 69(6):707–717, 1998.

[64] J. H. Holland. *Adaptation in Natural and Artificial Systems.* MIT University Press Group, 1992.

[65] X. Huang and Y. M. Xie. Evolutionary topology optimization of continuum structures with an additional displacement constraint. *Structural and Multidisciplinary Optimization*, 40:409–416, 2010.

[66] Xiaodong Huang and Mike Xie. *Evolutionary topology optimization of continuum structures: methods and applications.* John Wiley & Sons, 2010.

[67] Kai A. James, Edmund Lee, and Joaquim R.R.A. Martins. Stress-based topology optimization using an isoparametric level set method. *Finite Elements in Analysis and Design*, 58:20–30, 2012.

[68] Seung Hyun Jeong, Dong-Hoon Choi, and Gil Ho Yoon. Separable stress interpolation scheme for stress-based topology optimization with multiple homogenous materials. *Finite Elements in Analysis and Design*, 82:16–31, 2014.

[69] Seung Hyun Jeong, Dong-Hoon Choi, and Gil Ho Yoon. Separable stress interpolation scheme for stress-based topology optimization with multiple homogenous materials. *Finite Elements in Analysis and Design*, 82:16–31, 2014.

[70] Seung Hyun Jeong, Gil Ho Yoon, Akihiro Takezawa, and Dong-Hoon Choi. Development of a novel phase-field method for local stress-based shape and topology optimization. *Computers and Structures*, 132:84–98, 2014.

[71] Helio Emmendoerfer Jr. and Eduardo Alberto Fancello. A level set approach for topology optimization with local stress constraints. *International Journal for Numerical Methods in Engineering*, 99:129–156, 2014.

[72] Zhan Kang and Yiqiang Wang. A nodal variable method of structural topology optimization based on shepard interpolant. *International Journal for Numerical Methods in Engineering*, 90:329–342, 2012.

[73] Zhan Kang and Yiqiang Wang. Integrated topology optimization with embedded movable holes based on combined description by material density and level sets. *Computer Methods in Applied Mechanics and Engineering*, 255:1–13, 2013.

[74] Zhang Kang and Yiqiang Wang. Structural topology optimization based on non-local shepard interpolation of density field. *Computer Methods in Applied Mechanics and Engineering*, 200:3515–3525, 2011.

[75] Sun Yong Kim, Il Yong Kim, and Chris K. Mechefske. A new efficient convergence criterion for reducing computational expense in topology optimization: reducible design varialbe method. *International Journal for Numerical Methods in Engineering*, 90:752–783, 2012.

[76] M. Akif Kutuk and Ibrahim Gov. A finite element removal method for 3d topology optimization. *Advances in Mechanical Engineering*, 2013(413463), 2013.

[77] M Akif Kütük and İbrahim Göv. A finite element removal method for 3d topology optimization. *Advances in Mechanical Engineering*, 5:413463, 2013.

[78] Pedro Larranaga and Jose A Lozano. *Estimation of distribution algorithms: A new tool for evolutionary computation*, volume 2. Springer Science & Business Media, 2002.

[79] Euihark Lee and Hae Chang Gea. A strain based topology optimization method for compliant mechanism design. *Structural and Multidisciplinary Optimization*, 49:199–207, 2014.

[80] T. Lewinski and G.I.N. Rozvany. Exact analytical solutions for some popular benchmark problems in topology optimization ii: thresssided polygonal supports. *Structural and Multidisciplinary Optimization*, 33:337–349, 2007.

[81] T. Lewinski and G.I.N. Rozvany. Analytical benchmark for topological optimization iv: square-shaped line support. *Structural and Multidisciplinary Optimization*, 36:143–158, 2008.

[82] T. Lewinski, G.I.N. Rozvany, T. Sokol, and K. Bolbotowski. Exact analytical solutions for some popular benchmark problems in topology optimization iii: L-shaped domains revisited. *Structural and Multidisciplinary Optimization*, 47:937–942, 2013.

[83] Ming-Hua Lin, Jung-Fa Tsai, and Chian-Son Yu. A review of deterministic optimization methods in engineering and management. *Mathematical Problems in Engineering*, 2012, 2012.

[84] Guan-Chun Luh, L. Chun-Yi, and L. Yu-Shu. A binary particle swarm optimization for continuum structural topology optimization. *Applied Soft Computing*, 11(2):2833–2844, 2011.

