Journal of Convex Analysis
Volume 9 (2002), No. 2, 503–517

Topological Optimum Design with Evolutionary Algorithms

Hatem Hamda
*CMAP - UMR CNRS 7641 - École Polytechnique,
91128 Palaiseau Cedex, France
hatem@cmapx.polytechnique.fr*

Marc Schoenauer
*CMAP - UMR CNRS 7641 - École Polytechnique,
91128 Palaiseau Cedex, France*

Received December 5, 2000
Revised manuscript received December 26, 2001

This paper addresses a constrained optimization problem in the context of Topological Optimum Design (TOD): the aim is to 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 requirement (e.g. a bound on the maximal displacement under a prescribed loading). We restrict to stochastic optimization methods such as Evolutionary Algorithms (EAs): they do not require any a priori assumption about the function to optimize (or about the constraints) and they are able to tackle optimization problems on different kinds of search spaces. The most crucial step when constructing an EA is the choice of representation, which determines the search space. In order to overcome limitation of previous works, a new representation is presented, termed Voronoi representation, which is independent of any priori discretization. Moreover, constraints are accounted for through penalty function, and a new adaptive penalty method is proposed to explore the neighborhood of the boundary of the feasible region. The results of TOD of standard benchmark 2-D cantilever problems are improved. Further, this approach allows to address 3-D problems, on which it demonstrates its ability to find multiple quasi-optimal solutions.

1. Introduction

Evolutionary Algorithms are well-known stochastic methods of global optimization based on the principles of natural biological evolution ([16], [5], [46]). They do not require any a priori assumptions about the objective function (or constraints) i.e continuity, differentiability, etc. These methods have been widely applied to constrained optimization problems in different real-world applications ([13, 34, 38]).

The problem at hand consists in finding an optimal shape of a structure in a given design domain, i.e a partition of material in the design domain that minimizes the weight of the structure subject to suitable constraints on mechanical response expressed in terms of displacement, for given loading cases. It can be formulated as the following constrained optimization problem:

(P) *minimize* Weight

subject to $D^i_{Max} \leq D^i_{lim}$

where D^i_{Max} is the maximal displacement of the structure under loading i, and D^i_{lim} its prescribed limit.

One of the principal difficulties when solving this problem using Evolutionary Algorithms is the choice of representation. The straightforward representation for shapes that has

ISSN 0944-6532 / $ 2.50

been used in [23, 11, 29] is a bit-array representation based on a mesh of the design domain. However, the complexity of this representation depends on the mesh: it poorly scales up when refining the mesh, or when considering 3-D shapes. In order to overcome this limitation, an other representation (termed Voronoi representation) which is independent of any a priori discretization, is proposed.

Recently, several constraint-handling methods used within Evolutionary Algorithms have been designed ([36], [45], [35]). This paper is based on the techniques of penalty function because it is straightforward to implement. The main difficulty to use this method is the adjustment of the penalty coefficient: this paper proposes a new adaptive penalty approach that allows to explore the neighborhood of the boundary of the feasible region.

The paper is organized as follows. Section 2 presents the mechanical background and briefly reviews some previous works, discussing their limitations. Section 3 presents a brief overview of Evolutionary Algorithms. In Section 4 we introduce the Voronoi representation together with its variation operators. Section 5 describes the fitness function and introduces the new adaptive penalty approach. In Section 6, experimental results on difficult cantilever benchmark problems are presented: the proposed algorithm finds good quality solutions for the 2D 10×1 cantilever, and is able to propose alternative original solutions to a 3D problem.

2. Background

2.1. The mechanical problem

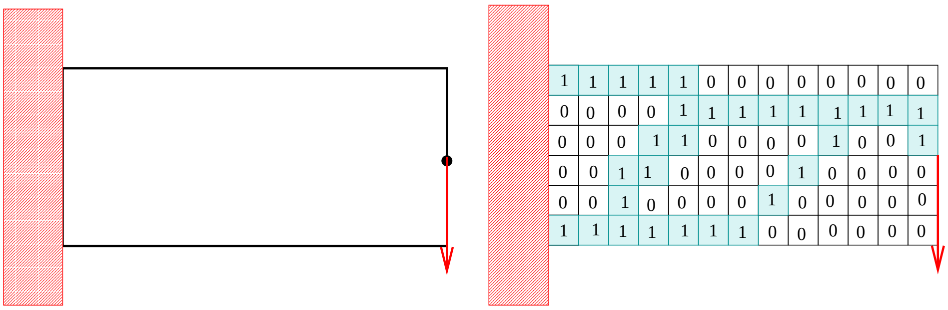
The general framework of this paper is the problem of finding the optimal shape of a structure (i.e. a repartition of material in a given *design domain*) such that the mechanical behavior of that structure meets some requirements (e.g. a bound on the maximal displacement under a prescribed loading).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

The mechanical model used in this paper will be the standard two-dimensional (except in Section 6.3) plane stress linear model, and only linear elastic materials will be considered (see e.g. [12, 25]).

