Applied Intelligence 16, 139–155, 2002

Compact Unstructured Representations for Evolutionary Design

HATEM HAMDA AND FRANÇOIS JOUVE
CMAP—UMR CNRS 7641, École Polytechnique, 91128, Palaiseau Cedex, France

EVELYNE LUTTON
Projet Fractales—INRIA, Domaine de Voluceau, BP 105, 78153, Le Chesnay Cedex, France

MARC SCHOENAUER
CMAP—UMR CNRS 7641, École Polytechnique, 91128, Palaiseau Cedex, France

MICHÈLE SEBAG
LMS—UMR 7649, École Polytechnique, 91128, Palaiseau Cedex, France

Abstract. This paper proposes a few steps to escape structured extensive representations for objects, in the context of evolutionary Topological Optimum Design (TOD) problems: early results have demonstrated the potential power of Evolutionary methods to find numerical solutions to yet unsolved TOD problems, but those approaches were limited because the complexity of the representation was that of a fixed underlying mesh. Different compact unstructured representations are introduced, the complexity of which is self-adaptive, i.e. is evolved by the algorithm itself. The Voronoi-based representations are variable length lists of alleles that are directly decoded into object shapes, while the IFS representation, based on fractal theory, involves a much more complex morphogenetic process. First results demonstrates that Voronoi-based representations allow one to push further the limits of Evolutionary Topological Optimum Design by actually removing the correlation between the complexity of the representations and that of the discretization. Further comparative results among all these representations on simple test problems seem to indicate that the complex causality in the IFS representation disfavors it compared to the Voronoi-based representations.

Keywords: structural design, variable length representation, adaptive complexity

1. Introduction

Evolutionary Algorithms (EAs) have been widely used in the framework of parametric optimization, i.e. when the search space is a *structured* space of fixed length vectors. In that context, EAs are just yet another optimization method: They are indeed a powerful zero-th order global optimization method, and, as such, they have been successfully applied in many domains.

But the most innovative and outstanding results have been recently obtained by taking advantage of the ability of EAs to deal with very unusual *unstructured* search spaces, such as spaces of parse-trees [1–3], of unordered lists, of graphs, and the like [4, 5]. Indeed, structured search spaces are more likely to reflect the unavoidable biases of the programmer, and hence to limit the creativity of the overall design process. On the other hand, evolving complex representations opens up unsuspected regions of huge search spaces. The "only" prerequisites are an initialization procedure and variation operators that respect some minimal requirements with respect to the problem at hand [6].

One first step away from structured representations is to use sparse variable-length lists instead of full

extensive descriptions[1]: for instance, when searching in the space of polynomials (of several variables), the extensive structured representation would be to look for the vector coefficients of all monomials up to a certain degree; an unstructured representation can be that of variable length lists of coefficients, describing only some particular monomials. The critical issue of such *compact unstructured* representations is of course the design of meaningful variation operators (crossover, mutation) that will make evolution differ from random search.

Another further step toward scalability is to use lists or groups of elementary items, also called component-based representations in [7]: in the context of polynomial identification, that would amount to manipulate some elementary polynomials not limited to simple monomials. But such compact unstructured representations can also be organized into . . . structured spaces, to make their evolution easier: using Genetic Programming [8, 9] is an alternative representation for variable degree polynomials, with well-designed variation operators. Such search space also allows one to add useful features, such as modularity and recursion, to the representations [1], making another step toward the evolution of complex solutions: for instance, when the solution to a problem is known to have some symmetries, it seems at least resource-wasting, and at worse bound to failure to let evolution "discover" multiple instances of the same optimal half-solution.

But the to-date ultimate research direction toward the evolution of complex solutions seems to lie in the so-called *morphogenetic* approach [5]: instead of evolving parts of solutions (simple item or more complex components), one evolves some programs, or rules, that in turn give the solution when they are executed or applied. One of the early attempts of morphogenetic approach is the Cellular Encoding of F. Gruau [10] where a Neural Network is built from an embryo by a GP-like program—while many recent successes have been reported using GP in different domains [2, 11]. As they also can evolve modular solutions, morphogenetic approaches really are appealing to build very complex solutions to difficult problems whose components can hardly be designed directly. However, the increase in scalability goes together with a loss in causality: it is almost impossible for anyone to guess the influence of small parts of the genotype on the final solution.

In the framework of Topological Optimum Design, the plain direct extensive representation is the widely used bitarray approach based on a fixed mesh of the design domain. Though very successful to overcome the main limitations of deterministic methods for TOD [12–14], this representation does not scale up with the complexity of the mesh. Different compact unstructured representations based on Voronoi diagrams are introduced, that exhibit a self-adaptive complexity (i.e. the complexity of the solutions is adjusted by the algorithm). These representations do not exactly involve components, but do require some elementary alleles to be defined by the programmer; such alleles can be viewed as some sort of variable components: due to the high degree of epistasis of those representations, the phenotypic expression of each allele strongly depends on the other alleles. In an attempt to avoid the biases resulting from the manual choice of these alleles, the IFS representation, a morphogenetic approach based on fractal theory, is defined.

The paper is organized the following way. The context of evolutionary TOD is briefly recalled in Section 2, from the mechanical background to the adaptive penalty method used within the fitness function. Section 3 introduces a series of three different representations based on the idea of Voronoi diagrams while Section 4 presents original experimental results obtained with those representations, assessing the power of the compact unstructured approach. Comparative results on cantilever benchmark problems are then presented, allowing one to discriminate among those representations. Section 5 introduces the IFS representation, based the fractal theory, together with preliminary experimental results assessing its possible advantages and setting its limits, at least for simple problems of TOD. Section 6 discusses the relevance of the different representations introduced in the paper and concludes on further directions of research.

2. Optimum Design of Mechanical Structures

2.1. The Mechanical Problem

The general framework of this paper is the Topological Optimum Design (TOD) problem: find 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—here a bound on the maximal displacement under a prescribed loading, but it could also involve bounds on the eigenfrequencies, or any combination of stiffness and modal optimization. The optimality criterion is here the

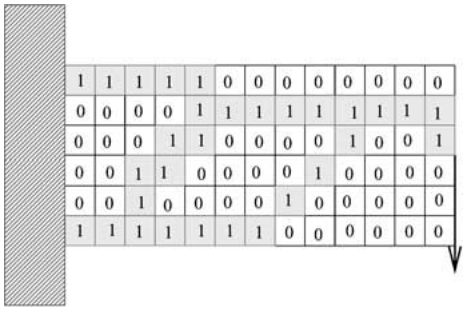


Figure 1. The 2×1 cantilever plate test problem, and a bitarray representation of a structure derived from a regular 13×6 mesh. See Section 2.4.

weight of the structure, but it could also involve other technological costs.

The mechanical model used in this paper is the standard two-dimensional (except in Section 4.4) plane stress linear model, and only linear elastic materials will be considered (see e.g. [15]). All mechanical figures are adimensional (e.g. the Young modulus is set to 1) and the effects of gravity are neglected.

One of the most popular benchmark problem of Optimum Design is the optimization of a cantilever plate: the design domain is rectangular, the plate is fixed on the left vertical part of its boundary, 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. *State of the Art in Shape Optimization*

The main trends in structural optimization can be sketched as follows. A first approach is that of *domain variation* [16] (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. The original approach had 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). However, the idea of topological gradient was recently proposed and successfully used in [17], allowing the modification of the topology of the solution. Nevertheless, this method is strictly limited to the linear elasticity framework.

The other method for topology optimization is the now standard approach of homogenization, introduced in [18], which deals with a continuous density of material in [0, 1]. This relaxed problem is known to have a solution in the case of linear elasticity [19]—and the corresponding numerical method does converge to a (non-physical) generalized solution. That solution must then be post-processed to obtain an admissible solution with boolean density [20]. The homogenization method is also insofar limited to linear-elasticity. The theoretical results about optimal micro-structures only handle single-loading cases, though numerical solution to multi-loading cases have been proposed [21]. In addition, this method cannot address loadings that apply on the (unknown) actual boundary of the shape (e.g. uniform pressure).

A possible approach to overcome these difficulties of TOD is to use stochastic optimization methods. Stochastic optimization methods have been successfully applied to other problems of structural optimization: in the framework of discrete truss structures, for cross-section sizing [22, 23] among others, as well as for topological optimization [24, 25] and for the optimization of composite materials [26].