[85] Guan-Chun Luh and Chun-Yi Lin. Structural topology optimization using ant colony optimization algorithm. *Applied Soft Computing*, 9:1343–1353, 2009.

[86] Junzhao Luo, Zhen Luo, Liping Chen, Liyong Tong, and Michael Yu Wang. A semi-implicit level set method for structural shape and topology optimization. *Journal of Computational Physics*, 227:5561–5581, 2008.

[87] Yangjun Luo and Zhan Kang. Topology optimization of continuum structures with drucker-prager yield. *Computers and Structures*, 90-91:65–75, 2012.

[88] Yangjun Luo, Michael Yu Wang, and Zhan Kang. An enhanced aggregation method for topology optimization with local stress constraints. *Computer Methods in Applied Mechanics and Engineering*, 254:31–41, 2013.

[89] Z. Luo, N. Zhang, W. Gao, and H. Ma. Structural shape and topology optimization using a meshless galerkin level set method. *International Journal for Numerical Methods in Engineering*, 90:369–389, 2012.

[90] Zhen Luo and Liyong Tong. A level set method for shape and topology optimization of large-displacement compliant mechanisms. *International Journal for Numerical Methods in Engineering*, 76:862–892, 2008.

[91] Zhen Luo, Liyong Tong, and Zhang Kang. A level set method for structural shape and topology optimization using radial basis functions. *Computers and Structures*, 87:425–434, 2009.

[92] Zhen Luo, Michael Yu Wang, Shengyin Wang, and Peng Wei. A level set-based parameterization method for structural shape and topology optimization. *International Journal for Numerical Methods in Engineering*, 76:1–26, 2008.

[93] Takahashi M. and Kita H. A crossover operator using independent component analysis for real-coded genetic algorithm. *Proceedings of the 2001 Congress on Evolutionary Computation*, pages 643–649, 2001.

[94] Julio César Madera Quintana. *Hacia una generación más eficiente de algoritmos evolutivos con estimación de distribucines: Pruebas de independencia+ Paralelismo.* PhD thesis, 2009.

[95] K. Matsui and K. Terada. Continuous approximation of material distribution for topology optimization. *International Journal for Numerical Methods in Engineering*, 59:1925–1944, 2004.

[96] Paul E McKenney. Is parallel programming hard, and, if so, what can you do about it? *Linux Technology Center, IBM Beaverton*, 2011.

[97] Fernando Gutiérrez Mendez. *Optimización Multiobjetivo Usando Algoritmos Genéticos Culturales.* PhD thesis, Centro de Investigacion y Estudios Avanzados del Insituto Politecnico Nacional. Mexico DF, 11 2011. Master Thesis.

[98] Salvador Botello Miguel A. Ochoa, S.Ivvan Valdez and Victor Cardoso. Topology optimization benchmark i. results for minimum compilance and minimum volume in plane stress problems. Technical Report I-16-01/09-03-2016, Centro de Investigación en Matemáticas, A.C., Departamento de Ciencias Computacionales, Jalisco S/N, Col. Valenciana CP 36240 Guanajuato Gto., México, March 2016. Technical Report.

[99] H Mühlenbein. The equation for response to selection and its use for prediction. *Evolutionary Computation 5*, 3:303–346, 1998.

[100] P. B. Nakshatrala, D. A. Tortorelli, and K. B. Nakshatrala. Nonlinear structural desing using multiscale topology optimization. *Computer Methods in Applied Mechanics and Engineering*, 261-252:167–176, 2013.

[101] Rajiv Kumar Naveen kumar, Karambir. A comparative analysis of pmx, cx and ox crossover operators for solving travelling salesman problem. *International Journal of Latest Research in Science and Technology 2*, 1:98–101, 2012.

[102] Luis Carretero Neches and Adrián P. Cisilino. Topology optimization of 2d elastic structures using boundary elements. *Engineering Analysis with Boundary Elements*, 32:533–544, 2008.

[103] Tam H. Nguyen, Glaucio H. Paulino, Junho Song, and Chau H. Le. Improving multiresolution topology optimization via multiple discretizations. *International Journal for Numerical Methods in Engineering*, 92:507–530, 2012.

[104] Jorge Nocedal and Stephen J. Wright. *Numerical Optimization.* Springer, 2 edition, 2006.

[105] Norapat Noilublao and Sujin Bureerat. Simultaneous topology, shape, and sizing optimisation of plane trusses with adaptive ground finite elements using moeas. *Mathematical Problems in Engineering*, 2013, 2013.

