Topological Optimum Design using Genetic Algorithms

Couro Kane, Marc Schoenauer

Topological Optimum Design using Genetic Algorithms

Couro Kane & **Marc Schoenauer**
Centre de Mathématiques Appliquées
Ecole Polytechnique
91128 Palaiseau Cedex - FRANCE
{Couro.Kane,Marc.Schoenauer}@polytechnique.fr

Published in **Control and Cybernetics**, Vol. **25** No. 5 (1996)

Abstract

Structural topology optimization is addressed through Genetic Algorithms: A set of designs is evolved following the Darwinian *survival-of-fittest* principle. The goal is to optimize the weight of the structure under displacement constraints.

This approach demonstrates high flexibility, and breaks many limits of standard optimization algorithms, in spite of the heavy requirements in term of computational effort: Alternate optimal solutions to the same problem can be found; Structures can be optimized with respect to multiple loadings; The prescribed loadings can be applied on the unknown boundary of the solution, rather than on the fixed boundary of the design domain; Different materials as well as different mechanical models can be used, as witnessed by the first results of Topological Optimum Design ever obtained in the large displacements model.

But these results could not have been obtained without careful specific handling of the specific aspects of topological genetic optimization: First, specific genetic operators (crossover, mutation) were introduced; Second, special attention was paid to the design of the objective function; The nonlinear geometrical effects of the large displacement model lead to non viable solutions, unless some constraints are imposed on the stress field.

1 Introduction

Since the seminal work of Holland [27] and the comprehensive study of Goldberg [21], Genetic Algorithms (GAs) have gradually been recognized as powerful stochastic optimization algorithms. More recently, the initial framework of fixed length bitstrings has been widened to other search spaces [42, 39, 7, 6], emphasizing the need for problem-specific modifications of the basic algorithms. The field of *Evolutionary Computation* covers all alternate evolutionary algorithms [49, 19, 4].

The main interest of stochastic methods in Engineering Sciences is to break the limits of standard deterministic methods in many optimization problems: when the search space involve both discrete and continuous domains (e.g. for the optimal design of truss structures); when the objective function or the constraints lack regularity; or when the objective function admits a huge number of local optima. In counterpart, stochastic methods are computationally expensive: GAs for instance, are slower than classical optimization methods by about one or two orders of magnitude, when comparison is possible, i.e. when classical methods apply.

This paper focuses on applying GAs to some well-studied problems in mechanical engineering, namely the structural topology optimization of cantilever plates. Section 2 briefly introduces the mechanical problem, and surveys previous works in structural topology optimization, discussing their advantages and limitations;. Section 3 presents the broad lines of Genetic Algorithms in the historical bitstring perspective, for the sake of completeness. Section 4 is devoted to the construction of the problem-specific Genetic Algorithm. The chosen representation of structures is discussed. The standard genetic operators are then tailored for topology optimization. Finally, the design of the fitness function is thoroughly discussed. Experimentations in the linear elasticity case are presented in section 5: After validation results on the standard cantilever plate problem, this problem is modified to highlight the ability of the GA-based approach to handle problems having multiple optimal or quasi-optimal solutions. Multi-loading results are then presented on the problem of optimizing the structure of a bicycle, and finally, the problem of optimizing the shape of an underwater dome, involving loading on the unknown boundary of the structure, is addressed and solved. Section 6 presents the first results of nonlinear topological optimization, in the context of linear elasticity with large displacements. The nonlinear geometrical effects clearly show the need to take into account the stress field in the fitness function to obtain realistic solutions.

The out breaking results on Topological Optimum Design presented in this paper have been obtained during the first author's PhD work. Hence, the reader is referred to the PhD dissertation [31] for all technical details.

2 Mechanical Background

2.1 The problem

The general framework of this paper is the problem of finding the optimal shape of a structure (i.e. a repartition of material in a given *design domain*) such that the mechanical behavior of that structure meets some requirements (e.g. a bound on the maximal displacement under a prescribed loading). The optimality criterion is is here the weight of the structure, but it could involve other technological costs.

Throughout this paper, except in Section 6, the mechanical model will be the standard two-dimensional plane stress linear model, and only linear elastic materials will be considered (see [16, 30]). The effects of gravity are neglected.

One of the well-known benchmark problem of Optimum Design is that of the cantilever

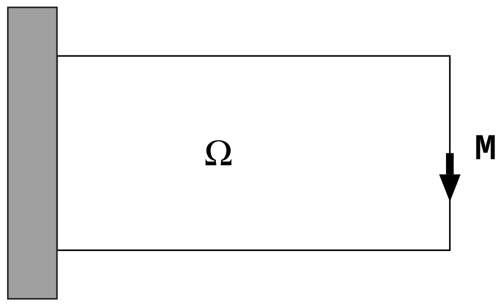


Figure 1: The standard 2×1 cantilever plate.

plate: a rectangular plate is fixed on the left vertical part of its boundary (both displacements are set to 0), and the loading is made of a single force applied on the middle of its right vertical boundary. Figure 1 shows the design domain for the 2×1 cantilever plate problem.

2.2 Related Works

The main trends in structural optimization can be sketched as follows.

A first approach is that of *domain variation* [11] (also termed *sensitivity analysis* in Structural Mechanics). It consists in successive small variations of an initial design domain, and is based on the computation of the gradient of the objective function with respect to the domain. This approach has two major defects: first, it requires a good initial guess, as it demonstrated unstable for large variations of the domain; second, it does not allow to modify the *topology* of the initial domain (e.g. add or remove holes).

A more recent approach to *topology design*, introduced in [9], is that of homogenization; it consists in dealing with a continuous density of material in $[0, 1]$. In the end of this deterministic optimization, the current density is forced toward value 1 or 0, that respectively stands for material or void. However, this approach requires the design of the homogenized operator, as thoroughly described in [2], and is insofar limited to the linear elasticity case. Moreover, it cannot address loadings that apply on the actual boundary of the shape to be determined (e.g. uniform pressure, as described in section 5.4), and hardly handles optimization for multiple loadings.

A possible approach to overcome these difficulties of topological optimum design is to use stochastic optimization methods, such as simulated annealing [34] and genetic algorithms [27]. Both methods have been successfully applied to other problems of structural optimization: in the framework of discrete truss structures, for cross-section sizing [23, 25, 36, 48] among others, as well as for topological optimization [26, 24, 53] and for the optimization of composite materials [35].

More recently, some problems of structural components optimization have been addressed by stochastic methods: Simulated Annealing is used to find the optimal shape of the cross-section of a beam, a simple problem on which interesting theoretical results are proved in [3, 20]; and GAs are used to solve the cantilever problem presented in Section 2.1 in [28, 15]. However, the work presented in this paper goes further than these latter works: The geometrical constraints are weakened, increasing the range of possible solutions (Section 4.5.1); The fitness function is carefully designed, and a fine control of the mechanical behavior of the solution is thus possible (Section 4.5); Last, this paper is not limited to feasibility results on problems that can be solved by the homogenization method, as demonstrated by the breakthrough results presented in Section 5 and 6.

3 Genetic Algorithms

This section gives the broad lines of basic GAs; the reader is referred to [21] or [39] for further details.

3.1 Historical GAs

Given a search space E and a *fitness* function F defined from E onto $\mathbb{R}^+$, GAs evolve a set of p *individuals* (points of E), termed *population*. This evolution crudely mimics the Darwinian evolution: according to the Darwinian *survival-of-the-fittest* principle, the fittest individuals, i.e. near-optimal points of fitness function F will appear in population P_i for some i.
The basic step in GAs, called *generation*, is the transformation used for population P_i to give birth to population P_{i+1}. This transformation involves four steps :

- **Evaluation**: the value of the fitness of all individuals on the current population is computed. Note that this step involves p independent computations that can easily be parallelized.
- **Selection** builds population P_i' by copying elements of P_i ; the number of copies of an individual increases with its fitness, the total number of elements in P_i' being same as in P_i.
- **Crossover** applies on population P_i' to build population P_i". From two individuals x and y in P_i', crossover builds two offsprings x' and y' with probability p_c (p_c usually varies from .2 to 1.). When considering a bitstring representation ($E = \{0,1\}^N$), a crossover c can be represented as a bitstring itself, $c = (c_1, \ldots c_N)$:

$$\begin{matrix} x_1 & .. & x_N \\ y_1 & .. & y_N \end{matrix} \rightarrow \begin{matrix} x_1\prime & .. & x_N\prime \\ y_1\prime & .. & y_N\prime \end{matrix}$$

$$\text{with } x_i\prime = \begin{cases} x_i & \text{if } c_i = 1 \\ y_i & \text{if } c_i = 0 \end{cases}$$

$$\text{and } y_i\prime = \begin{cases} y_i & \text{if } c_i = 1 \\ x_i & \text{if } c_i = 0 \end{cases}$$

  Most authors only consider *one-point crossovers*, corresponding to masks (1..10..0), or *two-point crossovers*, corresponding to masks (1..10..01..1). The general case represented here is called *uniform* crossover [52].

