Mechanical Inclusions Identification by Evolutionary Computation.

Revue européenne des éléments finis. Vol 5(5-6), 1996. Pages 619–648.

Marc Schoenauer **Leila Kallel** **François Jouve**

CMAP – URA CNRS 756
Ecole Polytechnique
Palaiseau 91128, France

RÉSUMÉ. L'identification d'inclusions est un problème théoriquement mal posé, pour lequel les méthodes numériques déterministes se révèlent instables. Les algorithmes évolutionaires sont des méthodes d'optimisation stochastiques d'ordre 0 fournissant un cadre général à la résolution de problèmes inverses. Cependant, l'application des algorithmes évolutionaires au problème de l'identification d'inclusions doit faire face au problème crucial du choix de la représentation des solutions, c'est-à-dire de l'espace de recherche. Trois représentations sont proposées. Les résultats numériques constituent une avancée importante dans le domaine de l'identification d'inclusions mécaniques, et la robustesse des résultats en présence de bruit est expérimentalement établie. Enfin, les mérites respectifs des différentes représentations sont discutés.

ABSTRACT. The problem of the identification of mechanical inclusion is theoretically ill-posed, and to-date numerical algorithms have demonstrated to be inaccurate and unstable. On the other hand, Evolutionary Algorithms provide a general approach to inverse problem solving. However, great care must be taken during the implementation: The choice of the representation, which determines the search space, is critical. Three representations are presented and discussed. Whereas the straightforward mesh-dependent representation suffers strong limitations, both mesh-independent representation provide outstanding results on simple instances of the identification problem, including experimental robustness in presence of noise.

MOTS-CLÉS : Algorithmes génétiques, optimisation stochastique, problème inverse, inclusions méchaniques, élasticité linéaire.

KEY WORDS : Genetic algorithms, stochastic optimization, inverse problem, mechanical inclusions, linear elasticity.

1. Introduction

In Structural Mechanics, the – non-destructive – identification of inclusions is a difficult problem, resisting to-date numerical methods: in its simplest instance, a structure is known to be made of two different materials of given mechanical characteristics, but their repartition in the structure is unknown. The available data consist of records of the mechanical behavior of the structure under known loadings. The goal is to find the geometrical repartition of both materials from these experimental data. In steel manufacturing plants, for instance, it is of vital importance to check if coal scories are included in steel parts, and if their repartition does not dangerously weaken the whole part. For a given repartition of both materials, the computation of the simulated mechanical behavior of the structure is straightforward, using any Finite Element Analysis (FEM) package. The identification can then be viewed as an inverse problem.

Evolutionary Algorithms (EAs) are stochastic optimization methods that have been demonstrated useful to solve difficult, yet unsolved optimization problems. The most well-known algorithms referring to the Evolutionary Computation paradigm are the Genetic Algorithms (GAs), but other variations on the *"survival of the fittest"* theme also gave birth to powerful optimization algorithms. Requiring no regularity of the objective function (or on the constraints), EAs are able to tackle optimization problems on different kinds of search spaces, such as continuous, discrete or mixed spaces, as well as spaces of graphs or lists. The only prerequisite is the definition of evolution operators such as crossover and mutation, satisfying as much as possible heuristically derived requirements. The two main drawbacks of EAs are first the large number of evaluations of the objective function they usually require before eventually reaching a good, if not optimal, solution; and second, their stochastic aspect, weakening their robustness. Hence, EAs should be used with care, on problems beyond the reach of standard deterministic optimization methods.

This paper addresses the inverse problem of Mechanical Inclusions Identification using EAs. A possible objective function for such inverse problems is the difference between the simulated mechanical behavior of a tentative repartition of both materials, and the actual (experimental) behavior of the real structure. However, the main difficulty is to define the search space in which the EA will search. Considering past works on the Optimum Design problem, (a closely related problem from the Evolutionary Computation point of view, where the goal is to find a partition of a design domain into material and void), a straightforward representation is defined from a fixed mesh of the structure, leading to a fixed-length bitstring representation well-suited to standard Genetic Algorithms. This "bitarray" approach allowed significant breakthroughs in Topological Optimum Design [34, 36]. However, this approach suffers from strong limitations due to the dependency of its complexity on the underlying mesh: the optimization problem rapidly becomes intractable when this mesh is

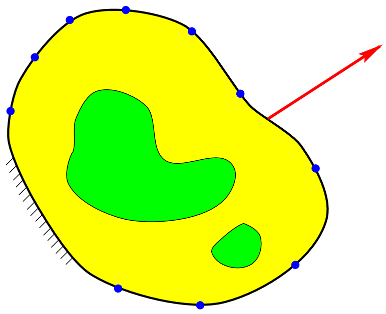


Figure 1: *A sample inclusion problem*

refined. To cope with that difficulty, two non-standard representations (termed the *Voronoï representation* and the *H-representation*) were introduced [48, 49], independent of any *a priori* discretization, but leading to variable-length "individuals" for which specific operators were designed and implemented.

This paper is concerned in applying Evolutionary Algorithms working on these representations to the inclusion identification problem in the context of linear elasticity, and is organized as follows:

The mechanical problem of inclusion identification is presented in details in Section 2. Section 3 gives a brief overview of the state-of-the-art in Evolutionary Algorithms (EAs), introducing the different components of Evolutionary Computation. In Section 4, a general framework to address inverse problems with EAs is introduced: The only prerequisite is a good (i.e. accurate and robust) numerical model of the direct problem, ... and, of course, a good representation of the target search space. Three representations for that problem are presented in Section 5, together with their specific evolution operators (crossover and mutations). The first results, using the Voronoï representation, are presented in Section 6, demonstrating outstanding performance on artificial instances of the inclusion identification problem. Finally, comparative results of all three representations are discussed in Section 6.3, first reinforcing the a priori arguments against the bitstring representation, then giving hints on how to a priori choose between the Voronoï representation and the H-representation on a given instance of a problem. Further directions of research are sketched in the conclusive Section 7.

2. The Mechanical Problem

This section gives a detailed presentation of the mechanical problem of inclusion identification, and states the simplification hypotheses made throughout this paper.

2.1. Background

Consider an open bounded domain $\Omega \subset \mathbb{R}^N$ ($N = 2, 3$), with a sufficiently smooth boundary $\partial\Omega$, filled with a linear elastic material (as in Figure 1). Under the hypothesis of small deformations (linear elasticity) the following equations hold:

$$\varepsilon(x) := \frac{1}{2}(\nabla u(x) \nabla u^T(x)) \qquad \text{the strain tensor,} \tag{1}$$

and

$$\sigma(x) := A(x)\varepsilon(x) \qquad \text{the stress tensor,} \tag{2}$$

where $u(x)$ is the displacement field at point x and $A(x)$ is the elasticity tensor (a fourth order tensor) involved in the *Hooke's law* (2): A is supposed to be *inhomogeneous*, which means that its value depends on the point x, and $A(x)$ is a positive definite tensor which satisfies some symmetry conditions.

When A is given, one can state two kinds of boundary value problems, respectively of *Dirichlet* and *Neumann* type:

$$\left\{ \begin{array}{rcll} \mathrm{div}\sigma & = & 0 & \text{in } \Omega, \\ u & = & u_0 & \text{on } \partial\Omega, \end{array} \right. \tag{3}$$

and

$$\left\{ \begin{array}{rcll} \mathrm{div}\ \sigma & = & 0 & \text{in } \Omega, \\ \sigma.n & = & g_0 & \text{on } \partial\Omega, \end{array} \right. \tag{4}$$

where u_0 and g_0 are respectively a given displacement field and a given external force field on the boundary $\partial\Omega$.

It is well known (see *e.g.* [13] that each of these problem has a unique solution (for the Neumann's problem, one has to impose an integral condition on g_0 to ensure existence, and an integral condition on u to eliminate rigid displacements).