TOD problems have also already been addressed by stochastic methods: Simulated Annealing has been used to find the optimal shape of the cross-section of a beam in [27]; and Evolutionary Algorithms have been used to solve cantilever problems as the one presented in Section 2.1 in [12, 28, 29].

The above-mentioned limitations of the deterministic methods have been successfully overcome by these works—in [12–14] for instance, results of TOD in nonlinear elasticity, as well as the optimization of an underwater dome (where the loading is applied on the unknown boundary) have been proposed, both out of reach for the deterministic methods.

2.3. *Fitness Computation*

The problem tackled in this paper is to find a structure of minimal weight such that its maximal displacement stays within a prescribed limit D_{lim} when some given point-wise force is applied on the loading point (see Fig. 1). The computation of the maximal displacement is made using a Finite Element Analysis solver [30].

From mechanical considerations, all structures that do not connect the loading point and the fixed boundary are given an arbitrary high fitness value. Moreover, the material in the design domain that is not connected to the loading point—and has thus no effect on the mechanical behavior of the structure—is discarded during

the Finite Element Analysis, but slightly penalizes the structure at hand (see [12, 13] for a detailed discussion on both these issues). In summary, for connected structures, the problem is to minimize the (connected) weight subject to one constraint for each loading case, namely $D^i_{Max} \leq D^i_{lim}$, where D^i_{Max} its maximal displacement computed by the FEM under loading i, and D^i_{lim} its prescribed limit.

Introducing the positive *penalty parameters* α_i, the fitness function to minimize is

$$Weight + \sum_i \alpha_i \left(D^i_{max} - D^i_{lim}\right)^+ \qquad (1)$$

However, adjusting α_i is not an easy task, and many specific methods exist in Evolutionary Computation [31].

The adaptive penalty method used here updates the penalty parameter based upon global statistics of feasibility in the population. Its main goal is to explore the neighborhood of the boundary of the feasible region by trying to keep in the population individuals that are on both sides of that boundary: in the context of stiffness optimization in TOD, the solution does lie on the boundary, . . . but for the continuous problem only! Once discretized, this is no longer true, and it can only be said that the solution lies close to the boundary.

The objective is to maintain in the population a minimum proportion of feasible individuals as well as a minimum proportion of infeasible individuals. Denote by $\Theta^k_{feasible}$ the proportion of feasible individuals at generation k, and by Θ_{inf} and Θ_{sup} two user-defined parameters. As small penalty parameters favor the infeasible individuals (and vice-versa), the following update rule for the α_i parameters is proposed to try to keep $\Theta^k_{feasible}$ in $[\Theta_{inf}, \Theta_{sup}]$:

$$\alpha_{k+1} = \begin{cases} \beta \cdot \alpha_k & \text{if } \Theta^k_{feasible} < \Theta_{inf} \\ (1/\beta) \cdot \alpha_k & \text{if } \Theta^k_{feasible} > \Theta_{sup} \\ \alpha_k & \text{otherwise} \end{cases} \qquad (2)$$

with $\beta > 1$. User-defined parameters of this method are Θ_{inf}, Θ_{sup}, β and the initial value α_0. The robust values $\beta = 1.1$, $\Theta_{inf} = 0.4$, and $\Theta_{sup} = 0.8$ were used in all experiments presented this paper.

Note that the variations of α are non monotonous, and hence there is no a priori guarantee that the best individual in the population is feasible. It can even happen that the population contains no feasible individual—though in that case the steady increase of α should favor individuals with lower constraint violation, and rapidly result in the emergence of feasible individuals.

Some comparative results assessing the power of that population-based adaptive penalty method can be found in [32] for test problems, and in [33] in the context of TOD.

2.4. Representations of Structures for TOD

All the works cited in Section 2.2 that address TOD problems with EAs use the same 'natural' binary representation, termed *bitarray* in [12]: it relies on a mesh of the design domain—the same mesh that is used to compute the mechanical behavior of the structure in order to give it a fitness (see Section 2.3). Each element of the mesh is given value 1 if it contains material, 0 otherwise (see Fig. 1). Note that this bit-based representation is not equivalent to the usual bitstring representation, and that some specific geometrical crossover operators had to be designed [34], similar to the crossover operator described in Section 3.1 below for the Voronoi-based representations.

In spite of its successes in solving TOD problems [12–14], bitarray representation suffers from a strong limitation due to the dependency of its complexity on that of the underlying mesh. Indeed, the size of the individual (the number of bits used to encode a structure) is the size of the mesh. Unfortunately, according to both theoretical results [35] and empirical considerations [36], the critical population size required for convergence should be increased at least linearly with the size of the individuals. Moreover, larger populations generally require a greater number of generations to converge. Hence it is clear that the bitarray approach will not scale up when using very fine meshes. This greatly limits the practical application of this approach to coarse (hence imprecise) 2D meshes, whereas Mechanical Engineers are interested in fine 3D meshes!

These considerations appeal for some more compact representations whose complexity does not depend on a fixed discretization. The ultimate step in the direction of complexity-free representation is to let the complexity itself evolve and be adjusted by the EA.

3. Voronoi-Based Representations

The Voronoi representation is a first attempt toward unstructured representations for TOD. It has first been proposed in [37], but has since then been used mainly in the context of identification problems [38, 39]. This

section recalls the definition of Voronoi representation, and proposes two other representations that also derive from the same ideas.

3.1. Voronoi Representation

3.1.1. Voronoi Diagrams. Consider a finite number of points $V_0, \ldots, V_N$ (the *Voronoi 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 Voronoi site is V_i, termed *Voronoi cell*. The *Voronoi diagram* is the partition of the design domain defined by the Voronoi 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 Voronoi diagram of at least one set of Voronoi sites (see [40] for a detailed introduction to Voronoi diagrams, and a general presentation of algorithmic geometry).

3.1.2. The Genotype. Consider now a (variable length) list of Voronoi sites, each site being labeled 0 or 1. The corresponding Voronoi diagram represents a partition of the design domain into two subsets, if each Voronoi cell is labeled as its associated site (see Fig. 2(a)).

3.1.3. Decoding. Of course, as some FE analysis is required during the computation of the fitness function, and as re-meshing is a source of numerical noise that could ultimately take over the actual difference in mechanical behavior between two very similar structures, it is mandatory to use the very same mesh for all structures at the same generation. A partition described by Voronoi sites is easily mapped on any mesh: the subset (void or material) an element belongs to is determined from the label of the Voronoi cell in which the center of gravity of that element lies.

However, the complexity of the individuals (i.e. the number of Voronoi sites in their representation) is totally independent of the choice of the mesh used for fitness computation, and will evolve according to the Darwinian principles underpinning the whole evolutionary process.

3.1.4. Initialization. The initialization procedure for the Voronoi representation is a uniform choice of the number of Voronoi sites between 1 and a user-supplied maximum number, a uniform choice of the Voronoi sites in the design domain, and a uniform choice of the boolean label.

3.1.5. Variation Operators. The variation operators for the Voronoi representation are problem-driven:

- The *crossover operator* exchanges Voronoi sites on a geometrical basis. In this respect it is similar to the specific bitarray crossover described in [34]. Figure 3 is an example of application of this operator.
- The *mutation operator* is chosen by a roulette wheel selection based on user-defined weights among the following operators (Fig. 4):
  - the *displacement mutation* performs a Gaussian mutation on the coordinates of the sites. As in Evolution Strategies [41], adaptive mutation is used: one standard deviations is attached to each coordinate of each Voronoi site, undergoes log-normal mutation before being used for the Gaussian mutation of the corresponding coordinate.
  - the *label mutation* randomly flips the boolean attribute of one site.
  - the *add* and *delete mutations* are specific variable-length operators that respectively randomly add or remove one Voronoi site on the list.