- **Mutation** applies on population P_i" to build population P_{i+1}. Mutation transforms an individual x in P_i" into an offspring x' ; when considering a bitstring representation, a mutation can similarly be represented by a bitstring $m = (m_1, \ldots m_N)$:

$$x_1 \;\; .. \;\; x_N \;\rightarrow\; x_1\prime \;\; .. \;\; x_N\prime$$
$$\text{with } x_i\prime = \begin{cases} 1 - x_i & \text{if } m_i = 1 \\ x_i & \text{if } m_i = 0 \end{cases}$$

The probability for m_i to take value 1 is noted p_m (p_m usually varies from 10^{-2} to 10^{-4}).

3.2 The representation issue

One of the main difficulty arising when applying GAs to some optimization problem is the choice of the search space. Two different spaces have to be considered. While the fitness function is defined on the *phenotype* space, genetic operators usually apply in the *genotype* space, be it the space of bitstrings, as described above, or any other space, as in modern Evolutionary Computation works. The *representation* of a phenotype in the genotype space involves a mapping (or *coding*) which can cause a loss of information (it is generally not bijective, not isometric, ...). The choice of the genotype space is therefore related to a compromise between the simplicity of the coding and the possibility to design useful genetic operators in the genotype space. Whereas standard GAs emphasize the use of fixed bitstrings as genotypes, other trends of Evolutionary Computation try to use the same space for both the genotypes and the phenotypes, the effort being then put on the design of genetic operators. A thorough discussion on the topic can be found in [19]. Section 4 will provide an instance of such a situation.

3.3 Pros and Cons

When considered as function optimizers (which they are not in the first place [18]), the main advantage of GAs is to be a zero-th order method: The only prerequisite is to be able to compute values of the objective function, and of possible constraints of the problem. Furthermore, no regularity is required, neither on the functions nor on the search space. GAs can hence be used for continuous parameter optimization, for totally discrete problems as well as for mixed integer-continuous optimization [38, 5], provided genetic operators are defined on the search space. However, the design of good genetic operators is still a matter of experience, and a posteriori numerical experimentation remains the only possible validation, though some general guidelines have been stated [42]. Section 4.4 will give examples of specific operators, justified by the performance of the resulting algorithm.

On the other hand, it is well-known that the main drawback of GAs is their slowness. In particular, when both a GA and some deterministic method (e.g. a gradient method) can be used on the same problem, the latter is usually some orders of magnitude faster [48]. Parallelization certainly brings a partial answer to that issue. However, it only distributes, and does not reduce, the CPU requirements needed for a successful GA-based optimization problem solving.

Moreover, GAs involve numerous user-defined parameters that have to be tuned carefully to get the best out of the algorithm. Here again, that tuning can only rely on the experience of

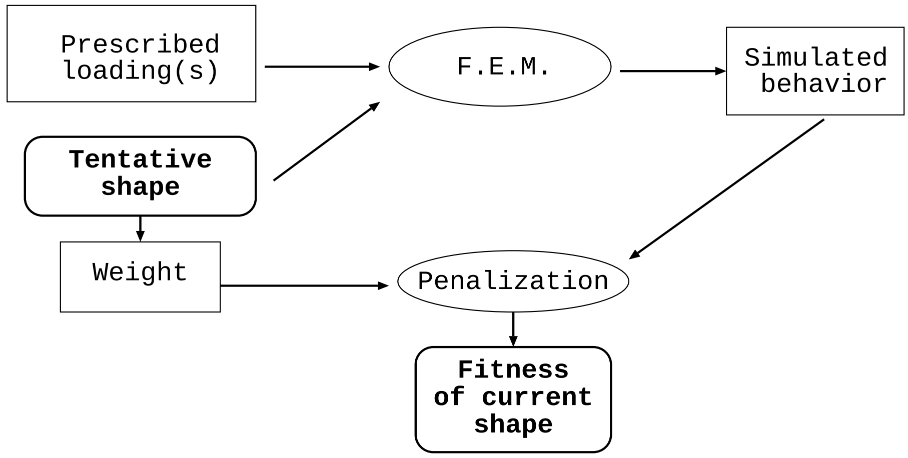


Figure 2: A tentative fitness for genetic optimum design

the programmer, and on the results systematic trial-and-error experiments.

Nevertheless, these drawbacks of GAs are negligible whenever they are the only way to a solution of the problem at hand, handling problems beyond the limits of other optimization methods. The following of the paper is devoted to such successful applications of GAs.

4 Genetic Optimum Design

This section will detail the implementation of GAs on the problem of Optimum Design presented in section 2.1, describing all necessary steps toward the successful implementation of GAs on this problem.

4.1 Tentative fitness

As emphasized in section 3.3, GAs only require values of the objective function. Hence, the broad lines of the implementation of GAs for the problem of Optimum Design described in section 2.1 can be represented as in Figure 2: The mechanical behavior of a structure can be numerically simulated by some Finite Element Method. From that simulation, and from the weight of the structure, a function to optimize can be designed (handling the constraints on the mechanical behavior by using penalty functions, for instance). This function can be used as the fitness function of a Genetic Algorithm.

However, the main question remains the choice of a search space: the target space is that of partitions of the design domain into two subsets (material and void). However, this general space is far too large (for instance, for obvious mechanical reasons, only structures with continuous boundaries between material and void need to be considered). Unfortunately, there is no natural *representation* for partition of continuous domains.

4.2 A priori Discussion for Shape Representations

A theoretical framework has been developed by Ghaddar, Maday and Patera [20] in the same context of Structural Optimum Design, but on the problem of optimizing the cross-section of a beam submitted to a bending momentum. The search space is restricted to partitions with polygonal boundaries. Theoretical results of existence and uniqueness of a solution are proven, approximation spaces are introduced and corresponding approximation results are obtained. Though the objective function considered in this paper is quite different from the one in [20], the same class of search space will be used here.

However, a significant difference between the objective functions in [20] and the one to be used here is that the topological Optimum Design problem requires some Finite Element Analyses (FEAs) on the direct problem to compute the fitness of a possible solution (i.e. a given repartition of material in the design domain), as shown by Figure 2, and detailed in forthcoming section 4.5. It is well-known that meshing is a source of numerical errors [17]. Hence, when comparing two structures of different shapes, using a fitness function based on the outputs of two FEAs performed on different meshes is bound to failure, at least when the actual differences of behavior becomes smaller than the unavoidable numerical noise due to remeshing. The use of the same mesh for all FEAs, at least inside the same generation, is thus mandatory in order to obtain significant results.

4.3 The bitarray representation

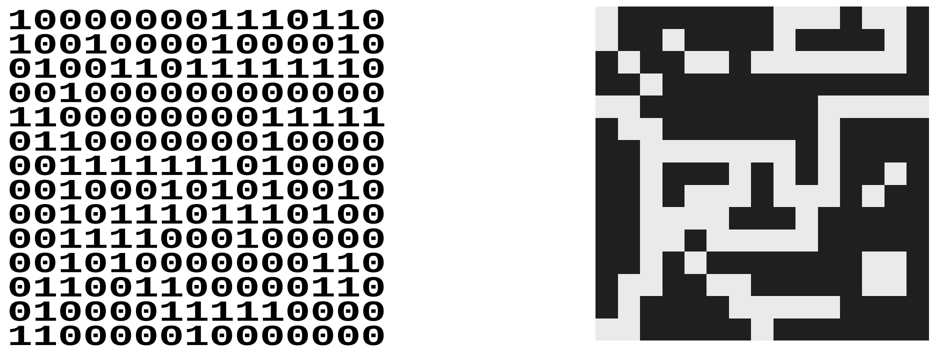
Once the decision to use a fixed mesh has been taken, and with even very little knowledge of GAs, the straightforward representation for a partition of the design domain is that of bitstrings: each element of the fixed mesh belongs to either one of the subsets of the partition, which can be symbolically labeled 0 or 1. Figure 3 is an example of a "chromosome", together with the corresponding repartition of the material, for a regular mesh of the design domain into quadrangles.