In the following, the *inverse problem* will be considered:

$$\text{find } A \text{ such that } , \forall i \in \{1, ..., n\}, \exists u, \left\{ \begin{array}{rcll} \mathrm{div}\ \sigma & = & 0 & \text{in } \Omega, \\ u & = & u_i & \text{on } \partial\Omega, \\ \sigma.n & = & g_i & \text{on } \partial\Omega, \\ \sigma & = & A\varepsilon, & \end{array} \right. \tag{5}$$

where $(g_i)_{i=1,..,n}$ and $(u_i)_{i=1,..,n}$ are given.

Problem (5) is a discrete version of the "ideal" inverse problem:

$$\text{find } A, \text{ given the application: } \Lambda_A : u_{|\partial\Omega} \longrightarrow \sigma_{|\partial\Omega}. \tag{6}$$

The underlying physical problem is still lacking much more of known data than problem (5) since Λ_A is only known through a finite number of *experimental measurements* performed at a finite number of points. Hence, the "real" identification problem treated by mechanical engineers can be stated as:

$$\text{find } A \text{ such that } , \forall i \in \{1,...,n\}, \exists u, \quad \left\{ \begin{array}{rcll} \text{div } \sigma & = & 0 & \text{in } \Omega, \\ u(x^j) & = & u_i^j & \forall j \in \{1,...,p\}, \\ \sigma(x^j).n & = & g_i^j & \forall j \in \{1,...,p\}, \\ \sigma & = & A\varepsilon, & \end{array} \right. \tag{7}$$

where information on the boundary is only known at a finite number of experimental points $(x^j)_{j=1,...,p}$ for a finite number (n) of experiments. In addition, these data may be known with a certain amount of experimental error or noise.

2.2. State of the art

If the aim is the numerical treatment of problem (6) (and *a fortiori* problem (5) or (7)) by "classical" (*i.e.* non-stochastic) methods, two theoretical points are crucial:

- existence and uniqueness of A as a function of Λ_A,
- continuity of the dependency of A with respect to Λ_A.

Existence is of course essential to the pertinence of the identification problem, but uniqueness and continuity are only needed to ensure the reliability and the stability of deterministic numerical algorithms. On the other hand, EAs can deal with non-continuous functionals and non-unique solutions.

Problem (6) is the elastic equivalent of the so-called *tomography* problem where the elliptic operator is the conductivity operator ($\text{div}(A\nabla u)$, u scalar field) instead of the elasticity one ($\text{div}(A\varepsilon(u))$, u vector field).

The tomography problem has been widely studied. Under some hypothesis, existence and uniqueness have been proved. However, the continuity of the functional is only known in a weak sense, that cannot help numerical simulations.

The elasticity problem (6) is more difficult. Existence and uniqueness have been proved for isotropic Hooke's laws, but there is no continuity result (see [8, 9], or [15] for a comprehensive bibliographical discussion on this subject).

Numerical simulations by classical methods have shown that both tomography [37] and elastic identification problems [15] are ill-posed, and thus EAs are good tentative choice for a robust numerical method.

2.3. The simplified problem

In this preliminary paper, attention has been focused on representation and on specific operators for EAs. To highlight specific problems involved in these algorithms, the mechanical problem (7) was restricted to a two-dimensional simpler class of problems:

Let A_1 and A_2 be two isotropic elasticity tensors, fully defined by Young's moduli E_1 and E_2 and Poisson ratios ν_1 and ν_2. The aim is to solve Problems (5) and (7), restricting allowable values of $A(x)$ to

$$A(x) = \begin{cases} A_1 & \text{if } \chi(x) = 0 \\ A_2 & \text{if } \chi(x) = 1 \end{cases} \tag{8}$$

where χ is a characteristic function defined on Ω.

These problems, although less general than (5) and (7) are still beyond the capabilities of deterministic algorithms (see [15]).

However, Problem (7) can be treated in the same way, as well as identification in non-linear elasticity (this is part of on-going work, see Section 7).

The direct elasticity problem is solved by a classical finite element method. All simulations (see Section 6) have been performed on a square domain, held fixed on its left side, discretized by a regular mesh of size 24×24 elements. All details and mechanical constants are specified in Section 6.

3. Evolutionary Algorithms

This section presents a brief overview of Evolutionary Algorithms, both giving a modern generic and pragmatic point of view, and presenting a historical perspective on Evolutionary Computation.

3.1. The Generic Evolutionary Algorithm

Evolutionary Algorithms are stochastic optimization algorithms based on a crude mimic of the Darwinian principle of the *Survival of the fittest*.

Let the search space be a metric space E, and let F be a function from E on $\mathbb{R}$ called the *fitness* function. The problem is to find the optimum of F on E.

A *population* of size $P \in \mathbb{N}$ is a set of P *individuals* (points of E) not necessarily distinct. This population is generally initialized randomly, uniformly on E. The fitnesses of all individuals are computed. The population then undergoes a succession of *generations*, that can be described the following way:

- **Selection**: Some P individuals are selected to give birth to offspring. Numerous selection processes can be used, either deterministic or stochastic. However, all implement in one way or another the Darwinian principle of the *Survival of the fittest*: the selection step is based on fitness comparisons among individuals. Depending on the selection scheme used, some individuals can be selected more than once. At that point, the selected individuals give birth to copies of themselves (clones).
- **Application of evolution operators**: To each one of these copies is applied some evolution operator, giving birth to one or more offspring. The

choice among possible operators is stochastic, according to user-supplied probabilities. These operators are always stochastic operators, and one usually distinguish between *crossover* (or *recombination*) operators, when two (or more) parents exchange part of their genetic material to build up one offspring, and *mutation* operators, in which one parent is randomly modified.

- **Evaluation**: Computation of the fitnesses of all newborn offspring.
- **Replacement**: Choice of which individuals will be part of next generation. The choice can be made upon offspring only, or among offspring and parents. It is also based on fitness values.
- **Stopping criterion**: The process stops when: A target value for the best fitness – when it is known – is reached, or some a priori maximal number of generations is run, or the best fitness value in the population does not improve during a user-defined number of generations, which ever comes first. If the stopping criterion is not fulfilled, another generation starts.

Ideally, the evolution operators should be defined on the same space than the fitness function, called *phenotype space*, or behavioral space. Unfortunately, in most cases, one must introduce an intermediate space, called *genotype space*, or representation space. The mappings from the phenotype space in the genotype space is termed *coding*. The inverse mapping from the genotype space in the phenotype space is termed *decoding*. Genotypes undergo evolution operators, and their fitness is evaluated on the corresponding phenotype. The properties of the coding mappings can greatly modify the dynamical behavior of the evolutionary algorithm.

3.2. Historical EAs

The most widely known Evolutionary Algorithms probably are Genetic Algorithms (GAs). First introduced by Holland [27], then popularized by Goldberg [24], historical GAs are based on a *binary representation* of the individuals (then often termed *chromosomes*) and heavily rely on the cross-over operator (i.e. stochastic operators generating one offspring from two Darwinian-selected parents). Parents are selected based on their fitness, the operator considered crucial is the crossover operator, and the replacement step simply replaces all parents by all offspring.

On the other hand, historical Evolution Strategies [50, 44] and Evolutionary Programming [20] try to use the "natural" search space for the problem at hand (e.g. $\mathbb{R}^n$ when it comes to real parameter optimization). All parents are selected randomly, the main operator is considered to be the mutation operator, and the replacement step is a deterministic choice (based on fitness values) among offspring and possibly parents in Evolution Strategies, and a stochastic tournament between parents and offspring in Evolutionary Programming. Moreover, for real-valued parameter optimization, all these algorithms use what is called *Gaussian mutation*:

$$x \stackrel{mutation}{\longrightarrow} x + N(0, \sigma), \tag{9}$$

where $N(0, \sigma)$ is a normal random variable with user-defined deviation $\sigma > 0$.

Nevertheless, modern Evolutionary Algorithms are defined on a pragmatic basis: be they real-encoded GAs [40] or generalized Evolution Strategies using recombination [50]. Moreover, any possible evolution scheme (i.e. choice of selection-replacement methods) can be used, "as long as it works". More details about the recent advances in Evolutionary Computation can be found in [1, 18]. These references also describe examples of successes of Evolution Algorithms on difficult global optimization problems, on which standard classical methods (e.g. gradient-based methods for numerical optimization) fail.