[106] M. Papadrakakis, N. D. Lagaros, Y. Tsompanakis, and V. Plevris. Large scale structural optimization: Computational methods and optimization algorithms. *Archives of Computational Methods in Engineering*, 3:239–301, 2001.

[107] J. Paris, I. Colominas, F. Navarrina, and M. Casteleiro. Parallel computing in topology optimization of structures with stress constraints. *Computers and Structures*, 125:62–73, 2013.

[108] Jaejong Park and Alok Sutradhar. A multi-resolution method for 3d multi-material topology optimization. *Computer Methods in Applied Mechanics and Engineering*, 285:571–586, 2015.

[109] Xiaoping Qian. Topology optimization in b-spline space. *Computer Methods in Applied Mechanics and Engineering*, 265:15–35, 2013.

[110] Osvaldo Maximo Querin, Mariano Victoria, and Pascual Marti. Topology optimization of truss-like continua with different material properties in tension and compression. *Structural and Multidisciplinary Optimization*, 42:25–32, 2010.

[111] Luis Vicente Santana Quintero. *Un Algoritmo Basado en Evolución Diferencial para Resolver Problemas Multiobjetivo.* PhD thesis, Centro de Investigacion y Estudios Avanzados del Insituto Politecnico Nacional. Mexico DF, 11 2004.

[112] Arash Radman. *Bi-directional Evolutionary Structural Optimization (BESO) for Topology Optimization of Material's Microstructure.* PhD thesis, RMIT University, 2013.

[113] Girdhar Gopal Rakesh Kumar and Rajesh Kumar. Novel crossover operator for genetic algorithm for permutation problems. *International Journal of Soft Computing and Engineering (IJSCE) 2*, 4:252–258, 2013.

[114] Stefan Riehl and Paul Steinmann. A staggered approach to shape and topology optimization using the traction method and an evolutionary-type advancing front. *Computer Methods in Applied Mechanics and Engineering*, 287:1–30, 2015.

[115] Jian Hua Rong, Xiao Hua Liu, Ji Jun Yi, and Jue Hong Yi. An efficient structural topological optimization method for continuum structures with multiple displacement constraints. *Finite Elements in Analysis and Design*, 47:913–921, 2011.

[116] G.I.N. Rozvany. Exact analytical solutions for some popular benchmark problems in topology optimization. *Structural Optimization*, 15:42–48, 1998.

[117] Hyunjin Shin, Akira Todoroki, and Yoshiyasu Hirano. Elite-initial population for efficient topology optimization using multi-objective genetic algorithms. *International Journal of Aeronautical & Space Sciences*, 14:324–333, 2013.

[118] O. Sigmund. A 99 line topology optimization code written in matlab. *Structural and Multidisciplinary Optimization*, 21:120–127, 2001.

[119] James C Spall. Stochastic optimization. In *Handbook of computational statistics*, pages 173–201. Springer, 2012.

[120] M. Stolpe and K. Svanberg. An alternative interpolation scheme for minimum compliance topology optimization. *Structural and Multidisciplinary Optimization*, 22:116–124, 2001.

[121] Dirk Sudholt. Parallel evolutionary algorithms. Technical report, University of Sheffield, UK. Chapter in the Handbook of Computational Intelligence.

[122] K. Svanberg and M. Werme. Topology optimization by a neighbourhood search method based on efficient sensitivity calculations. *International Journal for Numerical Methods in Engineering*, 67:1670–1699, 2006.

[123] Colby C. Swan and Iku Kosaka. Voigt-reuss topology optimization structures with linear elastic material behaviours. *International Journal for Numerical Methods in Engineering*, 40:3033–3057, 1997.

[124] Akihiro Takezawa, Gil Ho Yoon, Seung Hyun Jeong, Makoto Kobashi, and Mitsuru Kitamura. Structural topology optimization with strength and heat conduction constraints. *Computer Methods in Applied Mechanics and Engineering*, 276:341–361, 2014.

[125] Cameron Talischi, Glaucio H. Paulino, Anderson Pereira, and Ivan F. M. Menezes. Polygonal finite elements for topology optimization: A unifying paradigm. *International Journal for Numerical Methods in Engineering*, 82:671–698, 2010.

[126] Rouhollah Tavakoli. Multimaterial topology optimization by volume constrained allen-cahn system and regularized projected steepest descent method. *Computer Methods in Applied Mechanics and Engineering*, 276:534–565, 2014.