Throughout this paper, the most popular benchmark problem of Optimum Design will be used, that is the optimization of a cantilever plate: the design domain is rectangular, the plate is fixed on the left vertical part of its boundary (the displacement is set to 0), and the loading is made of a single force applied on the middle of its right vertical boundary (see Figure 2.1-a).

2.2. State of the art in Shape Optimization

The main trends in structural optimization can be sketched as follows. The first historical approach is that of *domain variation* [9] (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 has two major defects: first, in some particular cases, it may be unstable even for small 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 [18], allowing the



(a) The design domain (b) The bitarray representation

Figure 2.1: *The* 2×1 *cantilever plate test problem, and a bitarray representation of a structure derived from a regular* 13×6 *mesh.*

modification of the topology of the solution. Nevertheless, this method is strictly limited to the linear elasticity framework.

The other widely used method for topology optimization is the now standard approach of homogenization, introduced in [6], 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 [3] – and the corresponding numerical method does converge to a (non-physical) generalized solution made of fine composite material. It can then be post-processed to obtain an admissible solution with boolean density [2]. The homogenization method is also insofar limited to the linear-elasticity. The theoretical results about optimal micro-structures only handle single-loading cases, though numerical solution to multi-loading cases have been proposed [1]. 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 [33, 47] among others, as well as for topological optimization [19, 8] and for the optimization of composite materials [31].

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 [4]; and Evolutionary Algorithms have been used to solve cantilever problems as the one presented in Section 2.1 in [23, 11, 29].

The above-mentioned limitations of the deterministic methods have been successfully overcome by these works – in [29, 27, 30] 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. The representation issue

One of the most critical decisions made in applying Evolutionary techniques to a particular class of problems (e.g. problem at hand) is the choice of the representation which determines the search space. All the works cited above that address TOD problems with EAs use the same 'natural' binary representation, termed bitarray in [29]: 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. Each element of the mesh is given value 1 if it contains material, 0 otherwise (see Figure 2.1-b).

In spite of its success in solving TOD problem [29, 27, 30], bitarray representation suffers from strong limitations due to the dependency of its complexity on the underlying discretization of the design domain. Indeed, the size of the individual increases with the size of the discretization. However, according to the theoretical results in [10] and the empirical considerations in [51] the required population size for convergence increases linearly with the size of the individual. 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. Two such alternative representations have been proposed [48], and used to solve non destructive control problem in [43]. The rest of the paper will use one of these, namely the Voronoi representation (see Section 4). Before introducing the Voronoi representation and the adapted genetic operators, next section will briefly introduce EAs.

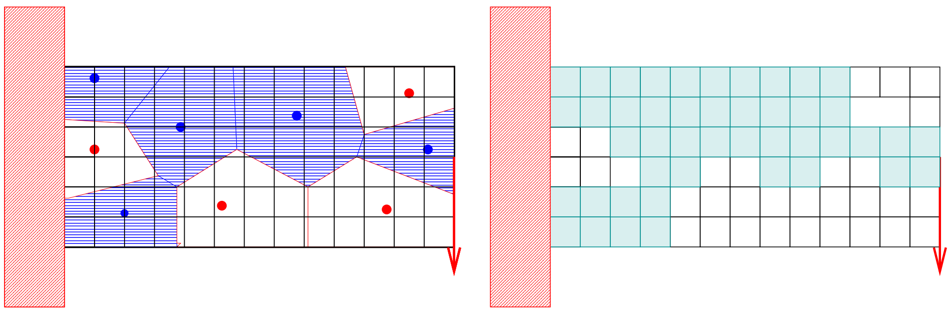


Figure 2.2: *The Voronoi representation on a* 2×1 *cantilever plate test problem, and the corresponding structure (repartition of void/material)*

3. Background on Evolutionary Algorithms

This section is a brief presentation of Evolutionary Algorithms. The reader is referred to ([16], [5], [46]) for more details. Based on the metaphor of natural biological evolution, Evolutionary Algorithms (EAs), of which the most widely known are Genetic Algorithms (GAs), are stochastic optimization methods nowadays allowing to solve a broad range of problems in many domains ([34, 14, 42]).