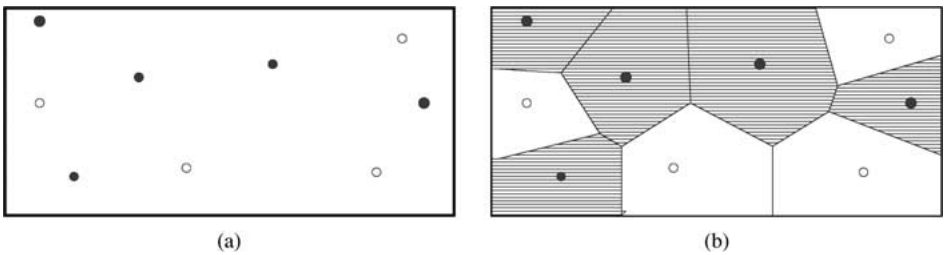


Figure 2. Voronoi representation on a 2×1 design domain. (a) The genotype: a list of labeled Voronoi sites. Black dots are sites with label 0 and white dots are sites with label 1. (b) The phenotype: the Voronoi cells receive the label of the corresponding site, and build a partition of the design domain.

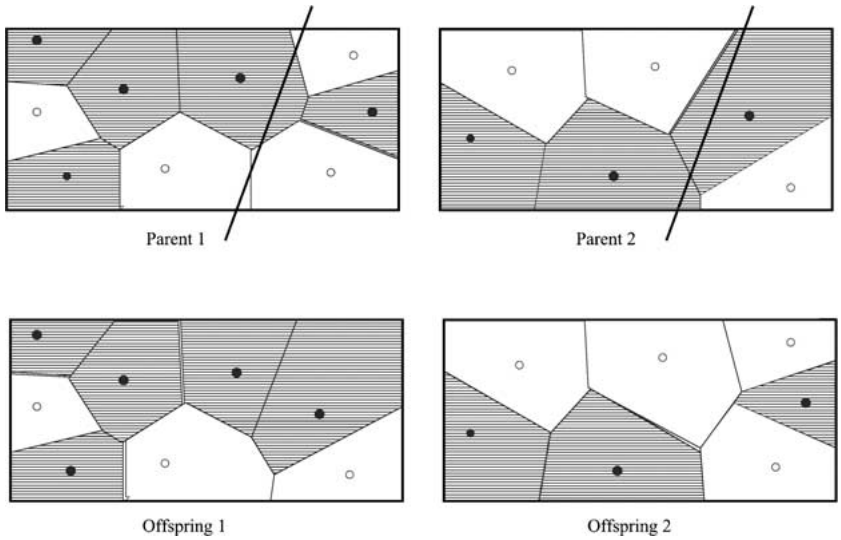


Figure 3. The crossover operator for the Voronoi representation: a random line is drawn across both diagrams, and the sites on either side are exchanged.

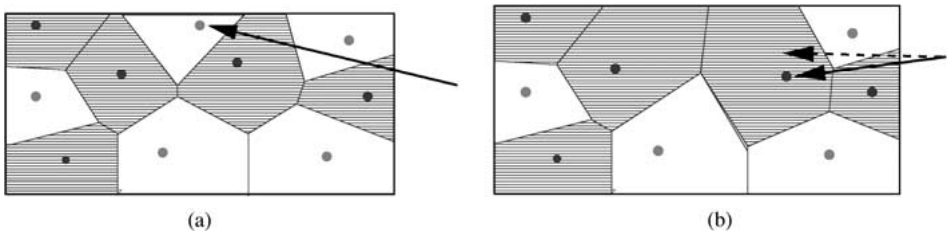


Figure 4. Two mutations for Voronoi representation. (a) The *add* mutation: the site at end of the arrow has been added to the genotype of Fig. 2(a). The phenotype is rather different from that of Fig. 2(b). (b) Mutation by site displacement: a small displacement of one site in Fig. 2(a) slightly modified the phenotype (see Fig. 2(b)).

3.1.6. Boundary Control. One crucial problem in TOD is the fine tuning of the boundary of the solution. The optimal shape can only be reached in reasonable time if the algorithm is able to precisely control the boundaries of the individuals in the population. Unfortunately, the Voronoi representation only offers indirect control of the boundary of the structure it represents. Moreover, the high epistasis of that representation makes it difficult to modify a single boundary without disturbing the adjacent ones. The idea behind the *dipole* representation presented in next section is to try to overcome that difficulty.

3.2. Dipole Representation

3.2.1. Dipoles. A dipole is a set of two Voronoi sites, one labeled 0 and the other labeled 1, standing almost at the same point in the design domain, but whose median has a prescribed angle in the plane. A dipole is hence defined by three real-valued variables, its coordinates (x, y) and the angle of its median with the x-axis θ. Figure 5(a) is an example of a dipole. The direct control over θ allows a precise control over that part of the boundary that goes through the (x, y) point.

3.2.2. The Genotype. One individual in the dipole representation is a (variable length) list of dipoles. As in the Voronoi representation, the corresponding Voronoi diagram represents a partition of the design domain into two subsets, as can be seen on Fig. 5(a).

3.2.3. Decoding. Considering each dipole as a single Voronoi site, the corresponding Voronoi diagram is drawn. Each dipole is then turned back into two Voronoi

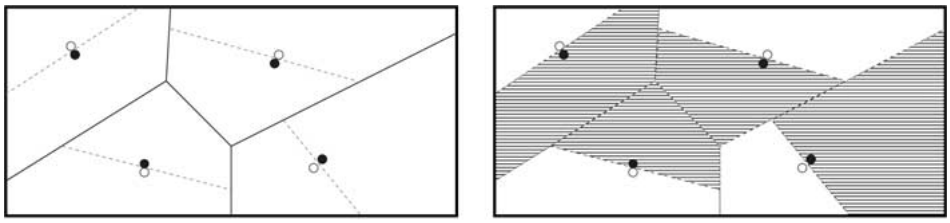


Figure 5. Dipole representation on a 2×1 design domain. (a) The genotype: the two sites of each dipole lie in fact at the same point of the design domain. The median are the dashed lines, the Voronoi cell boundaries the solid lines. (b) The phenotype: the medians are directly controlled by the variation operators, but many actual boundaries between void and material are made by the—uncontrolled—Voronoi boundaries.

sites of opposite color, the median being given by its angle θ with the *x-axis* (both the Voronoi cells and the median are represented on Fig. 5(a)). The full phenotype is then drawn, as on Fig. 5(b).

However, as can be seen on Fig. 5, the decoding of adjacent dipoles shows that the resulting structure has two kinds of boundaries: the median of the dipoles, which can hopefully be controlled by the evolutionary algorithm, and the medians between dipoles, whose fine tuning will be as difficult as in the Voronoi representation—and maybe even more, as some weird configurations will often arise, as the one shown in Fig. 5(b).

3.2.4. Variation Operators. These operators for the dipole representation are derived from the ones of the Voronoi representation: the initialization procedure chooses a number of dipoles, and initializes their coordinates uniformly in the design domain and their angle in $[0, 2\pi[$. The crossover operator exchanges dipoles exactly as its counterpart for Voronoi representation exchanged Voronoi sites (see Fig. 3). The mutation operators include the displacement mutation, the Gaussian mutation of the angle of a dipole, and of course the addition and destruction of dipoles in the list.

3.2.5. Truss-Like Structures. For cantilever problems, it is well-known that the best structures are in fact truss structures. Obtaining truss structures using Voronoi diagrams or dipoles requires the emergence of coupled subsets of either sites or dipole and thus might take some time to evolve.

Moreover, the defects of the dipole representation (see Fig. 5(b)), together with experimental results as the ones of Sections 4.2, demonstrate its inability to achieve the fine tuning of the boundary that was the main reason why it was designed.

The idea behind the Voronoi-bar representation, introduced in next section, is precisely to both achieve the fine tuning of the boundary, and favor the evolution of truss structures by providing alleles that already are truss elements.

3.3. Bar Representation

3.3.1. Voronoi-Bars. A Voronoi-bar is hence defined by four real-valued variables, its coordinates (x, y), the angle of the bar with the x-axis θ and its width. Figure 6(a) is an example of a single Voronoi-bar.

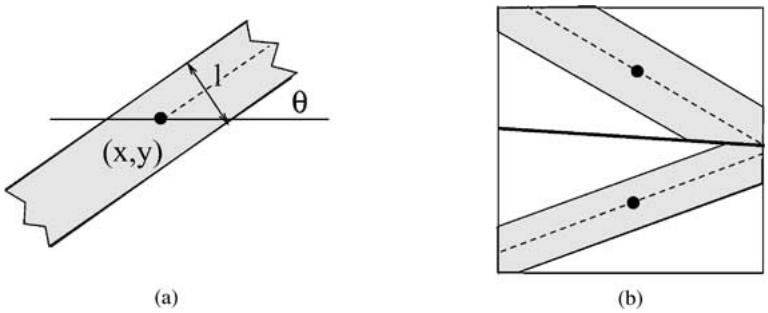


Figure 6. The Voronoi-bar representation. A single bar (a) and the structure built using two such bars (b): The thick line is the boundary between the two Voronoi cells and is part of the structure boundary only at the junction of the two bars.