At first sight, the resulting representation can be viewed as a bitstring, and the Optimum Design problem seems to have been brought back into the historical framework of GAs [27, 21], where the search space is the space of fixed-length bitstrings. Hence, all previous works using Genetic Algorithms on Optimum Design problems did use that representation [28, 15].

However, this simple approach is far from optimal, as will be shown in next section.

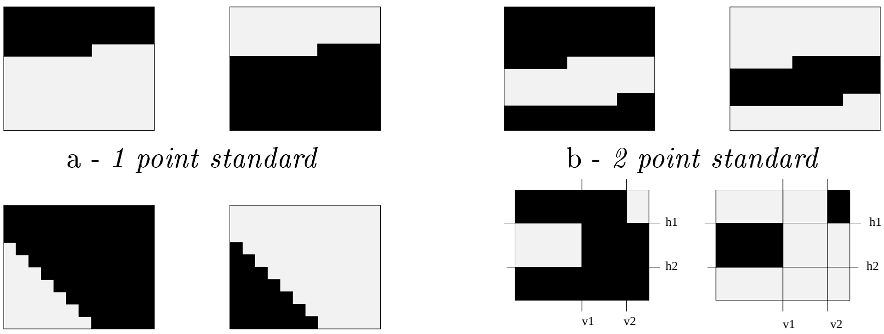
4.4 Bitarrays are not bitstrings

This section is devoted to the study of the crossover operators for the representation of shapes given in section 4.3. All crossover operators presented in this section will be symbolically presented on Figure 4: two offsprings of a black parent and a grey parent are plotted. The color of the bits of the children only tells which parent this bit comes from, regardless of its actual value.



a - *The bitarray genotype.* b - *The phenotype, or repartition of void/material.*

Figure 3: Bitarray representation: GA applies on the bitarray genotype (a), while the fitness is computed on the phenotype (b).



a - *1 point standard* b - *2 point standard*

c - *diagonal* d - *2-block*

Figure 4: Examples of "Black and white" offsprings from different crossover operators.

4.4.1 Geometrical bias of one-dimensional crossover operators

In order to evaluate the bias induced by handling a bitarray as a bitstring, the standard (bitstring) crossover operators are first considered.

One-point and two-points crossovers are geometrically biased, as they can only exchange horizontal bands of genetic material between the parents. Figures 4-a and 4-b symbolically witness that phenomenon. A detailed *Schema analysis* can be found in [33, 31], emphasizing this bias. On the opposite, the uniform crossover (random mixture of both parents' genetic material not presented in Figure 4) does not suffer from such a bias.

Nevertheless, two specific two-dimensional operators are introduced in next section to address this issue.

4.4.2 Specific two-dimensional crossover operators

Diagonal crossover

The basic idea of diagonal crossover is to generalize the popular one-point crossover to the

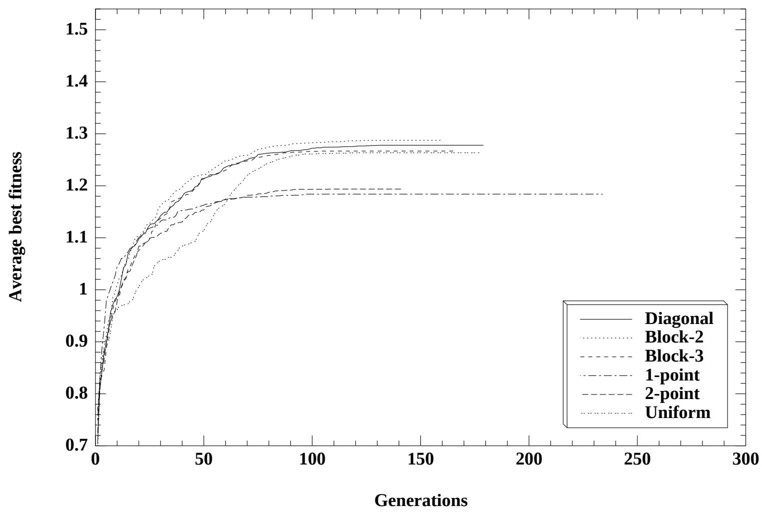


Figure 5: Performances of crossover operators without mutation

two-dimensional case. As shown on Figure 3-c, a randomly selected straight line separates the rectangle in two parts which are exchanged between both parents.

The block crossover

First introduced in [28], the idea of the block crossover is to cut the whole two-dimensional domain by two horizontal lines and two vertical lines and to exchange some of the large blocks defined by these lines. The values v_1 and v_2 (respectively h_1 and h_2) are chosen uniformly, and the 2 (or 3) blocks to be exchanged are selected randomly. Figure 3-d shows an example of 2-block crossover.

4.4.3 Experimental Comparison of Crossover Operators

These crossover operators have been experimentally compared on the problem of the optimization of the cross-section of a beam [20, 33]. Figure 5 shows the results obtained without any mutation, and with a crossover rate of 1.

The first conclusion that can be drawn from Figure 5 is the ineffectiveness of the one-dimensional crossover operators, significantly outperformed by both uniform and two-dimensional crossovers.

The second remark is that the uniform operator performs poorly in the early stage of the evolution, before catching up in the late generations. This fact is confirmed by all other experiments. This phenomenon is fairly general: in the beginning of the evolution, the uniform crossover equivalently disrupts emerging schemas, while all other crossovers (even one dimensional) preserve some large areas. These disruptive effects do not occur in the end of evolution, since neither kind of crossover does disturb regions where convergence already occurred: If the bits of both parents at a given position are the same, crossing over has no effect on those bits. In the meantime, the one-dimensional crossovers fail to beneficially exchange potentially good vertical parts (e.g. long vertical bars).

In all experiments of section 5 and 6, the block-3 crossover was used (it demonstrated slightly better performances than diagonal, block-2 and uniform crossovers) [31].

4.4.4 Mutation operators

The standard bit-flip mutation applies on bitarrays without inducing any geometric bias. Nevertheless, two other directions are explored regarding the mutation operator of bitarray shapes. The first one is problem-independent, and uses statistics on the whole population to keep some genetic diversity. The second mutation operator, purposely devised for the problem of shape optimization, favors small modifications of the boundary of the shape.

Population-based mutation
This mutation operator, first defined in [10] aims at preventing the premature convergence of the population. The probability of flipping a given bit is adjusted by considering all bits in the same position in the whole population. More precisely, the probability p_i for a given bit i to be flipped depends on the mean value m_i of this bit over the population: if this bit takes a uniform value ($m_i = 0$ or $m_i = 1$), the probability to mutate is set to a high value p_{max}; on the opposite, it is set to a low value p_{min} if there is about the same proportion of 0s and 1s ($m_i = 0.5$). The probability of flipping a bit is then a parabolic function of the mean value m_i between these points. This operator thus imposes high values of mutation rate at positions that have already converged.

Boundary mutation
The underlying idea of the boundary mutation is that, for a given topology, i.e. a given number of "holes" in the shape, the optimal design for that topology can be found by slightly moving the boundary. The boundary mutation is defined such that boundary bits, i.e. bits having one edge on the boundary of the connected component of the shape, are given higher probability ($p_{B,max}$) to be flipped than the other bits ($p_{B,min}$).

The population-based mutation performed slightly better than the standard bit-flip mutation (though many trials did no show significant differences), and therefore was used in all experiments presented below. The boundary mutation will only be used in the end of the evolution process, to fine-tune the solution. More details can be found in [33, 31].

4.5 Fitness computation

A given individual is evaluated in a 2-step process. The first step is a geometrical analysis of the repartition of the material in the design domain, followed by some Finite Element Analyses on the actual structure defined by this partition.