In any case, the price to pay for the wide application area is a fairly high computational cost. Moreover, many different parameters are involved (choice of the evolution scheme, population size, probability of application of different operators, stopping criterion). These parameters can have dreadful consequences on the overall convergence of the algorithm, they highly depend on the problem at hand, and no attempt to automatically adjust them has proved to be general enough [25, 26] but the usual trial-and-error systematic testing [47].

Theoretical results have been derived regarding the convergence of some specific instances of EAs. The strongest of such results deal with the convergence of ESs on convex functions. Not only do the algorithms converge to the global optimum with probability 1 [50], but also Beyer obtained very precise quantitative convergence rates for various schemes in the ES framework [2, 3, 4, 5, 6]. On the GA side, a nice global convergence result in probability has been proved by Cerf [10, 11] on a modified bitstring GA, while different partial results – including a non-convergence result on the non-elitist GA – have been shown by Rudolph [46, 45]. However, these theoretical results are still of poor use when it comes to practical implementation (e.g. no convex function needs EA to be optimized, and the minimum population sizes obtained by Cerf are too large to be actually used). Hence, the user can only refer to past experimentations described in the literature.

3.3. Implementation hints

As said above, one strength of EAs is their ability to work on weird search spaces (i.e. with non-standard representations), as long as evolution operators (crossover and mutation) are provided. Moreover, though no theoretical results exist to show the way to design a successful evolutionary algorithm, some heuristically derived basic principles have been stated:

- the variance of the fitness values of all individuals sharing some genetic materials (the *schemata* in GA terminology) should decrease when the amount of common material increases [43];
- crossing-over two individual should respect their common genetic material [42];
- mutation should be *ergodic* [42], i.e. a finite number of mutations should be able to join any two points of the search space; this point is also crucial for all theoretical convergence results based on Markov chain analysis [11];
- mutation should respect the principle of *strong causality* [44], i.e. small changes of the genotype should provide small variations of the fitness[1].

Other important features to be considered are the the degeneracy of the representation (how many different genotypes map onto the same phenotype) and its redundancy (genetic material that could be omitted without modifying the phenotype). Whereas redundancy can be beneficial (see for instance the metaphor of natural introns [38]), it implies some degeneracy, which is usually considered harmful to evolutionary optimization. Finally, the degree of *epistasis* of the representation (the interaction between different part of the genotype to build the phenotype) must also be taken into account. Section 7 will consider these questions for the specific representations used for the inclusion identification problem, in the light of the experimental results of the evolutionary optimization method described in the rest of the paper.

4. Evolutionary Inclusion Identification

This section presents the general numerical framework that will be used to solve the mechanical problem introduced in Section 2, using Evolutionary Algorithms.

4.1. An inverse problem

The problem to solve is the inverse problem corresponding to the direct problem described by Equation 8 of Section 2. Having a good simulation of a direct problem through Finite Elements Analysis, Evolutionary Computation can be used to address the inverse problem in a straightforward manner, as described schematically in Figure 2: the results of the FE Analyses performed

[1] another approach to the relation between operators and fitness can be found in [19].

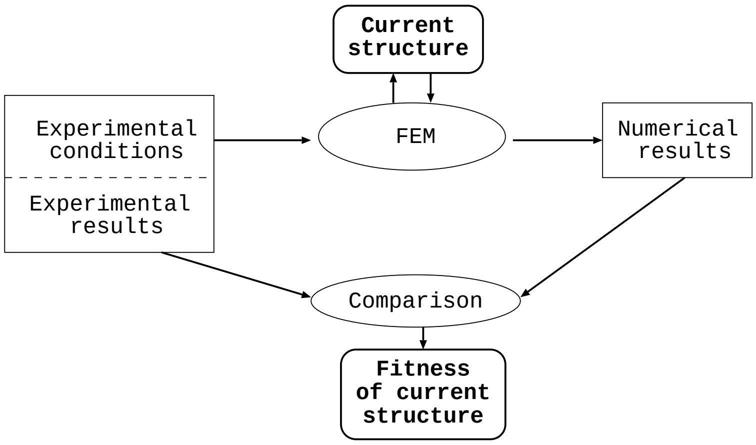


Figure 2: *Evolutionary approach for the inverse problem of mechanical inclusion identification: The fitness of the individual at hand is achieved by comparing the actual experimental results with the numerical results obtained by Finite Element Analysis on the structure that the individual represents.*

on the structure at hand are compared to original experimental results, and the goal is to reach the smallest possible error.

The experimental data will consist in given displacements of the structure under given loading cases.

4.2. Experimental settings

All numerical results for the problem at hand (Section 2) will be obtained on the two-dimensional square structure presented in Figure 3: the structure is held fixed on its left-side, and normal point-wise forces are applied at given points of the three other boundaries.

The aim is to identify the repartition of two materials into the square: a hard material ($E = 1.5$ and $\nu = 0.3$) and a soft material ($E = 1$ and $\nu = 0.3$). A fixed mesh of size 24×24 will be used used throughout the experiments, unless otherwise mentioned.

The reference experimental loading cases used to compute fitness values were actually computed from a known configuration of materials inside the structure. The optimal solution is thus known, which allows a better insight and understanding during the evolution of the population. Moreover, much flexibility was required during the tuning of the overall process, that actual experimental results could not have bought. Finally, considerations about the noise in the experimental reference recording also favor simulated results: there still is a bias (due to numerical error in the Finite Element Analysis), but, this

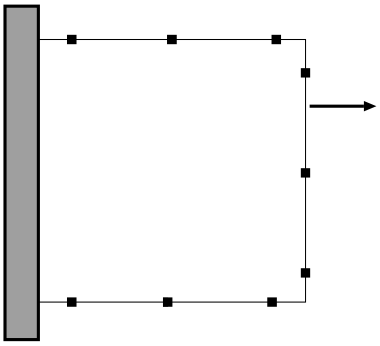


Figure 3: *The experimental structure is fixed on its left boundary, and normal forces are applied on different points of the free boundary. The large dots are the 9 real-world-like measure points.*

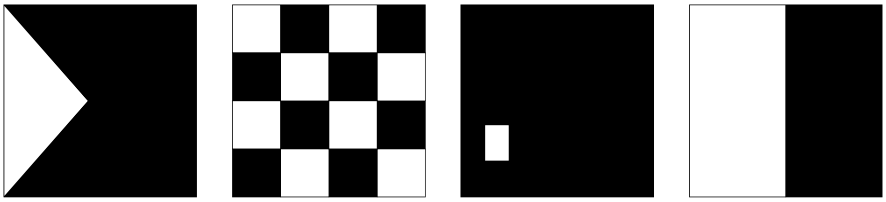


Figure 4: *The experimental structure, and the four test cases, respectively termed corner, checkerboard, asymmetric and symmetric.*

bias is hopefully the same for the experimental results and during the fitness computation. Hence, it should not weaken the results of the evolutionary algorithm, as could unpredictable noise in actual measures. Of course, further work will have to consider results of actual experiments to fully validate the approach.

Figure 4 shows the four reference structures that will be used throughout this paper: the fairly easy *corner* problem and the *checkerboard* problem, difficult from the mechanical engineers point of view, are taken from [15], while the *asymmetric* and *symmetric* problems are purposely designed to test different behaviors of the representations introduced further in this section.

4.3. The fitness functions

In real world situations, the design of the fitness function should take into account all available loading cases experimented on the real structure, using as much information as possible about the behavior of the structure to be identified. However, the choice made in these preliminary experiments of using "simulated experiments" makes it possible to use as many loading cases as needed. In an attempt to have a sampling of the mechanical behavior of the structure as uniform as possible over the domain, 37 different loading cases were used, each loading case consisting of applying a given normal force at one point of the boundary of the structure (as shown on Figure 3).

A very important issue in inverse problem solving is the *generalization capability* of the solution: How good is the result of the optimization process when used under experimental conditions that are different from the ones used during the optimization itself? The usual answer is to use, during the identification, many different experimental conditions, also termed *fitness cases*. The fitness is then the average of the error over all fitness cases. Needless to say, the total computational cost increases with the number of fitness cases.