Compared to traditional optimization methods, such as deterministic methods and enumerative strategies, the evolutionary algorithms are robust, global, zeroth order methods (they use only values of the function to optimize and do not require derivatives or other auxiliary knowledge) and generally rather straightforward to apply.

The process of an Evolutionary Algorithm is illustrated in the following algorithm.

Algorithm

1. An initial population of p individuals (points of search space) are randomly and uniformly initialized on the search space.
2. The fitness (values of the objective function) of all individuals are computed.
3. Iteratively perform the following sub-steps (called generation) on the population until termination criterion has been satisfied.
   - Selection operator: determines which individuals are chosen for mating and how many offspring each selected individual produces. More precisely, the selection process favors those individuals of higher fitness value to reproduce more often than worse individuals. Numerous selection processes can be used, either deterministic or stochastic.
   - Create a new population of individuals by applying the following genetic operators.

     Crossover operator: create new individuals (offsprings) in combining the genetic material contained in the parents (parents-mating population). This operator is applying with a given probability.

     Mutation operator: create a new individual from an existing parental individual by small perturbation, with low probability. Mutation introduces innovation into the population.
   - Evaluate each individual in the population according to the objective function.
   - Choose of witch individuals will be part of next generation (Replacement).

As Evolutionary Algorithms are stochastic search algorithms, the formal specification of stopping criterion is difficult. Usual criteria are stopping the process after a specified number of generations (or number of function evaluations) or when the best fitness value in the population remains unchanged for a given number of generations.

4. Voronoi representation

This section introduces an alternate representation of material/void repartition in a given design domain, more compact than the standard bitarray, and suitable for Evolutionary Algorithms to deal with.

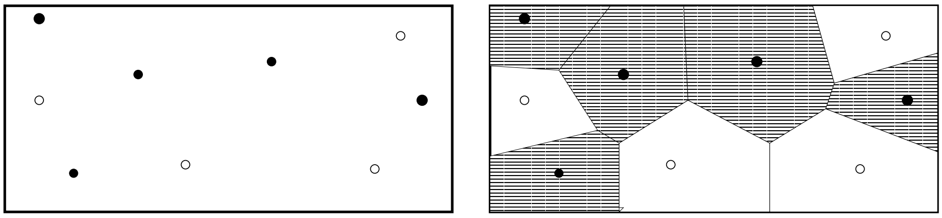
Voronoi diagrams: Consider a finite number of points $V_0, \ldots, V_N$ (the *Voronoi sites*) of a given subset of $\mathbb{R}^n$ (the design domain). To each site V_i is associated the set $Cell(V_i)$ of all points of the design domain for which the closest Voronoi site is V_i, termed *Voronoi cell*:

$$Cell(V_i) = \{M \in D / d(M, V_i) = min_{j=1 \cdots N} d(M, V_j)\}$$

where $d(.,.)$ denotes the Euclidean distance function.

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).

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 Figure 4.1).



(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 4.1: *Voronoi representation on a* 2×1 *design domain.*