3.3.2. The Genotype. One individual in the Voronoi-bar representation is a (variable length) list of Voronoi-bars. When all Voronoi-bars are simply considered as Voronoi sites, the corresponding Voronoi diagram represents a partition of the design domain into convex polygons. Each such polygon is then separated into two subdomains, namely the central part, made of material, and the outer part, "filled" with void (see Fig. 6). Whenever the width is large enough, the whole cell is 1, whereas a null value for the width turns the cell into a 0 cell: these extreme cases of the Voronoi-bar representation are nothing else than the Voronoi representation itself.

3.3.3. Decoding. As for the Voronoi representation, the fitness of all structures will be evaluated using a fixed mesh, and the projection on that fixed mesh is performed as in Section 3.1: an element is considered made of material if and only if its center of gravity falls within the material part of a Voronoi-bar.

As can be seen on Fig. 6(b), the decoding of adjacent Voronoi-bars allows to directly control almost the whole boundary of the resulting structure, apart from some limited portions at the junction of two "bars".

3.3.4. Variation Operators. These operators for the Voronoi-bar representation are once again derived from the ones of the Voronoi representation: the initialization procedure chooses a number of bars, and initializes their coordinates, angles and width uniformly. The crossover operator exchanges bars exactly as its counterpart for Voronoi representation exchanged Voronoi sites (see Fig. 3). The mutation operators include the displacement mutation, the Gaussian mutation of the angle and width of a bar, and of course the addition and destruction of bars in the list.

4. Experimental Results for Voronoi-Based Representations

This section introduces some results obtained using the Voronoi-based representations. Mesh-dependency experiments were run on the Voronoi representation to ensure the idea of compact unstructured representation was indeed playing its role: this was shown to be the case up to the error in discretization [33]. The most important part of this section deals with comparative results on the benchmark cantilever problems to try to assess the usefulness of the introduction of the other Voronoi-based representations. This section ends with some original results on a 3D cantilever problem, demonstrating that such unstructured representations did indeed allow innovative results in Evolutionary Topological Optimum Design.

4.1. Evolutionary Experimental Conditions

Unless otherwise stated, the experiments presented further on have been performed using the following settings: Standard GA-like evolution (linear rank-based selection and generational replacement of all parents by all offspring) with populations size 80; At most 40 Voronoi sites (or dipoles or bars) per individual; Crossover rate is 0.6 and mutation rate per individual is 0.3; Weights among the different mutations are 0.5 for the displacement mutation, the remaining mutations equally sharing the remaining 0.5; All runs are allowed at most 2000 generations, and the algorithm stops after 300 generations without improvement; All plots are the result of 21 independent runs; All CPU times are given related to a Pentium III processor running at 300 MHz under Linux. For instance, the cost of one generation for the 1×2 or the 2×1 cantilever problems discretized with 200 elements is 2 s.

4.2. Comparative Results of Voronoi-Based Representations

This section presents comparative benchmark results on the three Voronoi-based representations. Two benchmark problems are considered: the 1×2 and 2×1 cantilever plates with respective limits on the maximal displacement of 20 and 220. In both cases, the vertical left boundary is fixed, and the point-wise force is applied at half-height of the right vertical boundary. The experimental conditions for all representations are those described in Section 4.1.

Figures 7–9 show typical best structures obtained with respectively the Voronoi, the dipole and the Voronoi-bar representations, while Figs. 10 and 11 show statistics over 21 runs for both those test cases.

The first conclusion of these experiments is that all three representations find almost equally good solutions among the 21 runs. However, the best representation according to the quality criterion is the Voronoi-bar representation: almost all solutions were similar to the ones of Fig. 9, whereas many solutions found by the dipole representation were much worse, and the solutions found by the Voronoi representation were

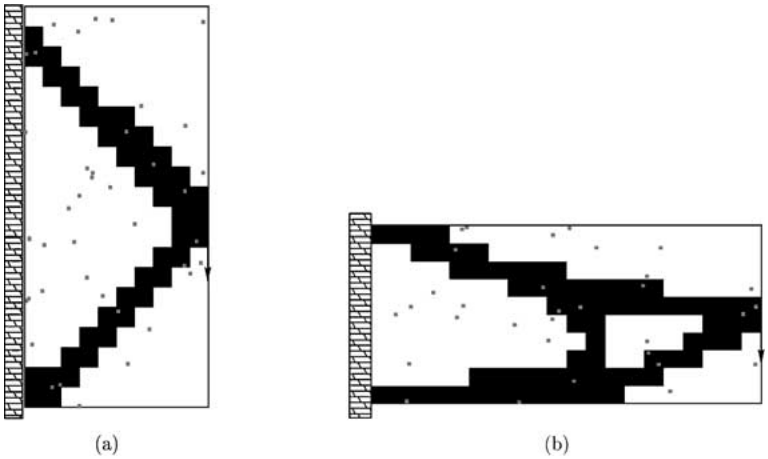


Figure 7. The two best benchmark results for the Voronoi representation. (a) weight = 0.215, 35 sites (b) weight = 0.35, 32 sites.

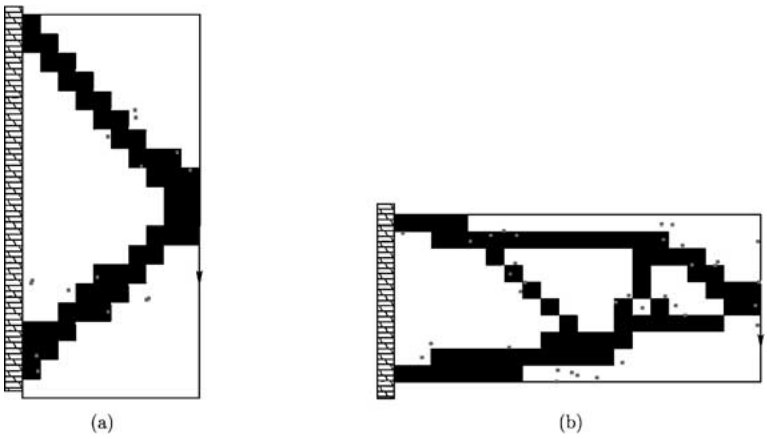


Figure 8. The two best benchmark results for the dipole representation. (a) weight = 0.215, 15 dipoles (b) weight = 0.325, 36 dipoles.

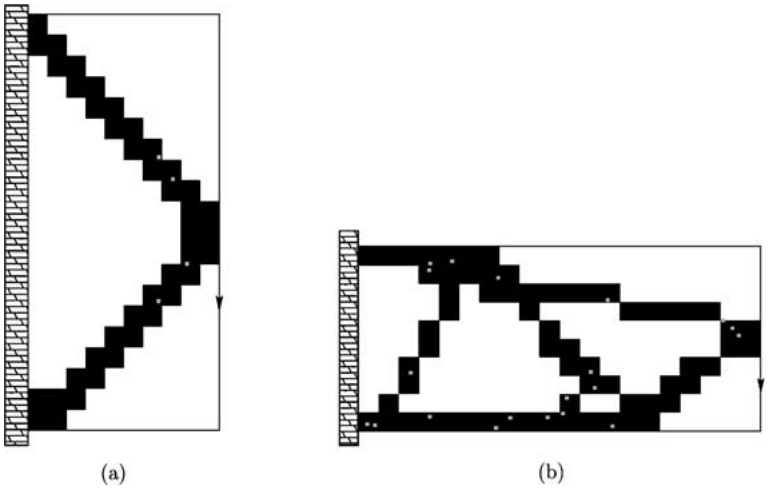
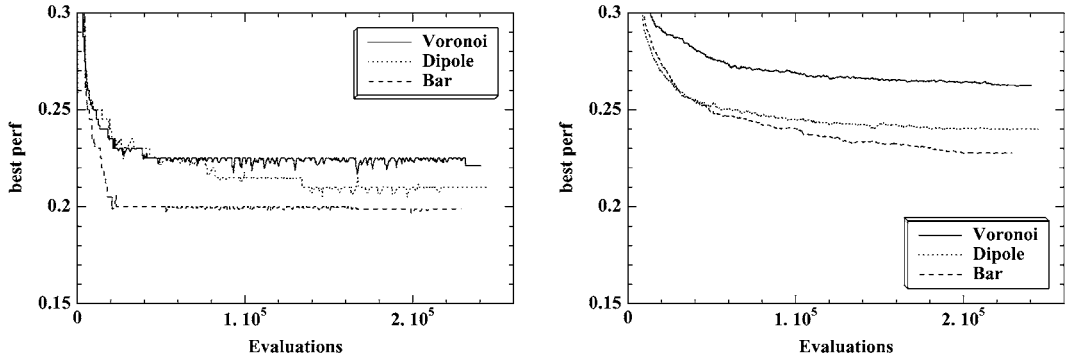