However, the use of simulated "experimental" values allows to test the overall algorithm with different points of view: In real world situations, some gauges are placed to the boundary of the structure, and only the displacements at those points are available. Figure 3 is an example of such a situation, where 9 gauges are represented. However, it seems clear that the more measure points, the easier is the identification task. Hence, three different fitness functions have been used throughout the numerical experiments presented in Section 6, using the 37 loading cases described above, with different numbers of measure points:

- The most informative fitness function, referred hereafter to as the *total fitness*, takes into account the displacements at *all nodes* of the mesh.
- An intermediate fitness function uses only the displacements of all nodes lying at the boundary fo the structure, and is termed the *boundary fitness*.
- The *real-world fitness* uses only the 9 measure points represented on Figure 3, incorporating much less information to the fitness function.

4.4. The Evolutionary Algorithm

The Evolutionary Algorithm used for all experiments presented in this paper is a standard GA scheme: rank-proportional selection (only the rank in the population of all individuals is taken into account in the selection process, regardless of the actual fitness values), crossover rate of 0.6, mutation rate per individual of 0.2, all offspring replace all parents. The population size is set to 100, and at most 300 generations of the algorithms are allowed – but it stops whenever 50 generations are run without any improvement of the overall best fitness. Hence, between 10000 and 25000 Finite Element Analyses (each involving 37 loading cases, but with the same global matrix) were performed for each run, requiring around 2 hours of CPU time on a high range HP workstation

(HP-PA 8000) for the 24×24 mesh used in most cases.

However, the stochastic nature of the algorithm imposes that all experiments are run many times with different independent initial populations: as clearly stated in [30], "You should never draw any conclusion of a single run of any Evolutionary Algorithm".

Hence, for each setting of experimental conditions (problem and representation), 21 independent runs have been performed, and the forthcoming results of Section 6.3 presenting the online performances (fitness along generations) or the off-line performances (fitness reached in the end of evolution) of these runs are *median* results: At each generation, the best individuals of all runs are sorted according to fitness, and the median value (here, the 11^{th} out of 21) is shown. Note that EA results are generally presented through their averages. But, as discussed in [21], whereas using the average values (together with standard deviation and/or T-test results) make the implicit assumption of normal repartitions for the sampled values, choosing the median values seems more adapted to non-parametric statistics. Moreover, when dealing with fitnesses that can take very small values (as the mean square errors of the displacements), averages hardy reflect actual behavior of the algorithm: Error values for different runs can be of fairly different orders of magnitude. Hence, a few large values might hide a majority of very small errors.

4.5. The genotype space

The most critical step in evolutionary algorithms is the choice of the representation, which defines the search space. In any case, the algorithm will find a good – if not the best – solution in the search space. Hence it seems that larger search spaces allow better solutions to be found. However, the larger is the search space, the more difficult is the optimization task, and a trade-off has to be found. Next section will present different tentatives to address this issue.

5. Representations for mechanical inclusions

This section introduces three representations for the problem described in Section 4.2. The first one is the obvious mesh-based bitstring representation. The next two ones are variable-length representations, and use real numbers as main components. Hence, specific operators (e.g. crossover and mutation) have to be designed for each of these representation.

5.1. Prerequisites

A solution to the inclusion identification problem is a partition of the domain of the structure into two subsets, each subset representing one of the materials involved. Moreover, all connected components of any subset should have a non-void interior and a regular boundary.

A theoretical framework has been developed by Ghaddar & al. [23] in the context of Structural Optimum Design: The search space is restricted to partitions with polygonal boundaries. Theoretical results 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 [23], the same search space will be used here.

However, a significant difference between the objective functions in [23] and the one to be used here is that the inclusion identification problem requires a Finite Element Analysis on the direct problem to compute the fitness of a point of the search space (i.e. a given repartition of both materials), as detailed in Section 4.3. It is well-known that meshing is a source of numerical errors [14]. Hence, for any Evolutionary Algorithm, using a fitness function based on the outputs of two Finite Element Analyses performed on different meshes is bound to failure, at least when the actual differences of behavior will become smaller than the unavoidable numerical noise due to remeshing. The use of the same mesh for all Finite Element Analyses is thus mandatory in order to obtain significant results, at least for structures whose mechanical behaviors are to be compared, i.e. inside the same generation.

5.2. The bitarray representation

Once the decision to use a fixed mesh has been taken, and with even very little knowledge of EAs, a straightforward representation for a partition of the given domain uses 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. The resulting representation can be viewed as a bitstring or, more precisely, as a bitarray. Indeed, considering such array of bits as a one-dimensional bitstring is bound to failure, and specific two-dimensional crossover operators have to be designed (see [35]). Nevertheless, almost all previous works using Genetic Algorithms on the related Optimum Design problem did use that representation [28, 12, 33, 36].

However, the limits of this bitstring representation clearly appear when it comes to refine the mesh (in order to reach better precision of the resulting shape) or if one wants to apply evolutionary techniques to solve 3-dimensional problems: this would imply a huge bitstring, as the size of the bitstring is that of the underlying mesh. In order to reach convergence with large bitstrings, larger populations are required, increasing linearly with the size of the bitstring, according to both theoretical results of Cerf [11] and empirical studies [51].

Moreover, more generations are also needed to reach convergence, and the resulting algorithm rapidly becomes intractable. Some numerical experiments of section 6 confirm this phenomenon.

These considerations show the need for other representations, not relying on a given mesh – even if a fixed mesh is used thereafter to compute the fitness function. Two of such representations have been designed, and successfully used on the Optimum Design problem [48, 49].

5.3. The Voronoï representation

A possible way of representing partitions of a given domain comes from computational geometry, more precisely from the Voronoï diagram theory. The ideas of Voronoï diagrams are already well-known in the FEM community, as a powerful tool to generate good meshes [22]. However, the representation of partitions by Voronoï diagrams to describe their evolutionary optimization seems to be original.

Voronoï diagrams: Consider a finite number of points $V_0, \ldots, V_N$ (the *Voronoï sites*) of a given subset of $\mathbb{R}^n$ (the design domain). To each site V_i is associated the set of all points of the design domain for which the closest Voronoï site is V_i, termed *Voronoï cell*. The *Voronoï diagram* is the partition of the design domain defined by the Voronoï cells. Each cell is a polyhedral subset of the design domain, and any partition of a domain of $\mathbb{R}^n$ into polyhedral subsets is the Voronoï diagram of at least one set of Voronoï sites (see [41, 7] for a detailed introduction to Voronoï diagrams, and a general presentation of algorithmic geometry).

The genotype: Consider now a (variable length) list of Voronoï sites, each site being labeled 0 or 1. The corresponding Voronoï diagram represents a partition of the design domain into two subsets, if each Voronoï cell is labeled as the associated site (here the Voronoï diagram is supposed regular, i.e. to each cell corresponds exactly one site). Example of Voronoï representations can be seen in Figure 5. The Voronoï sites are the dots in the center of the cells. Note that this representation does not depend in any way on the mesh that will be used to compute the mechanical behavior of the structure. Furthermore, Voronoï diagrams being defined in any dimension, the extension of this representation to $\mathbb{R}^3$ and $\mathbb{R}^n$ is straightforward.

An important remark is that this representation presents a high degree of *epistasis* (the influence of one site on the physical shape is modulated by all neighbor sites). This will be discussed in more details in Section 7.