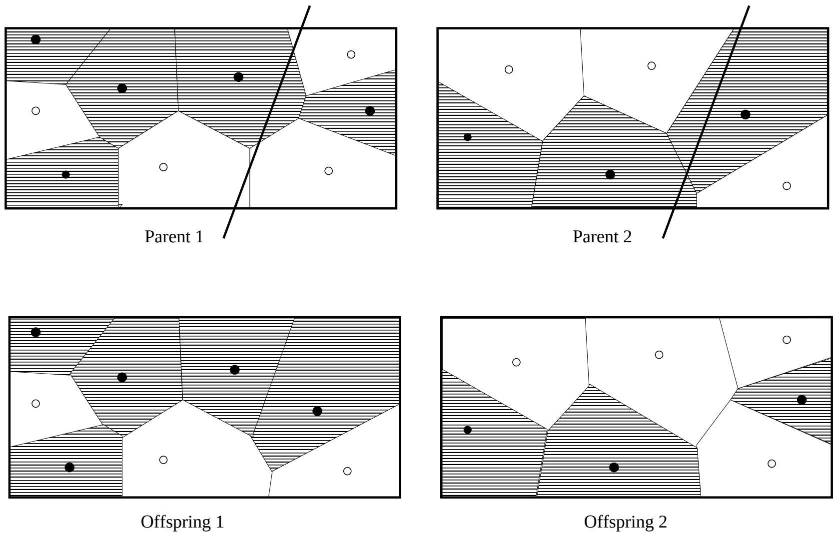


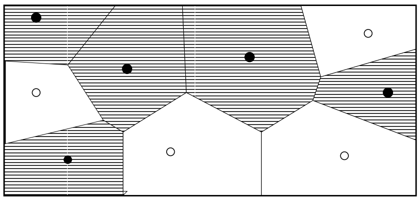
Figure 4.2: *The crossover operator: a random line is drawn across both diagrams, and the sites on either side are exchanged.*

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 gravity center of that element lies (see Figure 2.2).

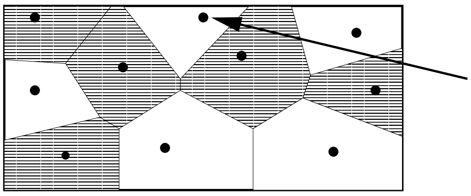
However, the complexity of the individuals (i.e. the number of Voronoi sites in their representation) is totally independent of the choice of the mesh used for fitness computation, and will evolve according to the Darwinian principles underwinning the whole evolutionary process.

Initialization: the initialization procedure for the Voronoi representation is a uniform choice of the number of Voronoi sites up to a user-supplied maximum number, a uniform choice of the Voronoi sites in the structure, and a uniform choice of the boolean void/material label.

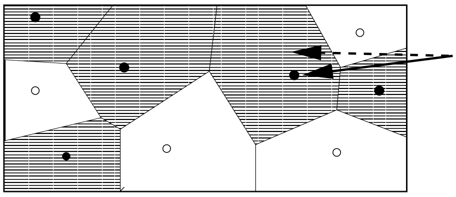
Variation operators: The variation operators for the Voronoi representation are problem-driven:



(a) The parent



(b) The *add* mutation: the site at end of the arrow has been added to the genotype of the parent. The phenotype is rather different from the parent



(c) Mutation by site displacement: a small displacement of one site of the parent slightly modified the phenotype.

Figure 4.3: *Two mutations for Voronoi representation.*

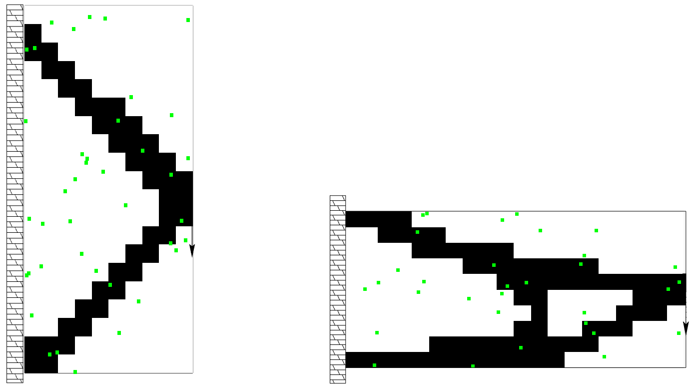
- The **crossover operator** exchanges Voronoi sites on a geometrical basis. In this respect it is similar to the specific bitarray crossover described in [28]. Figure 4.2 is an example of application of this operator.
- The **mutation operator** is chosen, based on user-defined weights, among the following operators (see Figure 4.3):
  - the *displacement mutation* performs a Gaussian mutation on the coordinates of the sites. As in Evolution Strategies [49], 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.

Once one has chosen the representation and defined the corresponding Genetic operators, the last missing ingredient to an Evolutionary Algorithm is the fitness function (Objective function).

5. 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 Figure 2.1-a). The computation of the maximal displacement is made using a Finite Element Analysis solver [25], using the discretized version of the Voronoi diagram (see Figure 2.2).

But first, 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



(a) : weight=21.5%, 35 sites (b) : weight=35%, 32 sites

Figure 4.4: *The two best benchmark results for the Voronoi representation*

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 before the Finite Element Analysis, but slightly penalizes the structure at hand (see [29, 27] 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} denotes its maximal displacement computed by the FEM under loading i, and D^i_{lim} its prescribed limit. The constraint handling method is an adaptive penalty function method, that will now be rapidly described in the context of TOD (see [7] for a detailed description in a general framework).