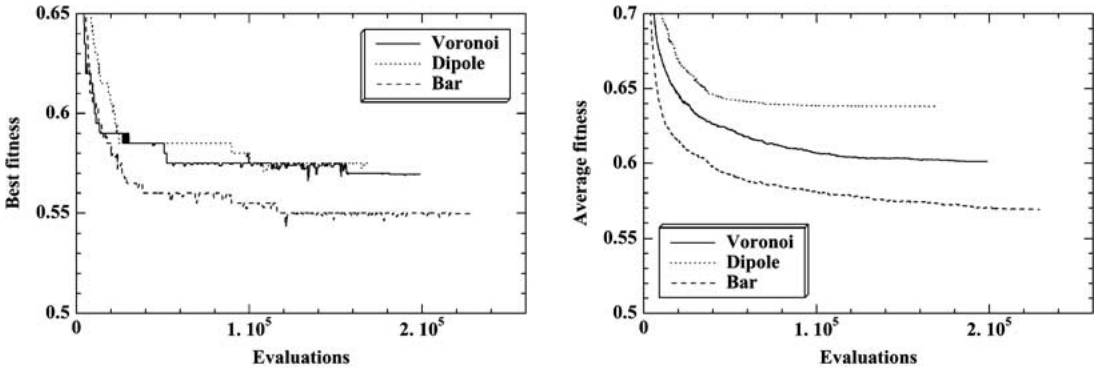
Figure 9. The two best benchmark results for the Voronoi-bar representation. (a) weight = 0.2, 4 bars (b) weight = 0.33, 20 bars.



Best fitness values averaged over 21 runs

Average fitness values averaged over 21 runs

Figure 10. Comparative on-line results for the Voronoi-based representations on the 1×2 cantilever for $D_{lim} = 20$.



Best fitness values averaged over 21 runs

Average fitness values averaged over 21 runs

Figure 11. Comparative on-line results for the Voronoi-based representations on the 2×1 cantilever for $D_{lim} = 220$.

consistently slightly worse. These trends are reflected on the plots shown in Figs. 10 and 11. Note that both Voronoi and dipole representations sometimes showed results similar to the Voronoi-bar representation, but the latter really appeared more robust.

Another criterion is the complexity of the solutions. The test cases are here very simple, and the solutions should reflect this simplicity. Here again the Voronoi-bar representation is a clear winner: In all runs, the Voronoi-bar representation found very compact solutions, compared with those found by the other representations. The perfect 2-bars V-shape was even found once for the 1×2 cantilever problem, even though there was no incentive in the algorithm for to decrease the number of Voronoi sites. But less Voronoi sites are probably easier to fine tune, and this might be the explanation for that.

Hence it seems that the additional complexity in the elementary alleles of the Voronoi-bar representation does pay off, at least on these benchmark problems.

4.3. The 10×1 *Cantilever*

The problem of the 10×1 cantilever (discretized using a 100×10 regular mesh) proved to be difficult for the bitarray representation as it raises an additional difficulty: most of initial random structures do not connect the fixed boundary and the point where the loading is applied. Hence an alternate initialization procedure is used, where the average weight of random structures can be tuned (see [42] for details). Furthermore, the maximal number of sites for each individuals is

Figure 12. Optimal structure on the 100 × 10 mesh for 10 × 1 cantilever plate for the *Voronoi* representation. $D_{lim} = 12$, number of cells = 105. weight = 0.479. CPU time = 14 s/gen.

Figure 13. Optimal structure on the 100 × 10 mesh for 10 × 1 cantilever plate for the *Voronoi-bar* representation. $D_{lim} = 12$, number of cells = 91. weight = 0.424. CPU time = 14 s/gen.

increased to 120, and the best results were obtained with a population size of 120.

Nevertheless, the dipole representation was unable to find satisfactory solutions—in most cases, it simply could never find a connected solution, similarly to the bitarray representation.

Figures 12 and 13 shows the most significant results obtained using respectively the Voronoi and the Voronoi-bar representations.

Again, a slight advantage can be seen for the Voronoi-bar representation in the quality of the best solution. However, the advantage in solution complexity is not so clear than it was on the 1 × 2 benchmark. But a very interesting feature is the quasi-regularity of the Voronoi-bar solution: indeed, any mechanical engineer would build such a structure by using the same part four or five times before ending with some specific part at the further end (think of how cranes are designed). But as the Voronoi-based representations do not have the ability to evolve modularity, such partial solutions have to be evolved six times. On-going work addresses this issue by introducing hierarchical representations based on the elementary Voronoi representations introduced in Section 3.

4.4. Three-Dimensional Problem

This section demonstrates that the Voronoi representation can indeed be applied to represent three-dimensional objects. Because the Voronoi diagrams theory is valid in any dimension, the extension of the representation defined in Section 3.1 to three dimensional objects is straightforward—note that this is true for the dipole representation, too (Section 3.2), but that the bar representation (3.3) will require some work, as multiple elementary geometrical shapes should be designed (e.g. 3D bars of different sections).

The test problem is the 3D equivalent of the cantilever benchmark problem described in Section 2.1: The design domain is a quadrangle subset of $\mathbb{R}^3$, the structure is fixed on a vertical plane, and a force is applied in the center of its opposite face (see Fig. 14). The problem is symmetrical with respect to a vertical plane perpendicular to the fixed wall. Hence only half of the domain is discretized, according to a $16 \times 7 \times 10$ mesh. Its left face is fixed, and the loading is applied on the middle of the right face.

The first experiments presented here were performed only with the Voronoi representation (Section 3.1). As for large 2D domains (Section 4.3), the higher complexity of the problem lead to modify the settings: the population size is again set to 120 and the maximum number of Voronoi sites is also increased to 120.

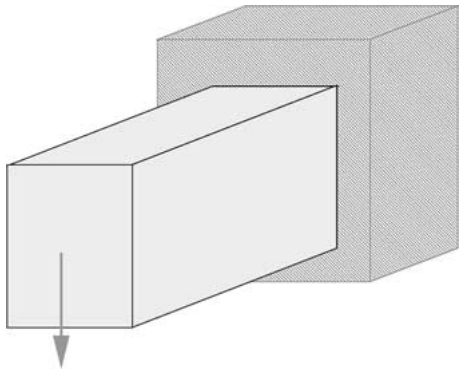
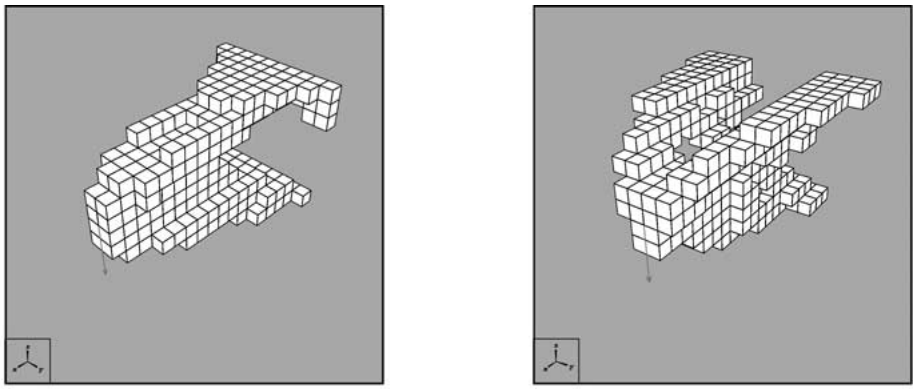


Figure 14. The design domain for the 3-dimensional cantilever problem.