4.5.1 Geometrical analysis

Some *seed* material is imposed at the point(s) where dynamical loading is applied. The connected component containing that seed is computed. Grid elements are connected if and only if they share an edge. Note that no seed material is prescribed on the part of the boundary where the plate is fixed, as it is in [15]. The optimization process "chooses" where to hang the structure on the fixed vertical boundary. This allows for a greater flexibility in solving the optimization problem, as witnessed in the range of alternative solutions proposed for the modified cantilever plate problem (see Section 5.2). On the other hand, more structures are likely to be disconnected from that fixed boundary, leading to a mechanically ill-posed problem: such structures are arbitrarily assigned zero fitness, and are therefore eliminated by the next Darwinian selection.

Another difficulty arises when considering the different connected components of the structure: Only material actually connected to both vertical boundaries takes part participate to the mechanical behavior of the structure. The disconnected parts have to be removed before the the FEM is called (see the example of Figure 3-b). But there are different possible ways to handle this difficulty from a GA point of view:

- The disconnected parts can be simply removed during the computation of the fitness. However this introduces a fairly high level of degeneracy in the representation: Many genotypes correspond to the same phenotype, and hence have exactly the same fitness. And, as pointed out in [43], this is not a desirable feature, as there is no preferred direction for the GA to go on these large plateaus of fitness landscape.
- The disconnected parts can be definitely removed from both the phenotype and the genotype, in a Lamarckian-like way. This is amenable to the so-called *repair* technique used in genetic constraint handling [40]. But these disconnected parts possibly are valuable genetic material (like the dominated part of diploid chromosomes), at least in the beginning of evolution: This strategy generally leads rapidly into some local minima. Experimental results account for that statement.
- The fitness can be modified to slightly favor the disappearing of such disconnected material: the Finite Element Method is run without the disconnected parts, and some penalty term is added, relative to the amount of disconnected material, in order to guide the algorithm toward better solutions.

This latter possibility will be used throughout this paper. Next section will give the precise formulation of the fitness (equation 1), including the penalization term to cope with the unconnected material.

4.5.2 The penalized fitness

Once the component connecting the seed(s) and the fixed boundary has been computed (if any), one FEA is performed for each one of the considered loading cases (generally one, except in multi-loading problems, see Section 5.3), using the same regular quadrangular mesh that supports the representation of the individuals. The FEM tool is detailed in [30]. Note the actual material boundary is used, in contrast with both [28] and [15] in which the FEA was done on the whole design domain, after assigning a very low Young modulus to void elements. Though the results of the analysis does not differ significantly from one method to the other, computing the actual boundary allows to take into account loading applied on this boundary (e.g. pressure, heat exchanges) as demonstrated in section 5.4.

Different criteria have been used in previous works on Optimum Design. The objective function for the homogenization method is based on the *compliance* of the structure, computed as the work of external loadings [2]. The main advantage of this objective function is to be differentiable. However, it does not allow a precise control of the mechanical behavior of the solution. As GAs do not require differentiable fitness function, the compliance will not be considered here. It has however been considered in the GA framework [33, 31] in order to make more precise comparisons between the homogenization method and the GA-based method proposed here [32, 31].

Yet another approach is to try to maximize the *stiffness* of the structure [15, 14]. Here again this does not allow a fine control of the desired behavior of the solution. Moreover, comparative experiments [31] demonstrate that using the stiffness as fitness function generally leads to heavier solutions without much improvement of the mechanical properties.

Hence, the following of the paper will address the problem described in section 2.1 of the minimization of the weight of the structure subject to some upper limits on the maximal displacement of the structure when subject to prescribed loadings.

This problem is a constrained optimization problem. Many methods have been designed to constrained evolutionary optimization (see [47] for a survey of such methods). However, the standard penalization method remains the simplest one to use, as it only requires a modification of the objective function. The drawback lies in the difficulty of adjusting the penalization parameters: Section 4.6 below will discuss that issue in in the context of this paper. Whatever the choice of penalization parameter, the formulation for the penalized fitness is the following:

$$\mathcal{F} = \frac{1}{A_{con} + \varepsilon A_{dis} + \alpha(D_{Max} - D_{Lim})^{+}} \tag{1}$$

where
A_{con} and A_{dis} are the area of the connected and disconnected material in the repartition (see section 4.5.1 above);
D_{Lim} the imposed limit value for the displacement;
D_{Max} is the maximal displacement of the structure when the prescribed loading is applied (computed using the FEM);
ε and α are positive user-supplied penalty parameters (a^{+} denotes the positive part of a).

Note that ε is also a penalization parameter, but whose value does not influence much on the performance of the algorithm: it was fixed to 0.1 throughout the experiments described in the following sections.

4.6 Adjusting the penalization parameter

As quoted above, using penalization method to handle constraints raises the difficulty of tuning the strength of that penalization, i.e. in the case of fitness function 1, adjusting precisely the parameter α.

The drawbacks of a fixed value of α are well-known (see [41] for a more thorough discussion):

- A small value of α may result in an optimal solution that violates the constraints;
- A large value of α ensures that the constraints will be strictly met, but forbids exploration and short-cuts in infeasible regions that might be essential for the overall success of the algorithm, at least during the beginning of evolution.

Hence a natural idea is to use for α a dynamic schedule, starting from small values, and thus letting the population explore even infeasible regions, and gradually enforcing the penalization by increasing α such that the constraints are finally met by the whole population. Two different solutions to achieve this goal have been tried on the problem of Optimum Design.

The first method uses a prescribed geometrical evolution schedule for α, and is termed *exogenous* scheme (similar dynamic penalty terms have been proposed in [29]). The value of α is increased by some multiplicative factor β every M generations. Hence the value of α at generation i is given by

$$\alpha_i = \alpha_0 \beta^{[\frac{i}{M}]}$$

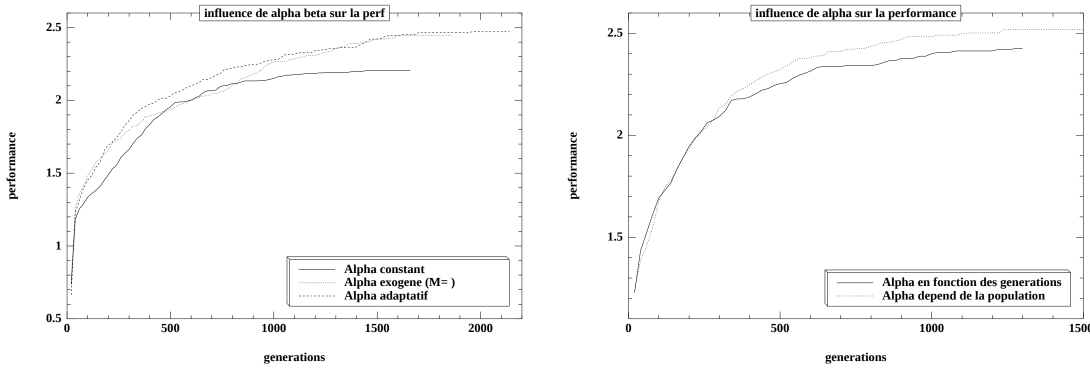
Typical values of M and β are 10 and 1.001, while the initial value α_0 is computed from the average weights and violations of the constraints in the initial population.

The second scheme is an *adaptive* scheme for α, that uses the information contained in the current population to compute further values, in the line of the adaptive penalties proposed in [8, 50]. Some $\bar{\alpha}_i$ is computed from the actual average weights and violations in population i, and α is set to $\bar{\alpha}_i$ if $\bar{\alpha}_i$ is greater than the current value of α, to ensure monotonicity of α.

Comparative experiments between the fixed scheme, the exogenous geometrical scheme and the adaptive scheme were performed on different instances of the cantilever problem. The first strong conclusion is that the fixed scheme is constantly outperformed by both dynamic schemes. A typical example of such a situation is shown on Figure 6-a. Note that the plots of both dynamic schemes are hardly distinguishable. But another typical result is shown on Figure 6-b, demonstrating a slight improvement of the adaptive scheme over the exogenous scheme. However, the latter was used in all experiments of next sections, as the being much simpler to tune.