5.1. Penalty methods

Introducing the positive *penalty parameter* α (in the case of a single constraint), the fitness function to minimize is

$$Weight + \alpha\; max(0, (D_{max} - D_{lim})) \qquad (1)$$

However, adjusting α (that can also be viewed as some sort of Lagrange multiplier for the constraint at hand) is not an easy task [37]. Static penalty, where α is kept constant, can give very good results, but requires a very careful tuning. Dynamic penalty, where α is modified according to a user-defined schedule, as proposed in [24] or in the framework of TOD in [29], requires a lucky guess for the schedule. Here, the use of an adaptive technique really is appealing.

Most Evolutionary Algorithms require a large number of parameters to be tuned by the user – and extensive trials and errors is the usual time-consuming method. So different techniques, grouped under the terminology of *adaptive techniques*, have been proposed to avoid such drawback: the values of a parameter can either be derived from some statistics on past iterations, as the 1/5th rule for mutation step-size adaptation proposed in [41], or even be evolved by the evolution operators themselves, as again the mutation strength in original Evolution Strategies [49]. A few user-defined parameters are still required (e.g. initial values and update schemes) but they can be thought as second order parameters that generally offer a full range of robust values. See [22] for a survey of adaptation in Evolutionary Algorithms.

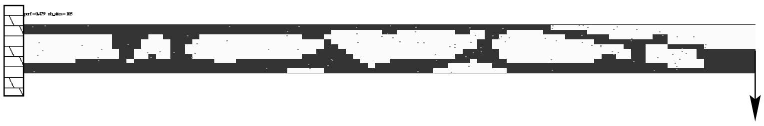


Figure 5.1: *Optimal structure on the* 100×10 *mesh for* 10×1 *cantilever plate for the* **Voronoi** *representation.* $D_{lim} = 12$, *number of cells = 105. weight = 47.9%. CPU time = 14s/gen.*

Adaptive penalty parameters have been successfully used in the context of (discrete) Constraint Satisfaction Problems [15], where the objective is to find at least one feasible individual. In the context of parameter optimization, adaptive schemes has been proposed in [20] and [50]: the penalty parameter is updated according to the feasibility of the best individual in the population in past generations.

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 (the same idea lead to the Segregated GA [32], that used two different fixed penalty parameters to achieve that goal).

Indeed, in many optimization problems, the solution is known or supposed to lie on the boundary of the feasible region. Some specific constraint handling techniques have been proposed to explore only that boundary [44, 45].

However, while it has not been established whether the solution of the constrained problem lies or not on the boundary, common sense suggests that it lies close to that boundary in some sense that will not be precised here. Moreover, that boundary is out of reach – but not the feasible region close to the boundary, thanks to the adaptive penalty method described in next section.

5.2. Population-based adaptive penalty

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 parameter favors the infeasible individuals (and vice-versa), the following update rule for α is proposed to try to keep $\Theta^k_{feasible}$ in $[\Theta_{inf}, \Theta_{sup}]$:

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

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

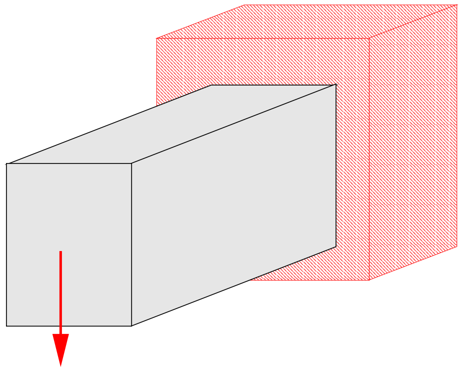


Figure 6.1: *The design domain for the 3-dimensional cantilever problem.*

feasible individuals.

Some comparative results assessing the power of that population-based adaptive penalty method can be found in [7] for test problems, and in [21] in the context of TOD.

6. Experimental results

We introduce in this section some results obtained for the the 2D 10×1 cantilever and 3D problem.

6.1. Experimental parameters

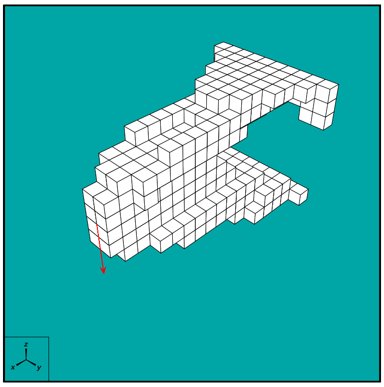
Unless otherwise stated, the experiments presented further on have been performed using the following settings: Standard GA-like evolution (linear rank-based selection and elitist generational replacement of all parents by all offspring) with populations size of 80; At most 40 Voronoi sites per individual; Crossover rate is 0.6 and mutation rate per individual is 0.3; Weights among the different mutations are 1/2 for the displacement mutation and 1/6 for the 3 other mutations; All runs are allowed at most 2000 generations, and the algorithm stops after 300 generations without improvement; every plot is the result of 21 independent runs; All CPU times are given related to a Pentium II processor running at 300MHz under Linux.
Figures 4.4(a) and 4.4(b) show best structures obtained using the Voronoi representation, with respective limits on the maximal displacement of 20 and 220.