[127] Liyong Tong and Jiangzi Lin. Structural topology optimization with implicit design variable-optimality and algorithm. *Finite Elements in Analysis and Design*, 47:922–932, 2011.

[128] B. Ullah and J. Trevelyan. Correlation between hole insertion criteria in a boundary element and level set based topology optimisation method. *Engineering Analysis with Boundary Elements*, 37:1457–1470, 2013.

[129] S Ivvan Valdez, Arturo Hernández, and Salvador Botello. *Efficient estimation of distribution algorithms by using the empirical selection distribution.* INTECH Open Access Publisher, 2010.

[130] S Ivvan Valdez, Arturo Hernández, and Salvador Botello. A boltzmann based estimation of distribution algorithm. *Information Sciences*, 236:126–137, 2013.

[131] N. P. van Dijk, M. Langelaar, and F. van Keulen. Explicit level-set-based topology optimization using an exact heaviside function and consistent sensitivity analysis. *International Journal for Numerical Methods in Engineering*, 91:67–97, 2012.

[132] Miguel Vargas-Felix and Salvador Botello-Rionda. Femt, open source tools for solving large systems of equations in parallel.

[133] Mathias Wallin and Matti Ristinmaa. Howard's algorithm in a phase-field topology optimization approach. *International Journal for Numerical Methods in Engineering*, 94:43–59, 2013.

[134] Mathias Wallin and Matti Ristinmaa. Boundary effects in a phase-field approach to topology optimization. *Computer Methods in Applied Mechanics and Engineering*, 278:145–159, 2014.

[135] Mathias Wallin and Matti Ristinmaa. Topology optimization utilizing inverse motion based form finding. *Computer Methods in Applied Mechanics and Engineering*, 289:316–331, 2015.

[136] Fengwen Wang, Boyan Stefanov Lazarov, Ole Sigmund, and Jakob Sondergaard Jasen. Interpolation scheme for fictitious domain

techniques and topology optimization of finite strain elastic problems. *Computer Methods in Applied Mechanics and Engineering*, 276:453–472, 2014.

[137] Michael Yu Wang and Shengyin Wang. Bilateral filtering for structural topology optimization. *International Journal for Numerical Methods in Engineering*, 63:1911–1938, 2005.

[138] Michael Yu Wang and Xiaoming Wang. ”color” level sets: a multiphase method for structural topology optimization with multiple materials. *Computer Methods in Applied Mechanics and Engineering*, 193:469–496, 2004.

[139] Michael Yu Wang and Xiaoming Wang. A level-set based variational method for design and optimization of heterogeneous objects. *Computer-Aided Design*, 37:321–337, 2005.

[140] Michael Yu Wang, Xiaoming Wang, and Dongming Guo. A level set method for structural topology optimization. *Computer Methods in Applied Mechanics and Engineering*, 192:227–246, 2003.

[141] S. Y. Wang, K. M. Lim, B. C. Khoo, and M. Y. Wang. An extended level set method for shape and topology optimization. *Journal of Computational Physics*, 221:395–421, 2007.

[142] S. Y. Wang and M. Y. Wang. An enhanced genetic algorithm for structural topology optimization. *International Journal for Numerical Methods in Engineering*, 65:18–44, 2006.

[143] Yiqiang Wang, Zhan Kang, and Qizhi He. Adaptive topology optimization with independent error control for separated displacement and density fields. *Computers and Structures*, 135:50–61, 2014.

[144] Yiqiang Wang, Zhan Kang, and Qizhi He. Adaptive topology optimization with independent error control for separated displacement and density fields. *Computers & Structures*, 135:50–61, 2014.

[145] Yiqiang Wang, Zhen Luo, Zhan Kang, and Nong Zhang. A multi-material level set-based topology and shape optimization method.

Computer Methods in Applied Mechanics and Engineering, 283:1570–1586, 2015.

[146] Darrell Whitley. A genetic algorithm tutorial. *Statistics and computing*, 4(2):65–85, 1994.

[147] Liang Xia and Piotr Breitkopf. Concurrent topology optimization design of material and structure within fe^2 nonlinear multiscale analysis framework. *Computer Methods in Applied Mechanics and Engineering*, 278:524–542, 2014.

[148] Liang Xia and Piotr Breitkopf. A reduced multiscale model for nonlinear structural topology optimization. *Computer Methods in Applied Mechanics and Engineering*, 280:117–134, 2014.