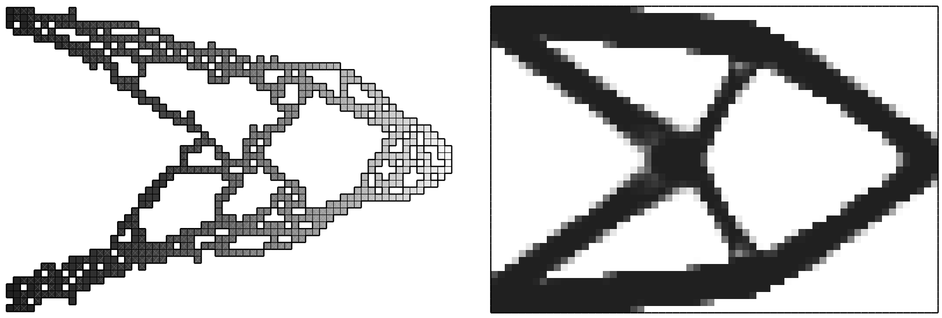
5 Results in Linear Elasticity

This section presents results obtained using the algorithm detailed in the preceding section in the framework of linear elasticity. Hence, as the first results (Section 5.1) deal with the standard cantilever problem, they could have been obtained – much more rapidly – by the homogenization method. But after these validation results, the limits of homogenization methods are broken, and the presented results deal with multiple quasi-optimal solutions (Section 5.2), multi-loadings cases (Section 5.3) and finally address the problem of the underwater dome, involving loading on the unknown boundary (Section 5.4).



(a) (b)

Figure 6: Comparative results (averaged over 15 independent runs) of different schemes for the penalization parameter α. (a) The fixed scheme is outperformed by both dynamic schemes. (b) The adaptive scheme sometimes slightly supersedes the exogenous geometrical scheme.



(a) (b)

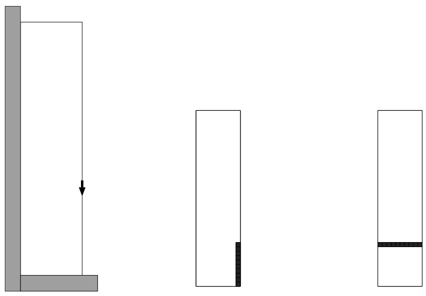
Figure 7: Results on the 64×44 mesh for the 2×1 cantilever plate problem. (a) Solution of the GA-based method. (b) Solution of the homogenization method.

5.1 Validation Results

Figure 7-a is a typical result of a GA run for the 2×1 cantilever plate, discretized according to a 64×44 regular mesh. The population size for all runs is 125, and the number of generations arbitrarily fixed to 2000. One run thus require about 150,000 FEM analyses, taking approximately 24 hours of a powerful HP workstation. The genetic operators are applied with the following probabilities: 0.6 for the block crossover (see Section 4.4.2), 0.1 for the population-based mutation, with p_{min} and p_{Max} set to 0.0001 to 0.01 (see Section 4.4.4). All these parameters were adjusted after exhaustive tests, and details can be found in [33, 31].

The result of Figure 7-a is to be compared to the result of homogenization of Figure 7-b. Both structures look alike, they have almost the same weight and same maximal displacement: these results validate the GA approach.

Moreover, the GA solution exhibits numerous small holes (of size one element) in the structure. This seems to indicate that the GAs somehow tries to approach the actual optimum, which



(a) The problem (b) (c) Two optimal solutions

Figure 8: The 1×4 modified cantilever plate.

is known to lie in the homogenized space [2]: The homogenized optimal solution can be viewed as the limit of a structure in which infinitely many holes of infinitely small size are drilled. Note however that this could have been avoided by imposing some lower bound on the thickness and the co-thickness of the structures, as in the theoretical study developed in [20].

Of course, complete validation of a stochastic method involves to check its robustness with respect to the random initialization. Different runs of the same program with different random have been performed: All lead to similar structures, but slight differences could be observed in their appearances, though their weights and maximal displacements were very close. GAs are known to be able to find quasi-optimal solutions on multi-modal problems [22, 37]. This latter remark suggested to consider a problem with known multiple solutions.

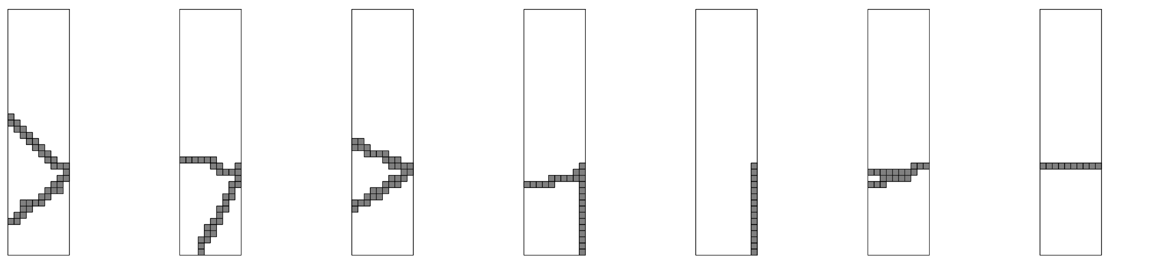
5.2 Multiple quasi-optimal solutions

The problem considered in this section is a modification of the standard cantilever plate problem to ensure the existence of multiple quasi-optimal solutions: the 1×4 cantilever plate is discretized according to a 10×40 mesh and both its left and bottom boundaries are fixed (see Figure 8-a).

Depending on the height of the point where the loading is applied and on the constraint on the displacement, the same problem can have multiple solutions. A simple example of such situation is given in Figure 8 (b) and (c): if the loading is applied at height 10, and provided that the displacement constraint is large enough, both structures (b) and (c) are optimal solutions.

And indeed, as no material is prescribed on the fixed boundary (see Section 4.5.1), the GA-based algorithm was able to find both solutions.

Moreover, if the height of the loading point is fixed, and the displacement constraint D_{Lim} is gradually relaxed, different quasi-optimal solutions exist for some ranges of D_{Lim}. The GA method was then able to find out different multiple solutions, as demonstrated in Figure 9. Here again only the most significant results are shown, for different values of D_{Lim}, while the loading is applied at height 15. The GA parameters are those of section 5.1, except for the population size (100) and the maximal number of generations (500).



0.32	0.4		0.42	7.66		46.6
0.294	0.3927	0.3919	0.3415	7.66	6.743	46.64
0.38	0.33	0.34	0.26	0.15	0.19	0.1

Figure 9: Optimal structures on problems with multiple solutions. $\mathcal{F}$ is applied at height 15. The figures below the structures are respectively D_{Lim} (the limit on the displacement), D_{Max} (the maximal displacement of the structure) and A (the area of the structure).

5.3 Multiple loadings

All problem addressed in the preceding sections considered constraints on the mechanical behavior of the structure in a single case of loading. But actual structures in real-world problem are generally subject to different loadings, and hence should be optimized while taking into account more than one loading case.

5.3.1 Modified fitness

Taking into account this new situation is straightforward in the context of GA-based Optimum Design: for each structure, one Finite Element Analysis is performed for each of the loading cases, and the fitness function defined by equation (1) is modified to agglomerate the constraints corresponding to the L loading cases:

$$\mathcal{F} = \frac{1}{A_{con} + \varepsilon A_{dis} + \sum_{i=0}^{i=L} \alpha^i (D^i_{Max} - D^i_{Lim})^+} \tag{2}$$

where
A_{con}, A_{dis} and ε are as in equation 1;
D^i_{Lim} is the limit value for the displacement for loading case i;
D^i_{Max} is the maximal displacement of the structure when loading i is applied;
α^i is the penalization parameter corresponding to the i^{th} constraint; all α^i are adjusted as described in Section 4.6.

Of course, the computation of the fitness of a single structure requires in that context L FEAs, and the computational time is therefore L times larger than when a single loading is considered.

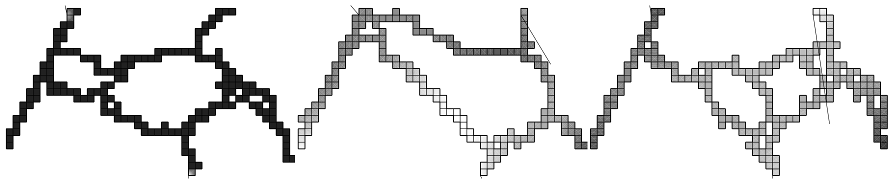


Figure 10: The three loading cases for the optimization of the structure of a bicycle, together with the solution of the corresponding single-loading optimizatio problem. (a) Steady ground. (b) Heavy slope (no force on the saddle). (c) Sitting up-hill position.