6.2. The 10×1 cantilever

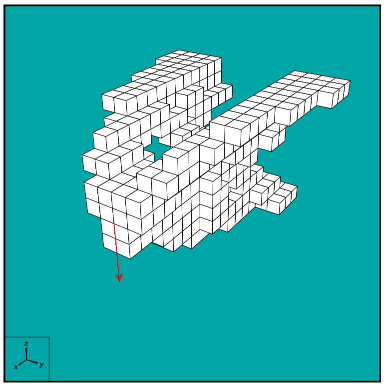
The problem of the 10×1 cantilever (discretized using a 100×10 regular mesh) raises an additional difficulty: most of initial random structures do not connect the fixed boundary and the point where the loading is applied. Hence an alternate initialization procedure was used, where the average weight of random structures can be tuned (see [26] for details). Furthermore, the maximal number of sites for each individual was increased (to 120), and the best results were obtained with a population size of 120. Figure 5.1 shows the most significant result for $D_{lim} = 12$.

6.3. Three-dimensional problem

This section introduces the first results of 3D TOD obtained using Evolutionary Computation (as far as we are aware of).



(a) : weight=0.15178, 103 sites



(b) : weight=0.166, 109 sites

Figure 6.2: *Two results for the symmetrical three-dimensional problem using a* $16 \times 7 \times 10$ *mesh for half of the structure, with same constraints (CPU time = 6mn/gen). The point of view is that of Figure 6, i.e. the structure is held fixed on a vertical wall at the back of the figure (not represented).*

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 of the center of its opposite face (see Figure 6). The problem is symmetrical with respect to a vertical plane perpendicular to the fixed wall. Hence only half of the domain is discretized, by a $16 \times 7 \times 10$ mesh. Its left face is fixed, and the loading is applied on the middle of the right face.

Here again the higher complexity of the problem leads to modify the settings: the population size was increased to 120 and the maximum number of Voronoi sites was increased to 120.

Figure 6.2 demonstrates that the algorithm is 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 number of elements). Moreover, it also stresses the ability of EAs to find multiple quasi-optimal solutions to the same problem, some of them quite original indeed compared to the results of the homogenization method on the same problem [2].

7. Conclusion

A global stochastic optimization method for solving the problem of Topological Optimum Design of Mechanical Structures has been presented, based on Evolutionary Algorithms. Unlike all previous evolutionary approaches, which used a “natural” search space built on a given mesh of the design domain, namely a vector of boolean values, the representation for structures proposed here is that of variable length lists of labeled Voronoi sites. The complexity of such representation is not linked to that of an artificial underlying mesh, but is one of the variables of the problem. On the one hand, this allows one to tackle much more complex problems that the fixed boolean representation, as witnessed by the first example of three-dimensional Structure Optimization performed with Evolutionary Algorithms. But on the other hand, that complexity is also adaptive in space: imagine that the best solutions have a large part made of void, and a small part where void and material are mixed with a rather fine grain. Whereas mesh-based representation would require a fine mesh to be able to account for the fine laced region, Voronoi representation will hopefully be able to use few Voronoi sites, except in that region.

Using Evolutionary Algorithms has also another advantage: Topological Optimum Design is basically a multi-objective problem – minimize both the weight and the displacement of the structure. This problem has been turned into a constrained optimization problem only because multi-objective optimization is more difficult to handle – and is usually handled by some a priori weighting of the different objectives. But there are specific Evolutionary methods to sample the whole Pareto front of multi-objective problems [17] (i.e. get a sample of the best possible compromises, structures such that no other structure has both a smaller weight **and** a smaller displacement. Hence the user can make his choice while actually seeing all possible compromises, instead of being asked a priori for relative importance of the different objectives. On-going work is studying such an approach.

References

[1] G. Allaire, Z. Belhachmi, F. Jouve: The homogenization method for topology and shape optimization. Single and multiple loads case, European Journal of Finite Elements 15(5–6) (1996) 649–672.