Decoding: Practically, and for the reasons stated in Section 5.1 the fitness of all structures will be evaluated using the same fixed mesh. A partition described by Voronoï sites is thus mapped on this fixed mesh: the subset an

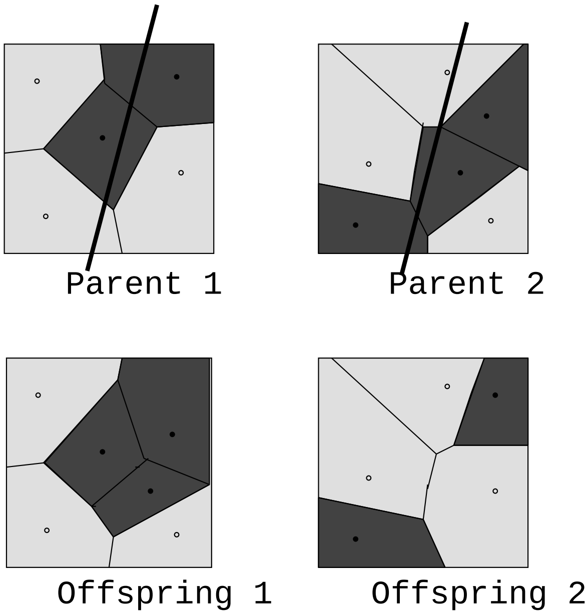


Figure 5: *The Voronoï representation crossover operator. A random line is drawn across both diagrams, and the sites on one side are exchanged*

element belongs to is determined from the label of the Voronoï cell in which the center of gravity of that element lies.

Evolution operators: The evolution operators on the Voronoï representation are inspired by the two-dimensional crossover operators designed for the bit-array representation [35, 33] and by the standard Gaussian mutation operators for real-valued evolutionary algorithms coming from Evolution Strategies [50].

- The crossover operators exchange Voronoï sites on the basis of geometrically-based choice. In this respect it is similar to the specific bitarray crossover described in [35]; moreover, this mechanism easily extends to any dimension [31]. Figure 5 demonstrates an application of this crossover operator.
- a first mutation operator performs a Gaussian mutation on the coordinates of the sites, as in Evolution Strategies (see Equation 9) or randomly flips the boolean attribute of some sites. In all cases, the strength of the mutation (the standard deviation of the Gaussian mutation, or the number of sites whose labels are flipped) is adjusted proportionally to the relative fitness of the individual at hand, as in Evolutionary Programming [17]: fit individuals undergo weak mutations while unfit individuals are more deeply modified.

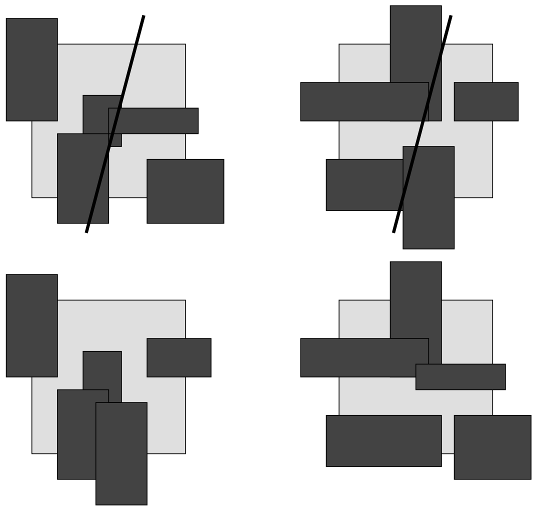


Figure 6: *The H-representation crossover operator. A random line is drawn across both structures, and the holes on one side are exchanged.*

- "standard" mutation operators for variable-length representations must include addition and destruction of some sites on the list.

5.4. H-representation

Another representation for partitions is based on an old-time heuristic method in Topological Optimum Design (TOD): from the initial design domain, considered as plain material, one removes material at locations where the mechanical stress is minimal, until the constraints are violated. However, the lack of backtracking makes this method useless in most TOD problems. Nevertheless, this idea gave birth to the "holes" representation [16], later termed H-representation.

The representation: The design domain is by default made of one material, and a (variable length) list of "holes" describes the repartition of the other material. These holes are elementary shapes taken from a library of possible simple shapes. Only homogeneous representations involving either rectangular or triangular holes are considered at the moment. Moreover, two different representations can be designed from each elementary form, depending on which material the "holes" actually stand for. Results of Section 6.3 will show that this choice does influence the results. Three representations will hence be considered in the rest of the paper: the 1-rectangle representation, in which holes are rectangles representing material 1, the 0-rectangle representation where the rectangular holes stand for material 0, and the 1-triangle representation (with self-explained meaning).

Example of structures described in the 1-rectangle representation are presented in Figure 6. The rectangles are taken in a domain larger than the design domain, in order not to bias the boundary parts of the design domain toward the default value.

All holes representations, as the Voronoï representation, is independent from any mesh, and hence its complexity does not depend on any required accuracy for the simulation of the mechanical behavior of the structure. Its merits and limitations will be discussed in the light of the experimental results presented in section 6.

Decoding: As for the Voronoï representation, the simulated behavior of the shapes is computed on a given fixed mesh, to limit the numerical noise due to re-meshing. The criterion to decide which subset an element does belong to, is based on whether its center of gravity belongs to a hole (in which case the whole element is considered made of the hole-material) or not.

Evolution operators: The evolution operators are quite similar to those of the Voronoï representation:

- crossover by geometrical (2D or 3D) exchange of holes (see Figure 6 for an example);
- mutation by Gaussian modification of the characteristics (coordinates of the center, width and length) of some holes, as defined by Equation 9;
- mutation by addition or destruction of some holes;

Initialization procedure: whereas the initialization procedure for the Voronoï representation was straightforward (uniform choice of the number of Voronoï sites between 1 and a user-supplied maximum number, and uniform choice of the sites in the structure, and of their label), that of the holes representation raises some interesting issues (discussed in details in [32]), as no obvious procedure exists that would be "uniform" over the space of holes. Based on results involving Fitness Distance Correlation analyses [29, 32] that are out of the scope of this paper, the following were used, assuming a rectangular structure $\Omega = [0, L_x] \times [0, L_y]$:

- In all cases, the number of holes is uniformly chosen in $[1, N_{Max}]$ for some user-supplied parameter N_{Max} (50 in all experiments presented here).
- For the rectangle representations, uniform random choice of the center of the rectangle in Ω, and uniform random choice of the width and height in $[0, \frac{L_x}{4}]$ and $[0, \frac{L_y}{4}]$ respectively.
- For the triangle representations, uniform random choice of a first point in Ω, then determination of the two other points by gaussian perturbations of mean deviation $x_{Max}/2$ of the coordinates of the first point.

6. Numerical Results

This section presents some numerical experiments on the inclusion identification problem using Evolutionary Algorithms. The first results address the feasibility issue. In subsection 6.2, EAs demonstrate their robustness in the presence of noise in the experimental data. Finally, comparative experiments of subsection 6.3 bring a partial answer to the question of the a priori choice of a representation for a given instance of problem. In particular, the bitarray representation definitely demonstrates its poor performances when compared to the mesh-independent representations (Voronoï , 1- and 0-rectangles).

6.1. Feasibility Results

The very first experiments were performed using the Voronoï representation, described in Section 5.3. They were obtained on the reference partitions represented in Figure 4 of section 2.

The first results on the corner problem were astonishingly good: the exact solution was found 3 times out of 10 independent runs, in less than 100 generations when using the "total" fitness, and even twice (in about 250 generations) using the boundary fitness. Figure 7 shows an example of a successful run, where the best individual at different stages of evolution is plot together with the error. The Voronoï sites are represented on the figure by grey little dots.

On the symmetrical and asymmetric problems of Figure 4, some difference begin to appear between the different fitnesses, but these examples are still easy problems. And when it came to the checker problem, the total fitness gave much better results than the real-world fitness, as can be seen in Figure 8. However, the real-world fitness gave some interesting results, as can be seen on Figure 8-b: the actual values are clearly identified along the free boundary, and badly along the fixed part of the boundary, where very little information is available.

6.2. Results on noisy data

After these first satisfactory results on exact data, further validation of the proposed approach had to take into account possible errors and noise in the data, before using actual – noisy – experimental data. In order to test the sensitivity of the algorithm to noise, artificial noise was purposely introduced in the reference "experimental" displacements.