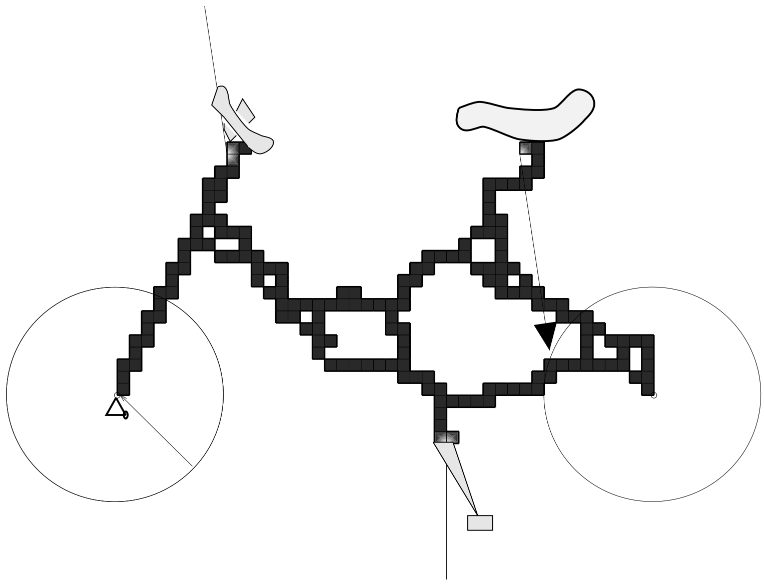


Figure 11: The multi-loading bicycle

5.3.2 An example

An example of such a situation is given by the structure of a bicycle: the forces applied to that structure are very different depending on the position of the rider, which in turn heavily depends on the slope of the road. For the sake of simplicity, only the following three different cases have been considered: on flat landscape, the greatest force is applied on the saddle, on up-hill ground, the rider pushes hard on the pedals and pulls on the handlebars and on steep roads, he does not sit on the saddle any more, pushing extremely hard on the pedals.

Figure 10 shows the result of the three single-loading optimizatio problem, while Figure 11 is the solution of the Three-loadings optimization problem. The advantages of solving the multi-loading problem appear clearly when comparing these two figures, as the resulting multi-loading solution differ significantly from any of the three solutions of the single-loading problems.

Another approach to the same problem could be to use multi-objective optimization methods. Specific modifications of GAs were made to handle multi-criterion optimization, taking advantage of the ability of GAs to find mutiple optima [44, 51]. The aim of the resulting algorithm is to sample the Pareto set. To our knowledge, such approach was not tested yet on the multi-loading Topological Optimum Design problem.

5.4 Loading on the unknown boundary

All problems addressed up to this point dealt with fixed loading, independent of the structure at hand: the loading(s) was (were) applied on the boundary of the design domain. There are situations, however, where some loading is applied on the actual boundary of the structure, which can be different from the boundary of the design domain. This is the case if one some uniform pressure is applied on the upper boundary of a civil engineering structure, or, in a different context, when optimizing the shape of heaters, which implies heat transfers through the unknown boundary of the solution [20].

This situation is of course intractable for the homogenization method, as the boundary of the target structure is not defined precisely before the projection step (nor sometimes is it precisely defined after that step either). From a GA perspective, this situation does not differ from the situation where all loading are applied on the boundary of the design domain. The boundary of any structure in the population is well-defined, and any loading can be applied there onto. Hence no modification of the algorithm is required to handle this case.

Consider the problem of optimizing the shape of an underwater dome: a uniform pressure is applied on the upper boundary of all structures. Some material is imposed at both ends of the lower boundary of the design domain, as well as at some point above the middle, to ensure that a minimum height of the structure[1].

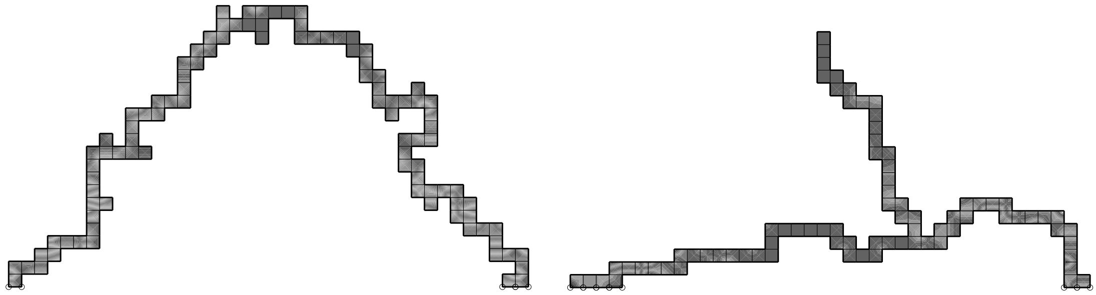
Figure 12 shows the resulting structures, for both a small pressure (12-a) and a large pressure (12-b). The latter structure is indeed a better solution than what was expected (a structure looking like a reinforced 12-a structure) as is certainly is lighter: This only shows that the material requirements that were imposed here were not the right ones: some void area should have been imposed in the middle of the design domain.

6 Nonlinear Geometrical Effects in Topology Optimization

This section considers standard plane stress problems in the context of large displacements. The material still obeys a linear law (the extension to any other constitutive law is straightforward), but the nonlinear geometric effects due to the large displacement hypothesis are taken into account. A thorough description of the theoretical model together with the numerical algorithm can be found in [16]. Details on the numerical model and implementation used here can be found in [30].

The first experiments used the penalized fitness function described by equation (1). The initial idea was to use different loadings $\mathcal{F}$, with a fixed ratio $\mathcal{F}/D_{Lim}$ (D_{Lim} is the constraint

[1]Otherwise, the optimal solution is the flat structure along the lower boundary, which indeed has a small displacement. Note that, due to an error, one run was performed without this material requirement above the ground, and that an almost flat solution was found by the algorithm [31]



(a) (b)

Figure 12: Results for the optimization of the underwater dome, with uniform pressure on the upper boundary. (a) For small pressure. (b) for large pressure.

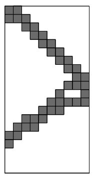
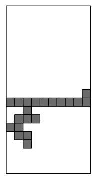
		
D_{Max}	0.022607	0.0199
σ_{Max}	0.076	0.77
$Area$	0.41	0.20
	(a): small displacements.	(b): large displacements.

Figure 13: Optimal (disastrous) designs: Only displacement constraints were considered. $\mathcal{F} = 0.009$ and $D_{Lim} = 0.02285$.

on the displacement). In the purely linear case, all such problems are of course equivalent.

A typical result of the GA based optimization in the large displacement context is shown in Figure 13-b, together with the result on the same problem in the pure linear model (Figure 13-a): one does not need to be a mechanical expert to see that such a structure is a disaster, from a mechanical point of view. And different values of $\mathcal{F}$ and D_{Lim} did produce similar results. And a closer look at the maximal stress of the solutions σ_{Max} confirmed the existence of a problem there: The value of σ_{Max} in Figure 13-b are far too high.

First, the large displacement model can have different solutions for the same loading, some with higher stress than others. Second, the stress field itself, on a rough domain like the structures obtained for the standard cantilever plate problem, and with such coarse discretization, presents some singularities.

As a matter of fact, Table 1 gives some idea of the nonlinear geometrical effects in the case of two simple structures, the perfect ">" shape and the straight beam; both the displacement and the maximum stress are given for varying loads. The displacement is what was expected, but the stress does present weird values. Note that this phenomenon depends on the numerical model [16, 30].