[2] G. Allaire, E. Bonnetier, G. Francfort, F. Jouve: Shape optimization by the homogenization method, Numerische Mathematik 76 (1997) 27–68.

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

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

[5] T. Bäck: Evolutionary Algorithms in Theory and Practice, Oxford University Press, New York (1995).

[6] M. Bendsoe, N. Kikuchi: Generating Optimal Topologies in Structural Design Using a Homogenization Method, Computer Methods in Applied Mechanics and Engineering 71 (1988) 197–224.

[7] S. BenHamida, M. Schoenauer: An adaptive algorithm for constrained optimization problems, in: Proceedings of the 6^{th} Conference on Parallel Problems Solving from Nature, M. Schoenauer et al. (eds.), Springer-Verlag, LNCS (2000) 529–539

[8] O. Bohnenberger, J. Hesser, R. Männer: Automatic design of truss structures using evolutionary algorithms, in: Proceedings of the Second IEEE International Conference on Evolutionary Computation 1, D. B. Fogel (ed.), IEEE Press (1995) 143–149.

[9] J. Cea: Problems of shape optimum design, in: Optimization of Distributed Parameter Structures - Vol. II, E. J. Haug, J. Cea (eds.), NATO Series, Series E 50 (1981) 1005–1088.

[10] R. Cerf: An asymptotic theory of genetic algorithms, in: Artificial Evolution, J.-M. Alliot, E. Lutton, E. Ronald, M. Schoenauer, D. Snyers (eds.), Springer-Verlag, LNCS 1063 (1996) 37–53.

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

[12] P. G. Ciarlet: Mathematical Elasticity, Vol I: Three-Dimensional Elasticity, North-Holland, Amsterdam (1978).

[13] D. Dasgupta, Z. Michalewicz (eds.): Evolutionary Computation in Engineering, Springer-Verlag (1997).

[14] D. Delahaye, J.-M. Alliot, M. Schoenauer, J.-L. Farges: Genetic algorithms for automatic regrouping of air traffic control sectors, in: Proceedings of the 4^{th} Annual Conference on Evolutionary Programming, J. R. McDonnell, R. G. Reynolds, D. B. Fogel (eds.), MIT Press, March (1995) 657–672.

[15] A. E. Eiben, Z. Ruttkay: Self-adaptivity for constraint satisfaction: learning penalty functions, in: Proceedings of the Third IEEE International Conference on Evolutionary Computation, T. Fukuda (ed.), IEEE Service Center (1996) 258–261.

[16] D. B. Fogel: Evolutionary Computation. Toward a New Philosophy of Machine Intelligence, IEEE Press, Piscataway, NJ (1995).

[17] C. M. Fonseca, P. J. Fleming: An overview of evolutionary algorithms in multiobjective optimization, Evolutionary Computation 3(1) (1995) 1–16.

[18] S. Garreau, Ph. Guillaume, M. Masmoudi: The topological sensitivity for linear isotropic elasticity, in: Proceedings of European Conference on Computational Mechanics (1999).

[19] D. Grierson, W. Pak: Discrete optimal design using a genetic algorithm, in: Topology Design of Structures, M. Bendsoe, C. Soares (eds.), NATO Series (1993) 117–133.

[20] A. B. Hadj-Alouane, J. C. Bean: A Genetic Algorithm for the Multiple-Choice Integer Program, Department of Industrial and Operations Engineering, The University of Michigan, TR 92-50 (1992).

[21] H. Hamda, M. Schoenauer: Adaptive techniques for evolutionary topological optimum design, in: Evolutionary Design and Manufacture, I. Parmee (ed.) (2000) 123–136.

[22] R. Hinterding, Z. Michalewicz, A. E. Eiben: Adaptation in evolutionary computation: a survey, in: Proceedings of the Fourth IEEE International Conference on Evolutionary Computation, T. Bäck, Z. Michalewicz, X. Yao (eds.), IEEE Press (1997) 65–69.

[23] E. Jensen: Topological Structural Design using Genetic Algorithms, Purdue University, November (1992).

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

[25] F. Jouve: Modélisation mathématique de l'œil en élasticité non-linéaire, RMA 26, Masson Paris (1993).

[26] L. Kallel, M. Schoenauer: Alternative random initialization in genetic algorithms, in: Proceedings of the 7^{th} International Conference on Genetic Algorithms, Th. Bäck (ed.), Morgan Kaufmann (1997) 268–275.