[149] Liang Xia and Piotr Breitkopf. Multiscale structural topology optimization with an approximate constitutive model for local material microstructure. *Computer Methods in Applied Mechanics and Engineering*, 286:147–167, 2015.

[150] Liang Xia, Jihong Zhu, Weihong Zhang, and Piotr Breitkopf. An implicit model for the integrated optimization of component layout and structure topology. *Computer Methods in Applied Mechanics and Engineering*, 257:87–102, 2013.

[151] Qi Xia, Tielin Shi, Shiyuan Liu, and Michael Yu Wang. A level set solution to the stress-based structural shape and topology optimization. *Computers and Structures*, 90-91:55–64, 2012.

[152] Qi Xia, Michael Yu Wang, and Tielin Shi. A level set method for shape and topology optimization of both structure and support of continuum structures. *Computer Methods in Applied Mechanics and Engineering*, 272:340–353, 2014.

[153] Qi Xia, Michael Yu Wang, and Tielin Shi. Topology optimization with pressure load through a level set method. *Computer Methods in Applied Mechanics and Engineering*, 283:177–195, 2015.

[154] Qi Xia and Yu Wang. Simultaneous optimization of the materials properties and the topology of functionally graded structures. *Computer-Aided Design*, 40:660–675, 2008.

[155] Yi M Xie and Grant P Steven. A simple evolutionary procedure for structural optimization. *Computers & structures*, 49(5):885–896, 1993.

[156] Huayang Xu, Liwen Guan, Xiang Chen, and Liping Wang. Guide-weight motion for topology optimization of continuum structures including body forces. *Finite Elements in Analysis and Design*, 75:38–49, 2013.

[157] Shintaro Yamasaki, Atsushi Kawamoto, Tsuyoshi Nomura, and Kikuo Fujita. A consistent grayscale-free topology optimization method using the level-set method and zero-level boundary tracking mesh. *International Journal for Numerical Methods in Engineering*, 101:744–773, 2015.

[158] Shintaro Yamasaki, Shinji Nishiwaki, Takayuki Yamada, Kazuhiro Izui, and Masataka Yoshimura. A structural optimization method based on the level set method using a new geometry-based re-initialization scheme. *International Journal for Numerical Methods in Engineering*, 83:1580–1624, 2010.

[159] Tong Yifei, Ye Wei, Yang Zhen, Li Dongbo, and Li Xiangdong. Research on multidisciplinary optimization design of bridge crane. *Mathematical Problems in Engineering*, 2013, 2013.

[160] G.P. Steven Y.M. Xie. A simple evolutionary procedure for structural optimization. *Computers & Structures 5*, 49:885–896, 1993.

[161] Xinjie Yu and Mitsuo Gen. *Introduction to Evolutionary Algorithms.* Springer-Verlag, 1 edition, 2010.

[162] Mei Yulin and Wang Xiaoming. A level set method for structural topology optimization and its applications. *Advances in Engineering Software*, 35:415–441, 2004.

[163] Weisheng Zhang, Wenliang Zhong, and Xu Guo. An explicit length scale control approach in simp-based topology optimization. *Computer Methods in Applied Mechanics and Engineering*, 282:71–86, 2014.

[164] Hongwei Zhao, Kai Long, and Z. D. Ma. Homogenization topology optimization method based on continuous field. *Advances in Mechanical Engineering*, 2010(528397), 2010.

[165] Junpeng Zhao and Chunjie Wang. Robust topology optimization under loading uncertainty based on linear elastic theory and orthogonal diagonalization of symmetric matrices. *Computer Methods in Applied Mechanics and Engineering*, 272:204–218, 2014.

[166] Junpeng Zhao and Chunjie Wang. Robust topology optimization under loading uncertainty based on linear elastic theory and orthogonal diagonalization of symmetric matrices. *Computer Methods in Applied Mechanics and Engineering*, 273:204–218, 2014.

[167] Benliang Zhu, Xianmin Zhang, and Sergej Fatikow. Structural topology and shape optimization using a level set method with distance-suppression scheme. *Computer Methods in Applied Mechanics and Engineering*, 283:1214–1239, 2015.

[168] Ji-Hong Zhu, Wei-Hong Zhang, and Liang Xia. Topology optimization in aircraft and aerospace structures. *Archives of Computational Methods in Engineering*, 2015.

[169] OC Zienkiewicz, RL Taylor, and JZ Zhu. The finite element method: its basis and fundamentals. 2005, 2005.