	">" shape		straight beam	
Load	D_{Max}	σ_{Max}	D_{Max}	σ_{Max}
$9 10^{-6}$	0.0002360	0.0005968	0.04161	0.03035
$9 10^{-5}$	0.00235	0.0059	0.34499	0.28344
$9 10^{-4}$	0.02286	0.05312	0.80763	1.272
$9 10^{-3}$	1.06731	10.99	1.08208	1.11683
$9 10^{-2}$	1.249	2.0763	1.6871	4.3787
$9 10^{-1}$	2.9574	break	3.177	break

Table 1: *Nonlinear effects on two reference shapes.* D_{Max} *and* σ_{Max} *are respectively the maximum displacement and the maximum stress.*

This suggests to incorporate a constraint on the maximal stress in the fitness function. Of course, some nice structures might be missed, like the perfect ">" shape for $\mathcal{F} = 9 10^{-3}$. But, hopefully, another structure very similar to it can arise, with the same mechanical properties, and with the FEM model giving a reasonable solution among the possible solutions. Following these ideas, the fitness becomes

$$\mathcal{F} = \frac{1}{A_{con} + \varepsilon A_{dis} + \alpha(D_{Max} - D_{Lim})^{+} + \beta(\sigma_{Max} - \sigma_{Lim})^{+}} \tag{3}$$

with the notations of equation 1, σ_{Lim} being the constraint on the stress, σ_{Max} the maximum of the stress on the structure, and β some positive penalty parameter which has to be adjusted. Figure 14 shows the optimal designs obtained with that modified fitness: Reasonable solutions are found, as long as the stress is imposed a strong constraint (computed from a purely linear numerical simulation).

7 Discussion and Perspectives

Breakthrough results have been presented in this paper, on simple problems of topological optimum design. They demonstrate the potentialities of the GA-based topological optimization. However, the computational cost remains the main limitation of the proposed method: using a middle-range HP workstation (HP-755), from 6 hours, for the simplest problem with a 10×20 mesh, up to 30-40 hours, for the three-loading 44×64 bicycle, were necessary for a single successful run. Of course, the parallelization of the fitness computations is straightforward, and would result in a linear speed-up without modification of the underlying algorithm. But it remains that the overall computational requirements are very heavy.

Moreover, the accuracy of all results presented above derives from that of the Finite Element Analyses performed during the computations of the fitnesses. And that accuracy is dictated by the size of the underlying mesh. All the results presented in this paper were obtained on rather

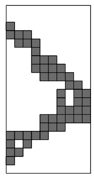
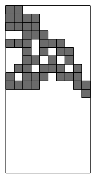
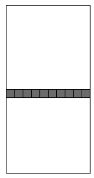
			
$\mathcal{F}$	0.009	0.018	0.09
D_{Lim}	0.22856	0.457	2.2856
σ_{Lim}	0.53	1.0622	5.3
D_{Max}	0.2143	0.4504	1.687
σ_{Max}	0.550	0.9835	4.379
$Area$	0.21	0.47	0.1

Figure 14: Optimal designs for nonlinear elasticity with displacement and stress constraints.

coarse meshes (from 10×20 to 44×64) whereas real-world problems and accurate analyses of the mechanical behavior of the shapes require much finer meshes.

Increasing the size of the mesh would not only increase the cost of the fitness computation, but also the size of the chromosomes. And increasing the size of the individuals would in turn require to increase both the size of the population and the number of generations to reach the same level of convergence (Cerf [12, 13] proved, in the bitstring case, the existence of a critical population size to ensure convergence in finite time; and this critical size increases linearly with the length of the bitstring). Hence, the bit-array representation poorly scales up when refining the mesh — not to speak of handling 3-D shapes.

Therefore other representations have to be used in order to overcome this limitation, and dissociate the complexity of the representation and the accuracy of the evaluation: First results [45, 46] seem to confirm the interest of this direction of research.

8 Conclusion

The feasibility of GA-based optimal design had already been witnessed by [28] and [15]. This paper intends to emphasize on its flexibility, and to demonstrate its potentialities to break some limits of up-to-date deterministic methods.

Multiple quasi-optimal solutions can be found, allowing to take into account some other criteria that could not be incorporated in some objective function (e.g. technological or esthetic criteria). Actually, the results in section 5.2 were obtained by successive runs of the GA starting from different random populations, thanks to the stochastic nature of the algorithm. However, it is possible by using some niching techniques (e.g. the sharing scheme of [22]), to locate in a single run more than one near-optimal solutions.

Results regarding the multi-loading optimization (Section 5.3 were the first of their kind at the time they were obtained. Moreover, they required very few modifications of the method handling a single loading. Since then, the homogenization method has been modified [1] and is now able to optimize structures for multiple loadings. However, deep modification of the numerical procedure was necessary: the modified homogenization algorithm relies on some local optimization performed in every element of the mesh, that was solved analytically in the single loading case, and which is now handled numerically, which dramatically increases its computational cost – though it remains faster than the GA-based algorithm. A fair comparison between stochastic and deterministic approaches should consider the development cost (e.g. the time necessary to work out the adjoint problem in inverse problem solving) together with the computational cost.

Finally, even in the linear framework, stochastic optimization is the only method able to handle topological optimum design with loading on the unknown boundary of the target structure, as demonstrated in section 5.4. Moreover, from the GA point of view, it is a strict application of the original method – the only modification took place during the computation of the fitness, as the external loading had to be computed for each structure.

Last but not least, in the context of large displacements, the experiments presented in section 6 acknowledge for the usual statement "the extension to other mechanical models is straightforward". Note however that the fitness function had to be adjusted in order to take into account the maximal stress of the structure: there is no free lunch for nonlinearity.

However, these good results must not hide the main drawback of the method, namely its computational cost, which makes it highly unlikely to be applicable to real-world optimization problems. Nevertheless, and this is confirmed by preliminary results of on-going work, we are convinced that a change of representation can remedy to that weakness, allowing to treat difficult problems of real size. And the flexibility of the method will then prove essential for some other yet unsolved problems of Topological Optimum Design.

Acknowledgments

The authors are deeply indebted to François Jouve (CMAP) for the invaluable help provided all along this work, from patient theoretical explanations to modifications of his Finite Element software to fit our requirements.

References

[1] G. Allaire and F. Jouve. Personnal communication, 1996.

[2] G. Allaire and R. V. Kohn. Optimal design for minimum weight and compliance in plane stress using extremal microstructures. *European Journal of Mechanics, A/Solids*, 12(6):839–878, 1993.

[3] G. Anagnostou, E. Ronquist, and A. Patera. A computational procedure for part design. *Computer Methods in Applied Mechanics and Engineering*, 97:33–48, 1992.

[4] T. Bäck. *Evolutionary Algorithms in theory and practice.* New-York:Oxford University Press, 1995.

[5] T. Bäck and M. Schütz. Evolution strategies for mixed-integer optimization of optical multilayer systems. In J. R. McDonnell, R. G. Reynolds, and D. B. Fogel, editors, *Proceedings of the 4^{th} Annual Conference on Evolutionary Programming.* MIT Press, March 1995.

[6] T. Bäck, M. Schütz, and S. Khuri. A comparative study of a penalty function, a repair heuristic and stochastic operators with set covering problem. In J.-M. Alliot, E. Lutton, E. Ronald, M. Schoenauer, and D. Snyers, editors, *Artificial Evolution*, number 1063 in LNCS. Springer Verlag, 1995.

[7] T. Bäck and H.-P. Schwefel. An overview of evolutionary algorithms for parameter optimization. *Evolutionary Computation*, 1(1):1–23, 1993.

[8] J. C. Bean and A. B. Hadj-Alouane. A dual genetic algorithm for bounded integer programs. Technical Report TR 92-53, Department of Industrial and Operations Engineering, The University of Michigan, 1992.

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

[10] D. L. Calloway. Using a genetic algorithm to design binary phase-only filters for pattern recognition. In R. K. Belew and L. B. Booker, editors, *Proceedings of the* 4^{th} *International Conference on Genetic Algorithms*. Morgan Kaufmann, 1991.

[11] J. Cea. Problems of shape optimum design. In E. J. Haug and J. Cea, editors, *Optimization of distributed parameter structures - Vol. II*, volume 50, pages 1005–1088. NATO Series, Series E, 1981.

[12] R. Cerf. *Une théorie assymptotique des algorithmes génétiques*. PhD thesis, Université de Montpellier II, March 1994.

[13] R. Cerf. An asymptotic theory of genetic algorithms. In J.-M. Alliot, E. Lutton, E. Ronald, M. Schoenauer, and D. Snyers, editors, *Artificial Evolution*, volume 1063 of *LNCS*. Springer Verlag, 1996.