(a) (b)

Figure 15. Two results for the symmetrical three-dimensional problem using a $16 \times 7 \times 10$ mesh for half of the structure, with same constraint (CPU time = 6 mn/gen). The point of view is that of Fig. 14, i.e. the structure is fixed on a vertical wall at the back of the figure (not represented). (a) weight = 0.15178, 103 sites (b) weight = 0.166, 109 sites.

Figure 15 demonstrates that the algorithm was able to find some good solutions in . . . a few days of CPU time (3D FEM analyses are far more costly than 2D for the same mesh size due to higher connectivity density). To the best of our knowledge, such results are the first results of 3D TOD obtained using Evolutionary Computation.

Moreover, it also stresses the ability of EAs to find multiple quasi-optimal solutions to the same problem, some of them quite original indeed when compared to the results of the homogenization method on the same problem.

5. IFS Representation

The Voronoi-based representations were some attempts to escape the direct encoding of discretized structures using a predefined mesh. However, the basic blocks that build the structure had to be designed by the programmer, and wrong choices can bias the search in a wrong direction, and hence once again limit the creativity of the overall process.

The following fractal-based representation is an attempt to go further in the morphogenetic direction: no assumption is made about what the building blocks of a structure could be—but the search space for the genotype is hopefully rich enough so that a large number of different structures can be evolved.

5.1. IFS Theory

An IFS $\Omega = \{F, (w_n)_{n=1,\ldots,N}\}$ is a collection of N functions defined on a complete metric space (F, d). Let W be the Hutchinson operator, defined on the space of subsets of F:

$$\forall\, K \subset F,\quad W(K) = \bigcup_{n\in[0,N]} w_n(K)$$

If all w_n functions are contractive (i.e. there exists a positive real number $s < 1$ such that $d(w(x), w(y)) \leq s.d(x, y)$ for all $(x, y) \in F^2$), the IFS is called *hyperbolic*, and there exists a unique set A, called the *attractor* of the IFS, such that $W(A) = A$.

The uniqueness of the attractor is a result of the contractive mapping fixed-point theorem for W, which is contractive according to the Hausdorff distance defined by

$$d_H(A, B) = \max\Big[\max_{x\in A}\Big(\min_{y\in B} d(x, y)\Big), \max_{y\in B}\Big(\min_{x\in A} d(x, y)\Big)\Big]$$

From a computational viewpoint, there are two known ways to compute the attractor of an IFS:

- *Stochastic Method* (*toss-coin*): Let x_0 be the fixed point of one of the w_i functions. Build the sequence x_n by $x_{n+1} = w_i(x_n)$, i being randomly chosen in $\{1..N\}$. Then $\bigcup_n x_n$ is an approximation of the attractor of Ω (the larger n, the more precise the approximation).
- *Deterministic Method*: From any kernel S_0, build the sequence $\{S_n\}$ of subsets by $S_{n+1} = W(S_n)$. When n goes to ∞, S_n is an approximation of the real attractor of Ω.

5.2. *Evolutionary IFS Identification*

The first attempts to evolve IFS using EAs dealt with the inverse problem: given a target shape $A \subset F$, find the IFS whose attractor is A.

This problem can be formulated as an optimization problem: find the IFS whose attractor minimizes the distance to the target shape A. As the function to be optimized is extremely complex, some *a priori* restrictive hypotheses are necessary. Usually, the search space is that of affine IFS, with a fixed number of functions: see [43, 44] for early computational methods. More recently, solutions based on Evolutionary Algorithms have been presented for affine IFS, i.e. IFS in which all functions are affine functions [45–47].

But affine IFS are a small subset of possible IFS, and some previous work of one of the authors [48] dealt with general non-affine IFS (called *Mixed IFS*) using GP, that allows to evolve any type of function. However, whereas assessing the contractivity of affine functions is straightforward, the contractivity of general functions defined as GP trees could only be numerically checked a posteriori—and at a heavy computational cost. This drawback motivated the very recent introduction of Polar IFS [49] in which the functions are sought (still using GP) in polar form around their fixed points: a simple condition on the ρ functions ensures the local contractivity of the function around its fixed point. While this does not ensure the global contractivity, the proportion of contractive functions among that class of polar functions is much larger than that of contractive general GP trees—and the inverse problem can be solved more rapidly and accurately.

Unfortunately, when the present work started, only the GP program to identify mixed IFS was operational. Hence the first results presented in next sections using IFS representation for the TOD problem have been obtained using the mixed IFS GP-based program described in detail in [48].

5.3. *IFS Representation for TOD: First Results*

The idea of shape representation using IFS is straightforward: The attractor of an IFS is a shape defined in the design domain. Hence the fitness of the IFS can be computed using that shape as a structure, potential solution of the TOD.

The attractor of a given IFS is computed on the mesh that is used for the FE analyses, and the fitness is computed as stated in Section 2.3. The same 1×2 and 2×1 benchmark cantilever problems than in Section 4 are used, and Fig. 16 shows the best results obtained in 10 runs.

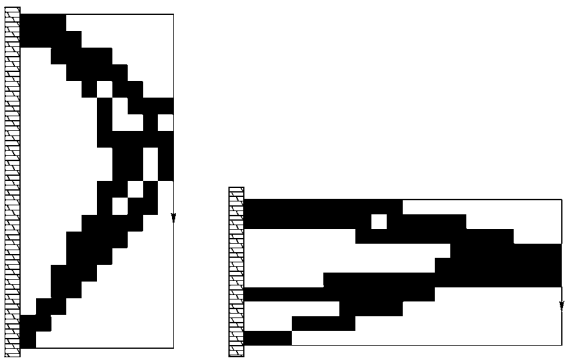


Figure 16. The two benchmark results for the IFS representation. (a) weight = 0.31 (b) weight = 0.43.

First, the good news is that reasonable structures were obtained. Moreover, their shapes are indeed more "lace-like" than when using a Voronoi-based representation—and without the cost of describing all small holes as in the bitarray representations.

However, the results are not as good as the results obtained by Voronoi representations, and that appealed for further experiments on less simple problems: the IFS representation was used for the 10×1 cantilever problem of Section 4.3. The best feasible result (out of 10 independent runs) can be seen on Fig. 17. Comparing this result to those presented on Figs. 12 and 13, it is clear that the optimal structure is heavier here. Note that a lighter structure (0.55) that violates the constraint on the maximal displacement very slightly (12.17 vs 12) has also be obtained in one of the runs.

These preliminary experiments raise some critical issues:

- The variance of the results was very high—some results were really not good at all. Of course, such high variance is a real problem only in exploitation situations, where it is mandatory that reasonable solutions are found at every run. Moreover, for design problems, it can even become an advantage, as one can hope to obtain *very* good solutions, even if rather rarely. But it did not happen with the IFS representation on the benchmarks problems presented here, maybe because the (known) best solutions are still too simple.
- The same adaptive penalty strategy was used here than for the Voronoi-based representations (see

Figure 17. Optimal structure on the 100×10 mesh for 10×1 cantilever plate for the *IFS* representation. $D_{lim} = 12$. weight $= 0.58$. CPU time $=15$ s/gen.

Section 2.3). However, whereas all runs of Voronoi-based representations found feasible solutions, most runs using the IFS representation found slightly infeasible solutions.

- the computational time for decoding is much larger for the IFS representation than for the Voronoi representation.
- The influence of mesh refinement on the actual shape obtained by decoding an IFS is not easy to guess. However, first experiments suggest that different meshes might result in quite different shapes up to very fine meshes.

These remarks suggest that too many things remain unknown both about the structure of the IFS search space and about the actual morphogenetic process. Further experiments are needed, on different problems where the solution is not clearly a simple object e.g. experimenting with design problems outside Structural Mechanical, e.g. in Image Analysis domain. Another issue is the extension to three-dimensional problems. Whereas the theoretical extension is straightforward, the complexity of the computation of the attractor of a 3D IFS will increase drastically.

6. Discussion and Conclusion

This paper has introduced new representations for the representation of objects in the framework of Evolutionary Computation. These representations were experimented with on some Design problems in Structural Mechanics. Departing from the raw bitarray representation based on a fixed discretization of the design domain, representations based on the theory of Voronoi diagrams have been proposed, from the simple Voronoi representation to the more complex dipole and Voronoi-bar representations.