Figure 9 shows the results obtained on the (easy) "corner" example, when the relative amount of noise is set to 2% and 5%: all reference displacements were multiplied by a term $(1+\varepsilon)$, where ε was a random variable uniformly distributed in $[-0.02, 0.02]$ and $[-0.05, 0.05]$ respectively. The results are (of course !) degraded, but they demonstrate a fairly good robustness, at least on

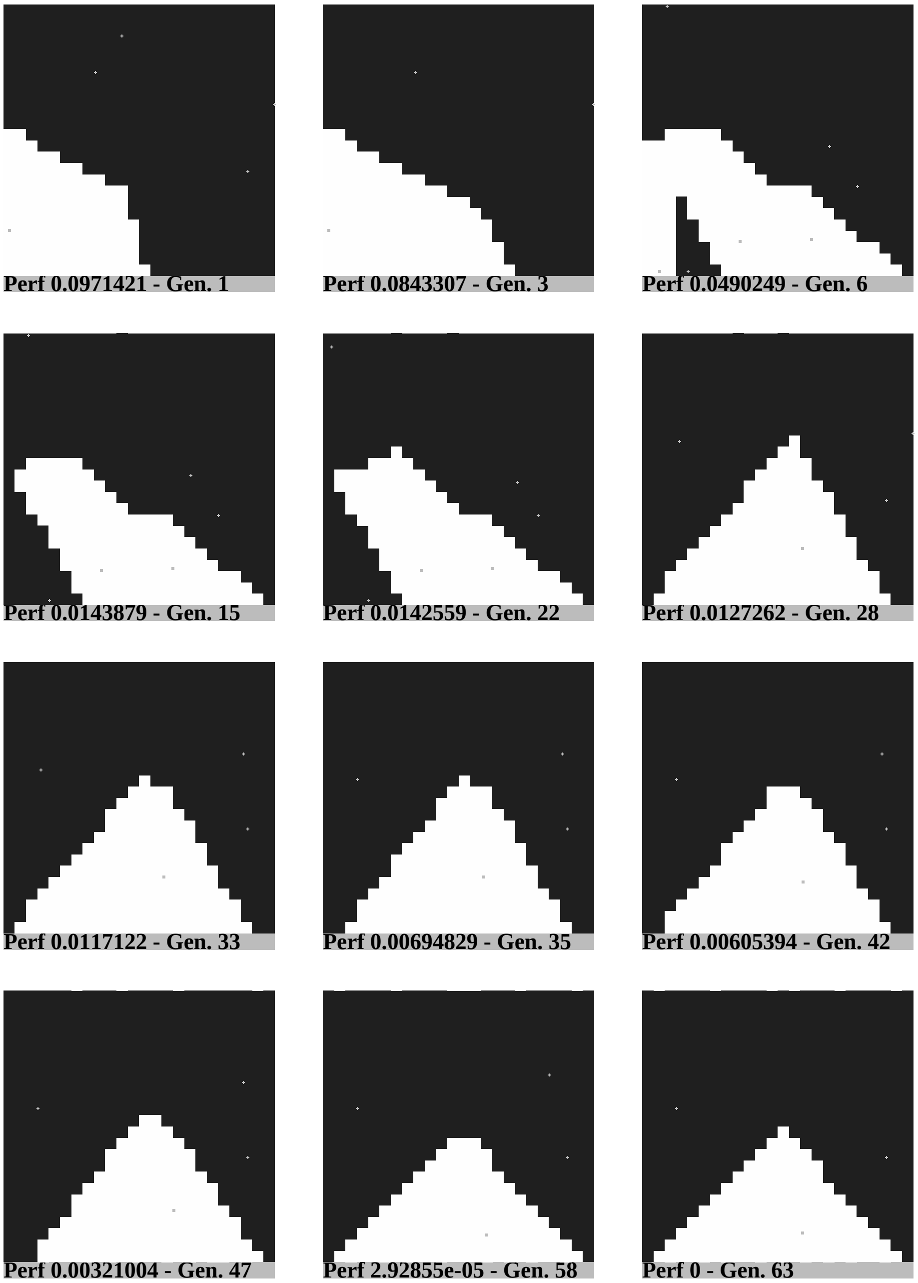
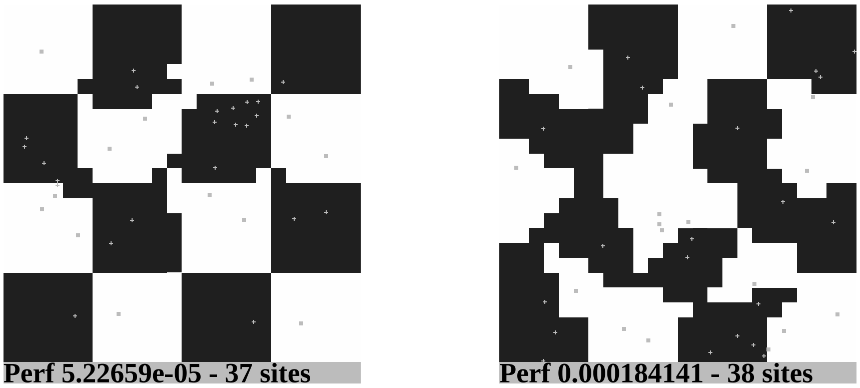


Figure 7: *A successful run of the Voronoï representation on the corner problem using boundary fitness. Plots of the best individual at different generations of the evolution*



(a) (b)

Figure 8: *The – difficult – checker problem: (a): with total fitness. (b): with real fitness.*

this example.

6.3. Comparative results

This section compares the four representations defined in section 5, namely the bitarray, the Voronoï , the 1- and 0-rectangles and the triangle representations. As discussed in Section 4.4, all results are median over 21 independent runs.

Figure 10 presents the on-line results, i.e. the plots of error on the displacements along generations, while Table 1 presents the off-line results after 300 generations. Note that the 0-rectangle was not tested on the two symmetrical problems (the symmetric and checker-board problems): though the problems are not mechanically symmetric (the hard and soft materials are exchanged), the asymmetry is hardly distinguishable from the EA point of view, and the results of the 0-rectangle representation are broadly of the same nature than those of the 1-rectangle on these problems.

Some clear tendencies emerge from these results:

- The bitarray representation is clearly outperformed by all other representations, except on the checker-board problem, where the difference is not significant. Moreover, the best run of the bitarray trials was evolved up to 1000 generations, with resulting errors of respectively 0.014, 0.0004, 0.0003 and 0.0005 for the 4 problems, and were worse than the results of most other representations in 300 generations. In particular they were never better than the Voronoï representation.

- The Voronoï representation performs globally better than all Holes rep-

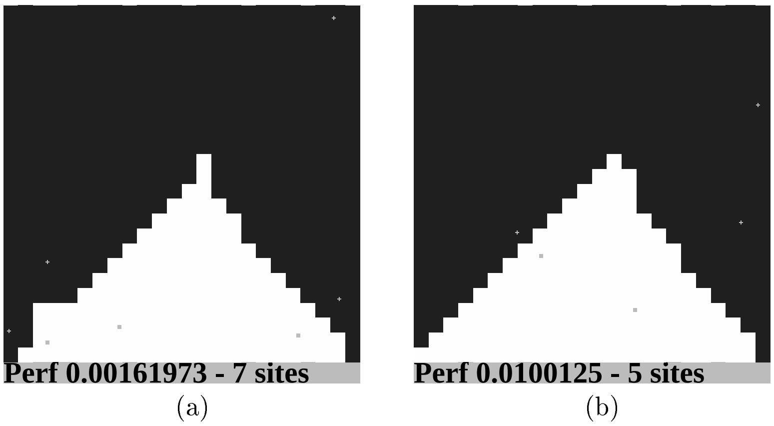


Figure 9: *Robustness to noise. (a): with 2% noise. (b): with 5% noise.*

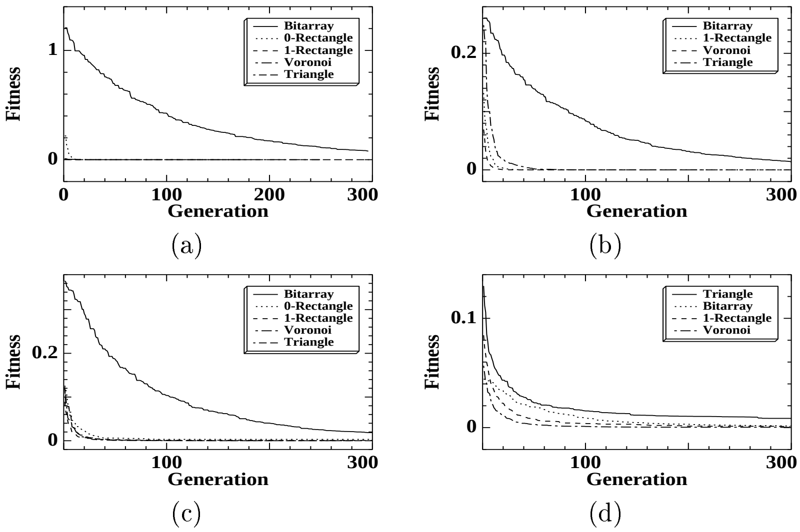


Figure 10: *Comparative runs on 300 generations involving all representations on the four test problems: bitarray performances are lousy, except for the checker-board problem.*