[14] C. D. Chapman and M. J. Jakiela. Genetic algorithm-based structural topology design with compliance and topology simplification considerations. *Journal of Mechanical Design*, 1995. Submitted.

[15] C. D. Chapman, K. Saitou, and M. J. Jakiela. Genetic algorithms as an approach to configuration and topology design. *Journal of Mechanical Design*, 116:1005–1012, 1994.

[16] P. G. Ciarlet. *Mathematical Elasticity, Vol I : Three-Dimensional Elasticity*. North-Holland, Amsterdam, 1978.

[17] P. G. Ciarlet. *The Finite Element Method for Elliptic Problems*. North-Holland, Amsterdam, 1988.

[18] K. A. DeJong. Are genetic algorithms function optimizers ? In R. Manner and B. Manderick, editors, *Proceedings of the* 2^{nd} *Conference on Parallel Problems Solving from Nature*, pages 3–13. North Holland, 1992.

[19] D. B. Fogel. *Evolutionary Computation. Toward a New Philosophy of Machine Intelligence*. IEEE Press, Piscataway, NJ, 1995.

[20] C. Ghaddar, Y. Maday, and A. T. Patera. Analysis of a part design procedure. *Submitted to Nümerishe Mathematik*, 1995.

[21] D. E. Goldberg. *Genetic algorithms in search, optimization and machine learning*. Addison Wesley, 1989.

[22] D. E. Goldberg and J. Richardson. Genetic algorithms with sharing for multi-modal function optimization. In J. J. Grefenstette, editor, *Proceedings of the* 2^{nd} *International Conference on Genetic Algorithms*, pages 41–49. Lawrence Erlbaum Associates, 1987.

[23] D. E. Goldberg and M. Samtani. Engineering optimization via genetic algorithms. In *Proceedings of the ninth Conference on Electronic Computation*, pages 471–482. American Society of Civil Engineers, University of Alabama at Birmingham, 1986.

[24] D. Grierson and W. Pak. Discrete optimal design using a genetic algorithm. In M. Bendsoe and C. Soares, editors, *Topology Design of Structures*, pages 117–133. NATO Series, 1993.

[25] P. Hajela. *Optimization and Artificial Intelligence in Civil and Structural Engineering*, volume 1, chapter Genetic Algorithms in Automated Structural Synthesis, pages 639–653. Kluwer Academic Publishers, 1992.

[26] P. Hajela, E. Lee, and C. Lin. *Topology Design of Structures*, chapter Genetic Algorithms in Structural Topology Optimization. NATO Series, 1993.

[27] J. Holland. *Adaptation in natural and artificial systems.* University of Michigan Press, Ann Arbor, 1975.

[28] E. Jensen. *Topological Structural Design using Genetic Algorithms.* PhD thesis, Purdue University, November 1992.

[29] J.A. Joines and C.R. Houck. On the use of non-stationary penalty functions to solve nonlinear constrained optimization problems with GAs. In Z. Michalewicz, J. D. Schaffer, H.-P. Schwefel, D. B. Fogel, and H. Kitano, editors, *Proceedings of the First IEEE International Conference on Evolutionary Computation*, pages 579–584. IEEE Press, 1994.

[30] F. Jouve. *Modélisation mathématique de l'œil en élasticité non-linéaire*, volume RMA 26. Masson Paris, 1993.

[31] C. Kane. *Algorithmes génétiques et Optimisation topologique.* PhD thesis, Université de Paris VI, July 1996.

[32] C. Kane, F. Jouve, and M. Schoenauer. Structural topology optimization in linear and nonlinear elasticity using genetic algorithms. In *Proceedings of the ASME 21st Design Automation Conference.* ASME, Boston, Sept. 1995.

[33] C. Kane and M. Schoenauer. Genetic operators for two-dimensional shape optimization. In J.-M. Alliot, E. Lutton, E. Ronald, M. Schoenauer, and D. Snyers, editors, *Artificial Evolution.* Springer Verlag, Septembre 1995. To appear.

[34] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220:671–680, 1983.

[35] R. Leriche and R. T. Haftka. Optimization of laminate stacking sequence for buckling load maximization by genetic algorithms. *AIAA Journal*, 31(5):951–970, May 1993.

[36] C. Lin and P. Hajela. Genetic search strategies in large scale optimization. In *Structures, Structural Dynamics, and Materials Conference*, La Jolla, CA, April 1993. AIAA paper #93-1585.

[37] S. W. Mahfoud. A comparison of parallel and sequential niching techniques. In L. J. Eshelman, editor, *Proceedings of the* 6^{th} *International Conference on Genetic Algorithms*, pages 136–143. Morgan Kaufmann, 1995.

[38] S. Martin, J. Rivory, and M. Schoenauer. Synthesis of optical multi-layer systems using genetic algorithms. *Applied Optics*, 34:2267, 1995.

[39] Z. Michalewicz. *Genetic Algorithms+Data Structures=Evolution Programs.* Springer Verlag, New-York, 1996. 3rd edition.

[40] Z. Michalewicz and C. Z. Janikow. Handling constraints in genetic algorithms. In R. K. Belew and L. B. Booker, editors, *Proceedings of the* 4^{th} *International Conference on Genetic Algorithms*, pages 151–157. Morgan Kaufmann, 1991.

[41] Z. Michalewicz and M. Schoenauer. Evolutionary Algorithms for Constrained Parameter Optimization Problems. *Evolutionary Computation*, 4(1), Spring 1996.

[42] N. J. Radcliffe. Equivalence class analysis of genetic algorithms. *Complex Systems*, 5:183–20, 1991.

[43] N. J. Radcliffe and P. D. Surry. Fitness variance of formae and performance prediction. In D. Whitley and M. Vose, editors, *Foundations of Genetic Algorithms 3*, pages 51–72. Morgan Kaufmann, 1994.

[44] D.J. Schaffer. Multiple objective optimization with vector evaluated genetic algorithms. In J. J. Grefenstette, editor, *Proceedings of the* 1^{st} *International Conference on Genetic Algorithms*. Laurence Erlbaum Associates, 1985.

[45] M. Schoenauer. Representations for evolutionary optimization and identification in structural mechanics. In J. Périaux and G. Winter, editors, *Genetic Algorithms in Engineering and Computer Sciences*, pages 443–464. John Wiley, 1995.

[46] M. Schoenauer. Shape representations and evolution schemes. In L. J. Fogel, P. J. Angeline, and T. Bäck, editors, *Proceedings of the* 5^{th} *Annual Conference on Evolutionary Programming*. MIT Press, 1996.

[47] M. Schoenauer and Z. Michalewicz. Evolutionary computation at the edge of feasibility. In W. Ebeling and H.-M. Voigt, editors, *Proceedings of the* 4^{th} *Conference on Parallel Problems Solving from Nature*, Berlin, 1996. Springer Verlag.

[48] M. Schoenauer and Z. Wu. Discrete optimal design of structures by genetic algorithms. In Bernadou and al., editors, *Conférence Nationale sur le Calcul de Structures*. Hermes, 1993.

[49] H.-P. Schwefel. *Numerical Optimization of Computer Models*. John Wiley & Sons, New-York, 1981. 1995 – 2^{nd} edition.

[50] A. Smith and D. Tate. Genetic optimization using a penalty function. In S. Forrest, editor, *Proceedings of the* 5^{th} *International Conference on Genetic Algorithms*, pages 499–503. Morgan Kaufmann, 1993.

[51] P.D. Surry, N.J. Radcliffe, and I.D. Boyd. A multi-objective approach to constrained optimization of gas supply networks. In T. Fogarty, editor, *Proceedings of the AISB-95 Workshop on Evolutionary Computing*, volume 993, pages 166–180. Springer Verlag, 1995.

[52] G. Syswerda. Uniform crossover in genetic algorithms. In J. D. Schaffer, editor, *Proceedings of the* 3^{rd} *International Conference on Genetic Algorithms*, pages 2–9. Morgan Kaufmann, 1989.

[53] Z. Wu. *Optimisations génétiques en Mécanique des Solides*. PhD thesis, Ecole Polytechnique, April 1996.