[27] C. Kane: Algorithmes génétiques et optimisation topologique, Université de Paris VI, July (1996)

[28] C. Kane, M. Schoenauer: Genetic operators for two-dimensional shape optimization, in: Artificial Evolution, J.-M. Alliot, E. Lutton, E. Ronald, M. Schoenauer, D. Snyers (eds.), Springer-Verlag, LNCS 1063 (1995) 355–369.

[29] C. Kane, M. Schoenauer: Topological Optimum Design using Genetic Algorithms, Control and Cybernetics 25(5) (1996) 1059–1088.

[30] C. Kane, M. Schoenauer: Optimisation topologique de formes par algorithmes génétiques, Revue Française de Mécanique 4 (1997) 237–246.

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

[32] R. G. Leriche, C. Knopf-Lenoir, R. T. Haftka: A segregated genetic algorithm for constrained structural optimization, in: Proceedings of the 6^{th} International Conference on Genetic Algorithms, L. J. Eshelman (ed.), (1995) 558–565.

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

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

[35] Z. Michalewicz: A survey of constraint handling techniques in evolutionary computation methods, in: Proceedings of the 4^{th} Annual Conference on Evolutionary Programming, J. R. McDonnell, R. G. Reynolds, D. B. Fogel (1995).

[36] Z. Michalewicz, C. Z. Janikow: Handling constraints in genetic algorithms, in: Proceedings of the 4^{th} International Conference on Genetic Algorithms, R. K. Belew, L. B. Booker (eds.), Morgan Kaufmann (1991) 151–157.

[37] Z. Michalewicz, M. Schoenauer: Evolutionary algorithms for constrained parameter optimization problems, Evolutionary Computation 4 (1996) 1–32.

[38] S. Oussedik, D. Delahaye, M. Schoenauer: Flight alternative routes generation by genetic algorithms, in: Proceedings of the 2000 Congress on Evolutionary Computation, IEEE (2000) 896–901.

[40] F. P. Preparata, M. I. Shamos: Computational Geometry: An Introduction, Springer-Verlag (1985).

[41] I. Rechenberg: Evolutionstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution, Fromman-Hozlboog Verlag, Stuttgart (1973).

[42] M. Schoenauer, F. Jouve, L. Kallel: Identification of mechanical inclusions, in: Evolutionary Computation in Engineering, D. Dasgupta, Z. Michalewicz (eds.), Springer-Verlag (1997) 477–494

[43] M. Schoenauer, L. Kallel, F. Jouve: Mechanical inclusions identification by evolutionary computation, European Journal of Finite Elements 5(5–6) (1996) 619–648.

[44] M. Schoenauer, Z. Michalewicz: Evolutionary computation at the edge of feasibility, in: Proceedings of the 4^{th} Conference on Parallel Problems Solving from Nature, H.-M. Voigt, W. Ebeling, I. Rechenberg, H.-P. Schwefel (eds.), Springer-Verlag, LNCS 1141 (1996) 245–254.

[45] M. Schoenauer, Z. Michalewiczkey: Boundary operators for constrained parameter optimization problems, in: Proceedings of the 7^{th} International Conference on Genetic Algorithms, Th. Bäck (ed.), Morgan Kaufmann (1997) 322–329.

[46] M. Schoenauer, Z. Michalewicz: Evolutionary computation, Control and Cybernetics 26(3) (1997) 307–338.

[47] M. Schoenauer, Z. Wu: Discrete optimal design of structures by genetic algorithms, in: Conférence Nationale sur le Calcul de Structures, Bernadou et al. (eds.) Hermes (1993) 833–842.

[48] M. Schoenauer: Representations for evolutionary optimization and identification in structural mechanics, in: Genetic Algorithms in Engineering and Computer Sciences, J. Périaux, G. Winter (eds.), John Wiley (1995) 443–464.

[49] H.-P. Schwefel: Numerical Optimization of Computer Models, John Wiley & Sons, New York (1985), 2^{nd} edition (1995).

[50] A. Smith, D. Tate: Genetic optimization using a penalty function, in: Proceedings of the 5^{th} International Conference on Genetic Algorithms, S. Forrest (ed.), Morgan Kaufmann (1993) 499–503.

[51] D. Thierens, D. E. Goldberg: Mixing in genetic algorithms, in: Proceedings of the 5^{th} International Conference on Genetic Algorithms, S. Forrest (ed.), Morgan Kaufmann (1993) 38–55.