	Asymmetric		Symmetric		Corner		Checker	
	best	median	best	median	best	median	best	median
bitarray	42	75	14*	14*	42	75	1.5*	1.5*
Voronoï	0	0.232	0	0.003	0.002	0.036	0.071	0.282
1-Rect.	0.004	0.048	0	0.028	0.24	0.81	0.355	1.07
0-Rect.	0.085	0.27			0.39	3.1		
Triangle	0.0083	0.035	0	0.014	0.004	0.026	2.8	8.4

(*): result of a single run

Table 1: *Off-line performances of GAs for all four* 24×24 *test problems and different representations. The best and median result from 21 runs of 300 generations each are presented. The figures are the* L^2 *error on the displacement at the measure points: A null best error means that the actual global solution was found exactly at least once.*

resentations, with significant difference for the symmetric and checker-board problems.

- The results on the asymmetric problem demonstrates ... the asymmetry of the holes representations: the 0-rectangle is widely outperformed by the 1-rectangle on that – purposely defined – problem.
- The shape of the target subset to be matched by the holes does influence the relative results of the Holes-representations: the triangle representation outperforms the rectangle representation on the corner problem, while the reverse is clearly true on the checker-board problem.

6.4. Influence of the mesh

This section presents preliminary investigations to a posteriori justify the a priori arguments against the bitarray representation related to its dependency on the underlying mesh (see Section 5.2). Three different meshes are considered: the 24×24 mesh used in the preceding sections, together with a two times coarser 12×12 mesh and a refined 48×48 mesh are used. The same 37 loading cases are considered in the fitness (see section 4.3), and the reference displacements are computed on the same mesh than the following runs.

Tables 2 and 3 give the respective off-line results of the bitarray, the Voronoï and the 1-rectangle representations, for the asymmetric and the checker-board problems of Figure 4. Further, Figures 11 and 12 show the result obtained by the bitarray-, the Voronoï - and the rectangle-representation on the asymmetric and the checker-board problems for one run of 300 generations on the 48×48 mesh: whereas both variable-length representations obtain a solution resembling the actual structure, the bitarray representation ends up in a

	Asymmetric		Checker	
	best	median	best	median
Voronoi	0	0.00012	0.00013	0.00026
rectangle	0	0	0.00032	0.0017
bitarray	0.003	0.0054	0.00055	0.00074

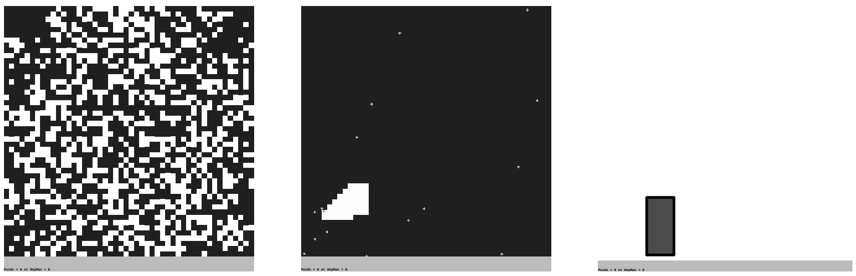
Table 2: *Off-line performances of GAs for two test problems on different representations using a 12×12 mesh after 300 generations.*

	Asymmetric		Checker	
	best	median	best	median
Voronoi	2.6e-05	0.0003	0.0005	0.0005
Rectangle	1.6e-05	4e-05	0.0015	0.0015
Bitarray	0.57	0.57	0.01	0.01

Table 3: *Partial results of off-line performances for two test problems on different representations using a 48×48 mesh. From one or three runs are aggregated here.*

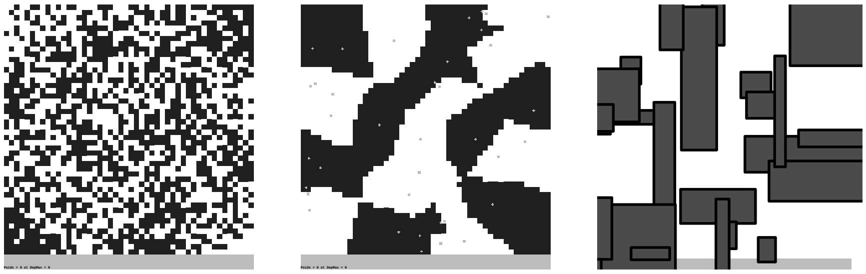
totally irrelevant result.

Although, for technical reasons (a single run for the 48×48 mesh take about 10 hours of CPU time on a powerful HP-PA8000 processor), the results for the 48×48 mesh are still incomplete, it is obvious that the performance of the bitarray representation does indeed decrease rapidly when the complexity of the mesh increases.



(a) Bitarray (b) Voronoï (c) 1-Rectangles

Figure 11: *Best structure after 300 generations on the asymmetric problem, on the* 48×48 *mesh (colors are inverted on the rectangle plot).*



(a) Bitarray (b) Voronoï (c) 1-Rectangles

Figure 12: *Best structure after 300 generations on the checker-board problem, on the* 48×48 *mesh (colors are inverted on the rectangle plot).*

7. Discussion and further work

Due to the monopolistic use of the bitarray representation in all works of the EA literature dealing with similar problems, it is necessary to emphasize here its inadequation for problems involving the search for a partition of a domain in $\mathbb{R}^n$. A priori considerations on the complexity of the representation for fine (or 3D) meshes, "Fitness Distance Correlation"-based heuristical results [32] and preliminary experimental results on the Topological Optimum Design problem [48, 49] gave a broad picture that is now clearly confirmed by the experimental results of Section 6. Note that the inclusion identification problems treated in this paper allow a finer control of the numerical experiments than the problems of Topological Optimum Design (e.g. the global optimum is known, the problem is unconstrained, ...).

The validation results in the presence of noise of Section 6.2 are of course only a first step toward using results of actual mechanical experiments as reference data. An intermediate step, part of on-going work, is to consider for reference data the results of simulations on a very fine mesh, and to test the reliability of the approach with respect to the accuracy of the numerical approximation.

Regarding the a priori choice of a representation for a given problem, the specific characteristics of the problem at hand should be taken into account: if there is strong evidence that one material is present only in small quantities, then one hole representation should be chosen, with the holes representing the minority material. Further, if the general shape of the small inclusions can be guessed, choosing elementary shapes for holes that easily fill up the supposed inclusion shapes can help the identification process. Though this is generally beyond what can be asked to mechanical experts, detailed Fitness Distance Correlation analyses [32] can be of some help here again.

On the other hand, if both materials are approximately present in the same proportions, the Voronoï representation clearly is to be preferred. Moreover, the Voronoï representation seems to be more robust to wrong guesses of actual proportions, and hence should probably be used in any doubtful case.

Another important direction of research is to address the more general problem (6), i.e. the case where the mechanical characteristics of the materials are unknown. The proposed representations can – and will – be extended easily to that framework: the label of Voronoï sites can easily be replaced by a set of real values (e.g. the E and ν coefficient of the material); in the same line, real-valued coefficients can be attached to each hole in the Hole representation. The decoded values for a point in the structure will then be the average of all values that cover it. First experiments on the problem of identification of spatial geological models in petroleum prospection [39] gave encouraging results.