These representations are unstructured, i.e. an individual is a variable-length unordered list of alleles. They are compact, in the sense that they don't require en extensive description of the object at hand on a (usually very large) fixed set of alleles: Though the structure of a single allele increases when going from the Voronoi representation to the Voronoi-bar representation, all three representation implement self-adaptive complexity of the solutions, i.e. the actual complexity of the individuals in terms of number of alleles is evolved by the algorithm and does not have to be predefined by the user.

These representations have been tried on simple test problems of Topological Optimum Design. The results suggest that all three representations can solve such problems, and require roughly the same computational effort for the same quality of solution, with a slight advantage for the Voronoi-bar representation. However, when examining the complexity of the solution, there is a clear advantage in using the Voronoi-bar representation, whose solutions consistently involve less alleles than both others. Note that this probably also explains the observed slight improvement in quality versus computation effort, as it is easier to fine tune the solution when only few alleles are to be adjusted. However, it should be kept in mind that all 2D cantilever problems have truss-like optimal solutions constructed from ... bar-like elements. Further experiments on problems for which the optimal solutions does not exhibit such characteristics should be carried on.

Finally, the IFS representation was presented, a morphogenetic representation in which the structure is indirectly defined as the attractor of a set of contractive mappings on the design domain. Such representation does not make any a priori supposition on the shape of building blocks for the solution of the problem at hand. This should allow more complex solutions to be evolved without designing specific alleles.

Reasonable results were obtained on simple 2D TOD problems, but slightly worse than those of any of the Voronoi-based representations. A possible reason for such results is that the increase of complexity of the morphogenetic process might only prove beneficial for problems where the solution is also complex—and further work will try to apply this representation to more difficult problems, in order to assess for that hypothesis. Moreover, it also might be the case that the lack of causality (direct feed-back from the mechanical

structure on the IFS) forbids any useful evolutionary process, at least with so few individuals and generations. Some experiments on highly parallel systems with distributed populations of hundreds of thousands of individuals might help answering that question.

Another critical issue is the dependency of the morphogenetic process on the mesh, that seems to be much higher for IFS representations than for all Voronoi representations. Two possible answers will be investigated: by using different unstructured meshes during evolution, or by making the decoding process smoother. First, by changing the mesh at every generation, or by averaging the fitness over a few meshes, it is hoped that only solutions that are robust with respect to the mesh will survive successive selections. Second, the numerical computation of the attractor of an IFS fills an element with material as soon as it is hit once by the toss-coin algorithm, whereas smoother decoding would be to consider only the hard core of the attractor requiring a minimal number of such hits before filling the corresponding element.

In the present state of this research, however, the Voronoi-bar representation seems a good choice when looking for representations of objects suitable for evolutionary processes, as witnessed by their use in this paper for evolutionary TOD: it achieves a good compromise between compactness of the solutions and efficiency of search for good solutions. Unfortunately, whereas the extension of the Voronoi and dipole representations to three dimensions is straightforward (see Section 4.4), that of the Voronoi-bar representation requires some more work: one will probably need plates and bars with different cross-section shapes to be included in the elementary alleles. In that perspective, the IFS representation will also be tested on 3D complex problems.

It is clear that compact expressive representations are a prerequisite to successful evolutionary-guided creativity [4]. In that respect, the representations of objects proposed in that paper are a step toward more efficient evolutionary design. However, as quoted in [1, 50], a key feature for creative design is the use of modularity, i.e. the ability to evolve sub-structures and to use them as new building blocks. None of the proposed representations does include high level constructs, such as the possibility to evolve symmetric, or re-usable sub-solutions. For instance, any mechanical engineer would design solutions of the $N \times 1$ cantilever problem for large N by using many almost-identical small truss-structures again and again.

In that direction, some hierarchical representations for shapes have been already proposed in the literature, such as the Quad-tree representation [51]. However quad-tree representation is not easy to evolve, as for instance standard tree crossover does not preserve the locality of quad-tree discretization. Another possible approach could be to hybridize the Voronoi representation with some IFS-like sub-representation: an IFS would be attached to each Voronoi site, and be used to define the shape of the object in the corresponding Voronoi cell (in a similar way that the angle and the width of a bar define the shape of the structure in the bar-representation of Section 3.3. The global evolution of such a representation might prove too time-consuming, but could be replaced by some two-steps evolution, in the line of [52]: first, identify the IFS adapted to the problem at hand; then use them as a (fixed) library where Voronoi sites would be allowed to pick up their internal shape.

Anyway, some coupling between a hierarchical approach to complex representations, and one of the unstructured representations presented here seems to be a possible route to the Grail of Evolutionary Design, the automatic design of highly complex structures. It is hoped that the work in this paper actually brings some building blocks to such higher level morphogenetic representation—while already allowing the direct computation of solutions to simple problems out of reach for "standard" extensive representations.

Acknowledgments

The authors wish to thank the anonymous reviewers, whose critical suggestions helped to improve both the readability and the discussion of the paper.

Note

1. Of course, there also exists *extensive unstructured* representation, such as the Messy GA representation [23], different representations for the TSP [54], or extensive description of variable-topology neural networks [55,56] which will not be considered further in this paper in the light of the scalability issue.

References

1. J.R. Koza, *Genetic Programming II: Automatic Discovery of Reusable Programs*, MIT Press: Massachussetts, 1994.
2. J.R. Koza et al., *Genetic Programming III: Automatic Synthesis of Analog Circuits*, MIT Press: Massachussetts, 1999.