8. Summary and Conclusion

This paper has presented the first results using Evolutionary Computation on an important engineering problem: non destructive inclusion identification in Structural Mechanics. In spite of theoretical results on the identifiability in the linear elastic case [9], the standard deterministic methods had to face a ill-conditioned problem, and has proved inaccurate and unstable. On the opposite, the evolutionary method demonstrates powerful on the simplified problem of linear elasticity involving two materials of known characteristics, widely outperforming the to-date results of specialized numerical methods.

However, the – almost – standard representation of structures by arrays of bits (based on a given mesh) proved to be totally inaccurate to that type of problem, and the breakthrough results were obtained using non-standard representations together with specifically designed genetic operators. The Voronoï representation uses a variable number of labeled Voronoï sites to define a partition, while the Holes representations uses a list of elementary shapes over the structure: these elementary shapes indicate which parts of the structure are not made of the default value.

First, the complexity of these representations are independent of any discretization of the structure. Hence the complexity of the algorithm itself, in terms of number of fitness evaluations, only depends on the problem at hand.

Second, the Voronoï representation seems more robust with respect to a priori knowledge about the quality of the solution: unless the solution is known to be made of very small parts of one material spread over in the structure, in which case a hole representation with default value to the majority material is the best choice, the Voronoï representation performs as good or even better than the rectangle and triangle representations tested here. Moreover, the Voronoï representation is independent of any guess regarding the shape of the inclusion itself, whereas the holes representation perform poorly if the wrong guess is made.

References

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

[2] H.-G. Beyer. Toward a theory of evolution strategies: Some asymptotical results for the $(1, +\lambda)$-theory. *Evolutionary Computation*, 1(2):165–188, 1993.

[3] H.-G. Beyer. Toward a theory of evolution strategies: The (μ, λ)-theory. *Evolutionary Computation*, 2(4):381–407, 1994.

[4] H.-G. Beyer. Toward a theory of evolution strategies: On the benefit of sex - the $(\mu/\mu, \lambda)$-theory. *Evolutionary Computation*, 3(1):81–111, 1995.

[5] H.-G. Beyer. Toward a theory of evolution strategies: Self-adaptation. *Evolutionary Computation*, 3(3):311–347, 1995.

[6] H.-G. Beyer. On the asymptotic behavior of multi-recombinant evolution strategies. In H.-M. Voigt, W. Ebeling, I. Rechenberg, and H.-P. Schwefel, editors, *Proceedings of the* 4^{th} *Conference on Parallel Problems Solving from Nature*, LNCS 1141, pages 122–131. Springer Verlag, 1996.

[7] J.-D. Boissonnat and M. Yvinec. *Géométrie algorithmique.* Ediscience International, 1995.

[8] H. D. Bui. *An introduction to inverse problems in Structural Mechanics.* CRC Press, Boca Raton, 1994.

[9] H. D. Bui. Sur quelques problèmes inverses élastiques en mécanique de l'endommagement. In *Deuxième Colloque National en Calcul des Structures*, 1995.

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

[11] 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.

[12] 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.

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

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

[15] A. Constantinescu. *Sur l'identification des modules élastiques.* PhD thesis, Ecole Polytechnique, June 1994.

[16] J. Dejonghe. Allègement de platines métalliques par algorithmes génétiques. Rapport de stage d'option B2 de l'Ecole Polytechnique. Palaiseau, Juin 1993.

[17] D. B. Fogel. An analysis of evolutionary programming. In D. B. Fogel and W. Atmar, editors, *Proceedings of the 1^{st} Annual Conference on Evolutionary Programming*, pages 43–51. Evolutionary Programming Society, 1992.

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

[19] D. B. Fogel. Phenotypes, genotypes and operators in evolutionary computation. In D. B. Fogel, editor, *Proceedings of the Second IEEE International Conference on Evolutionary Computation.* IEEE, 1995.

[20] L. J. Fogel, A. J. Owens, and M. J. Walsh. *Artificial Intelligence through Simulated Evolution.* New York: John Wiley, 1966.

[21] C. M. Fonseca and P. J. Fleming. On the performance assessment and comparison of stochastic multiobjective optimizers. In H.-M. Voigt, W. Ebeling, I. Rechenberg, and H.-P. Schwefel, editors, *Proceedings of the 4^{th} Conference on Parallel Problems Solving from Nature*, number 1141 in LNCS, pages 584–593. Springer Verlag, Sept. 1996.

[22] P.L. George. *Automatic mesh generation, application to Finite Element Methods.* Wiley & Sons, 1991.

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

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

[25] J. J. Grefenstette. Optimization of control parameters for genetic algorithms. *IEEE Trans. on Systems, Man and Cybernetics*, SMC-16, 1986.

[26] J. J. Grefenstette. Virtual genetic algorithms: First results. Technical Report AIC-95-013, Navy Center for Applied Research in Artificial Intelligence, February 1995.

[27] J. H. 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] T. Jones and S. Forrest. Fitness distance correlation as a measure of problem difficulty for genetic algorithms. In L. J. Eshelman, editor, *Proceedings of the* 6^{th} *International Conference on Genetic Algorithms*, pages 184–192. Morgan Kaufmann, 1995.

[30] K. E. Kinnear Jr. A perspective on gp. In Jr K. E. Kinnear, editor, *Advances in Genetic Programming*, pages 3–19. MIT Press, Cambridge, MA, 1994.

[31] A. B. Kahng and B. R. Moon. Toward more powerful recombinations. In L. J. Eshelman, editor, *Proceedings of the* 6^{th} *International Conference on Genetic Algorithms*, pages 96–103. Morgan Kaufmann, 1995.

[32] L. Kallel and M. Schoenauer. Fitness distance correlation for variable length representations. Submitted, 1996.

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

[34] 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.

[35] 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*, number 1063 in LNCS. Springer Verlag, Septembre 1995.

[36] C. Kane and M. Schoenauer. Topological optimum design using genetic algorithms. *Control and Cybernetics, Special Issue on Optimum Design*, 25(5), 1996.

[37] R. V. Kohn and A. McKenney. Numerical implementation of a variational method for electric impedance tomography. *Inverse Problems*, 6:389–414, 1990.

[38] J. R. Levenick. Inserting introns improves genetic algorithm success rate : Taking a cue from biology. In R. K. Belew and L. B. Booker, editors, *Proceedings of the* 4^{th} *International Conference on Genetic Algorithms*, pages 123–127, 1991.

[39] D. Luu. Algorithmes génétiques et diagrammes de voronoî : Optimisation et identification de modèles géologiques, Sept. 1996. Rapport de DEA d'Analyse Numérique de l'Université de Paris VI.

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

[41] F. P. Preparata and M. I. Shamos. *Computational Geometry: an introduction*. Springer Verlag, 1985.

[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 L. D. Whitley and M. D. Vose, editors, *Foundations of Genetic Algorithms 3*, pages 51–72. Morgan Kaufmann, 1995.

[44] I. Rechenberg. *Evolutionstrategie: Optimierung Technisher Systeme nach Prinzipien des Biologischen Evolution*. Fromman-Holzboog Verlag, Stuttgart, 1973.

[45] G. Rudolph. Convergence analysis of canonical genetic algorithm. *IEEE Transactions on Neural Networks*, 5(1):96–101, 1994.

[46] G. Rudolph. Convergence of non-elitist strategies. 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 63–66. IEEE Press, 1994.

[47] J. D. Schaffer, R. A. Caruana, L. Eshelman, and R. Das. A study of control parameters affecting on-line performance of genetic algorithms for function optimization. In J. D. Schaffer, editor, *Proceedings of the* 3^{rd} *International Conference on Genetic Algorithms*, pages 51–60. Morgan Kaufmann, 1989.

[48] 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.

[49] 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*, pages 121,129. MIT Press, 1996.

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

[51] D. Thierens and D.E. Goldberg. Mixing in genetic algorithms. In S. Forrest, editor, *Proceedings of the* 5^{th} *International Conference on Genetic Algorithms*, pages 38–55. Morgan Kaufmann, 1993.