3. F.H. Bennett III, J.R. Koza, M.A. Keane, and D. Andre, "Genetic programming: Biologically inspired computation that exhibits creativity in solving non-trivial problems," in *AISB'99 Symposium on Scientific Creativity*, The Society for the Study of Artificial Intelligence and Simulation of Behaviour, 1999, pp. 29–38.
4. P.J. Bentley, (Ed) *Evolutionary Design by Computers*, Morgan Kaufman: San Mateo, CA 1999.
5. P.J. Bentley and S. Kumar, "Three ways to grow designs: A comparison of embryogenies for an evolutionary design problem," in *Proceedings of the Genetic and Evolutionary Conference*, vol. 1. Morgan Kaufmann: San Mateo, CA 1999.
6. P.D. Surry and N.J. Radcliffe, "Formal algorithms + formal representations = search strategies," in *Proceedings of the 4th Conference on Parallel Problems Solving from Nature*, 1996, edited by H.-M. Voigt, W. Ebeling, I. Rechenberg, and H.-P. Schwefel, volume 1141 of LNCS, Springer Verlag: Berlin, pp. 366–375.
7. P. Bentley, "Exploring component-based representations," in *Adaptive Computing in Design and Manufacture—ACDM'2000*, edited by I. Parmee, pp. 161–172, 2000.
8. J.R. Koza, *Genetic Programming: On the Programming of Computers by means of Natural Evolution*, MIT Press: Massachusetts, 1992.
9. W. Banzhaf, P. Nordin, R.E. Keller, and F.D. Francone, *Genetic Programming—An Introduction On the Automatic Evolution of Computer Programs and Its Applications*, Morgan Kaufmann: San Mateo, CA 1998.
10. F. Gruau, "Neural network synthesis using cellular encoding and the genetic algorithm," PhD thesis, Ecole Normale Superieure de Lyon, 1994.
11. J.R. Koza, M.A. Keane, J. Yu, F.H. Bennett III, and W. Mydlowec, "Evolution of a controller with a free variable using genetic programming," in *EuroGP'2000*, edited by R. Poli et al., volume 1802 of LNCS, Springer Verlag, pp. 91–105, 2000.
12. C. Kane and M. Schoenauer, "Topological optimum design using genetic algorithms." *Control and Cybernetics*, vol. 25, no. 5, pp. 1059–1088, 1996.
13. C. Kane, "*Algorithmes génétiques et Optimisation topologique*," PhD thesis, Université de Paris VI, July 1996.
14. C. Kane and M. Schoenauer, "Optimisation topologique de formes par algorithmes génétiques," *Revue Française de Mécanique*, vol. 4, pp. 237–246, 1997.
15. P.G. Ciarlet, *Mathematical Elasticity, Vol I: Three-Dimensional Elasticity*, North-Holland: Amsterdam, 1978.
16. J. Cea, "Problems of shape optimum design," in *Optimization of Distributed Parameter Structures—Vol. II*, edited by E.J. Haug and J. Cea, vol. 50, NATO Series, Series E, 1981, pp. 1005–1088.
17. S. Garreau, Ph. Guillaume, and M. Masmoudi, "The topological sensitivity for linear isotropic elasticity," in *Proceedings of European Conference on Computational Mechanics*, 1999.
18. M. Bendsoe and N. Kikuchi, "Generating optimal topologies in structural design using a homogenization method," *Computer Methods in Applied Mechanics and Engineering*, vol. 71, pp. 197–224, 1988.
19. 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*, vol. 12, no. 6 pp. 839–878, 1993.
20. G. Allaire, E. Bonnetier, G. Francfort, and F. Jouve, "Shape optimization by the homogenization method," *Nümerische Mathematik*, vol. 76, pp. 27–68, 1997.
21. G. Allaire, Z. Belhachmi, and F. Jouve, "The homogenization method for topology and shape optimization. single and multiple loads case." *European Journal of Finite Elements*, vol. 15, no. 5/6, pp. 649–672, 1996.
22. 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.
23. M. Schoenauer and Z. Wu, "Discrete optimal design of structures by genetic algorithms," in *Conférence Nationale sur le Calcul de Structures*, edited by Bernadou et al., Hermes, pp. 833–842, 1993.
24. D. Grierson and W. Pak, "Discrete optimal design using a genetic algorithm," in *Topology Design of Structures*, edited by M. Bendsoe and C. Soares, NATO Series, pp. 117–133, 1993.
25. O. Bohnenberger, J. Hesser, and R. Männer, "Automatic design of truss structures using evolutionary algorithms," in *Proceedings of the Second IEEE International Conference on Evolutionary Computation*, 1995, edited by D.B. Fogel, vol. 1, IEEE Press, pp. 143–149.
26. R. Leriche and R.T. Haftka, "Optimization of laminate stacking sequence for buckling load maximization by genetic algorithms," *AIAA Journal*, vol. 31, no. 5, pp. 951–970, 1993.
27. G. Anagnostou, E. Ronquist, and A. Patera, "A computational procedure for part design," *Computer Methods in Applied Mechanics and Engineering*, vol. 97, pp. 33–48, 1992.
28. E. Jensen, "*Topological structural design using genetic algorithms*," PhD thesis, Purdue University, Nov. 1992.
29. C.D. Chapman, K. Saitou, and M.J. Jakiela, "Genetic algorithms as an approach to configuration and topology design," *Journal of Mechanical Design*, vol. 116, pp. 1005–1012, 1994.
30. F. Jouve, *Modélisation mathématique de l'œil en élasticité nonlinéaire*, vol. RMA 26. Masson Paris, 1993.
31. Z. Michalewicz and M. Schoenauer, "Evolutionary algorithms for constrained parameter optimization problems," *Evolutionary Computation*, vol. 4, no. 1, pp. 1–32, 1996.
32. S. Ben Hamida and M. Schoenauer, "An adaptive algorithm for constrained optimization problems," in *Proceedings of the 6th Conference on Parallel Problems Solving from Nature*, 2000, edited by M. Schoenauer et al., LNCS, Springer Verlag, pp. 529–539.
33. H. Hamda and M. Schoenauer, "Adaptive techniques for evolutionary topological optimum design," in *Evolutionary Design and Manufacture*, edited by I. Parmee, 2000, pp. 123–136.
34. C. Kane and M. Schoenauer, "Genetic operators for two-dimensional shape optimization," in *Artificial Evolution*, edited by J.-M. Alliot, E. Lutton, E. Ronald, M. Schoenauer, and D. Snyers, volume 1063 of LNCS, Springer Verlag: Berlin, pp. 355–369, 1995.
35. R. Cerf, "An asymptotic theory of genetic algorithms," in *Artificial Evolution*, edited by J.-M. Alliot, E. Lutton, E. Ronald, M. Schoenauer, and D. Snyers, vol. 1063 of LNCS, Springer Verlag: Berlin, pp. 37–53, 1996.
36. D.E. Goldberg, K. Deb, and J.H. Clark, "Genetic algorithms, noise and the sizing of populations," *Complex Systems*, vol. 6, pp. 333–362, 1992.
37. M. Schoenauer, "Representations for evolutionary optimization and identification in structural mechanics," in *Genetic*

Algorithms in Engineering and Computer Sciences, edited by J. Périaux and G. Winter, John Wiley: New York, pp. 443–464. 1995.
38. M. Schoenauer, L. Kallel, and F. Jouve, "Mechanical inclusions identification by evolutionary computation," *European Journal of Finite Elements*, vol. 5, no. 5/6, pp. 619–648, 1996.
39. M. Schoenauer, A. Ehinger, and B. Braunschweig, "Non-parametric identification of geological models," in *Proceedings of the Fifth IEEE International Conference on Evolutionary Computation*, 1998, edited by D.B. Fogel, IEEE Press: Las Alamitos, CA.
40. F.P. Preparata and M.I. Shamos, *Computational Geometry: An Introduction*, Springer Verlag: Berlin, 1985.
41. H.-P. Schwefel, *Numerical Optimization of Computer Models*, 2nd edn., John Wiley & Sons: New York, 1981.
42. L. Kallel and M. Schoenauer, "Alternative random initialization in genetic algorithms," in *Proceedings of the 7th International Conference on Genetic Algorithms*, 1997, edited by Th. Bäck, Morgan Kaufmann, pp. 268–275.
43. M.F. Barnsley, *Fractals Everywhere*, Academic Press: New York: 1988.
44. J. Lévy Véhel, "Analyse et synthèse d'objets bi-dimensionnels par des méthodes stochastiques," PhD thesis, Université de Paris Sud, 1988.
45. R. Vrscay, "Moment and collage methods for the inverse problem of fractal construction with iterated function systems, in *Fractal 90 Conference*, 1990.
46. B. Goertzel, "Fractal image compression with the genetic algorithm," *Complexity International*, vol. 1, 1994.
47. D.J. Nettleton and R. Garigliano, "Evolutionary algorithms and a fractal inverse problem," *Biosystems*, vol. 33, pp. 221–231, 1994.
48. E. Lutton, J. Lévy Véhel, G. Cretin, P. Glevarec, and C. Roll, "Mixed IFS: Resolution of the inverse problem using genetic programming," *Complex Systems*, vol. 9, pp. 375–398, 1995.
49. P. Collet, E. Lutton, F. Raynal, and M. Schoenauer, "Polar IFS + individual GP = Efficient Inverse IFS problem solving," *Genetic Programming and Evolvable Machines*, vol. 1, no. 4, pp. 339–361, 2000.
50. J.R. Koza, "Human-competitive machine intelligence by means of genetic algorithms," in *Festschrift in Honor of John H. Holland*, edited by L. Booker, S. Forrest, M. Mitchell, and R. Riolo, Center for the Study of Complex Systems: Ann Arbor, MI, pp. 15–22, 1999.
51. Ch.R. Dyer, A. Rosenfeld, and H. Samet, "Region representation: Boundary codes from quadtrees," *Commun. Ass. Comput. Mach.*, vol. 23, no. 3, pp. 171–178, 1980.
52. T. Schnier and J.S. Gero, "From Mondrian to Frank Lloyd Wright: Transforming evolving representations," in *Adaptive Computing in Design and Manufacture*, edited by I. Parmee, Springer Verlag: Berlin, pp. 207–220. 1998.
53. D.E. Goldberg, B. Korb, and K. Deb, "Messy genetic algorithms: Motivations, analysis and first results," *Complex Systems*, vol. 3, pp. 493–530, 1989.
54. N.J. Radcliffe and P.D. Surry, "Fitness variance of formae and performance prediction," in *Foundations of Genetic Algorithms 3*, edited by L.D. Whitley and M.D. Vose, Morgan Kaufmann: San Mateo, CA, pp. 51–72. 1995.
55. P.J. Angeline, G.M. Saunders, and J.B. Pollack, "An evolutionary algorithm that constructs recurrent neural networks," *IEEE Transactions on Neural Networks*, vol. 5, no. 2, pp. 86–91, 1993.
56. A. Fadda and M. Schoenauer, "Evolutionary chromatographic law identification by recurrent neural nets," in *Proceedings of the 4th Annual Conference on Evolutionary Programming*, edited by J.R. McDonnell, R.G. Reynolds, and D.B. Fogel, March 1995, MIT Press, pp. 219–235.