



HAL
open science

Problem-specific representations for heterogeneous materials design

Alain Ratle

► **To cite this version:**

Alain Ratle. Problem-specific representations for heterogeneous materials design. European Conference on Artificial Evolution, Nov 1999, Dunkerque, France. pp.111-122, 10.1007/10721187_8. hal-00116137

HAL Id: hal-00116137

<https://hal.science/hal-00116137v1>

Submitted on 17 Oct 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution - NonCommercial 4.0 International License

Problem-Specific Representations for Heterogeneous Materials Design

Alain Ratle

Département de génie mécanique, Université de Sherbrooke,
Sherbrooke, Québec, J1K 2R1 Canada
(current address: Laboratoire de mécanique des solides,
Ecole Polytechnique, 91128 Palaiseau, France)
`ratle@lms.polytechnique.fr`

Abstract. This paper investigates the use of problem-specific data structures and operators in evolutionary optimization for a specific class of combinatorial design problems. The problem consists of finding the optimal distribution of two or more phases of a sound absorbing material on a three-dimensional network, in order to maximize sound absorption properties. The natural structure of the problem is by the way very far from the linear chains classically used by evolutionary algorithms (EAs). Special operators exploiting the three-dimensional structure are proposed and compared with other operators that are working on a linear chain representation. The formers are potentially useful since the natural neighborhood relationships are lost in a linear representation.

1 Introduction

EAs have always relied on a linear chain encoding of the problem to be solved. The initial idea was that following schema theorem [Hol75], some structures and operators were more likely to give better results than others, regardless of the problem at hand. This was inspired from the fact that nature has evolved complex and efficient structures using a chain-like genetic code [Bäc96]. However, since most real-world problems are very different from nature’s problems, there is no reason to believe that nature’s solutions should give optimal results for artificial problems. Many of the real-world optimization problems do not fit easily into a linear representation, and better results have often been obtained using problem-specific data structures and operators. An example is the traveling salesman problem for which various data structures and operators have been tried out [FM91]. Moreover, recent works have suggested that any “blind” change of representation is futile as long as the new coding is not correlated with the problem at hand [Cul98]. This follows from the No Free Lunch theorem [WM97] which states that no optimization algorithm should be expected to give better results on average than any other, unless it has some correlation with a specific problem.

This paper presents the case of a multiphased material design problem which is naturally formulated into a three-dimensional structure. The objective is to

find out the optimal distribution of two or more types of sound absorbing material elements in order to maximize the absorption properties. The problem is subject to constraints on the desired fraction of each phase. In a first approach, evolutionary operators preserving the constraints are developed. These operators are however designed with no regard to the particular data structure. In a second approach, evolutionary operators based on the natural structure are elaborated, and the optimization results are compared with the former approach.

2 The Physical Problem

Porous materials are often used for various noise control applications [All93]. It have been pointed out by Allard [All93] that the efficiency of these materials might be improved by an heterogeneous stratification. More recently, Atalla et al. [APA96] have suggested that the use of three-dimensionally heterogeneous networks might yields even better performances, but this hypothesis has not been tested out.

The aim of this paper is to give an answer to the question as whether or not an optimal 3-D distribution of multiple phases can give better solutions compared to a simple one-dimensional stratification. The proposed approach is to use problem-specific EAs as an experimental tool to gain knowledge on this question. Recent results by Ratle & Atalla [RA98] have shown that constraint preserving operators greatly ease the search by a restriction of the search space. However, the exploitation of the natural data structure in the design of evolutionary operators has never been addressed.

In the present case, the material is modeled by a finite element mesh having $N_x \times N_y \times N_z$ elements, and made of two or more different materials. The resulting multiphased material is placed at one end of a semi-infinite acoustical wave guide, as shown on Figure 1. The boundaries of the domain are assumed to be rigid and the porous medium is excited by a plane wave. The resolution gives the absorption coefficient at a specific frequency, and the optimization criteria is the average value of this coefficient in some frequency range.

The design efficiency relies on the availability of a suitable optimization method. A major constraint is that only a small number of fitness function evaluations can be allowed, due to computational cost. The chosen approach [RA98] transforms a parametric problem into a combinatorial “ N choose m_1, m_2, \dots, m_M ” problem, where N is the total number of elements and the m_i ’s are the number of elements required for each of the M materials. The coding consists of a sequence of integers where each one is mapped to a type of material. In the binary case, a 0 represents an element of the base material, and a 1, an element of the so-called additive material. Such a representation have often been used for topological design of mechanical shapes (see for example Kane and Schoenauer [KS95]).

Elementary combinatorics shows that the number of possible solutions with N elements and M materials is equal to $N! / \prod_{i=1}^M (m_i!)$. The optimization problem is worked out with respect to two concepts. First, problem-specific ope-

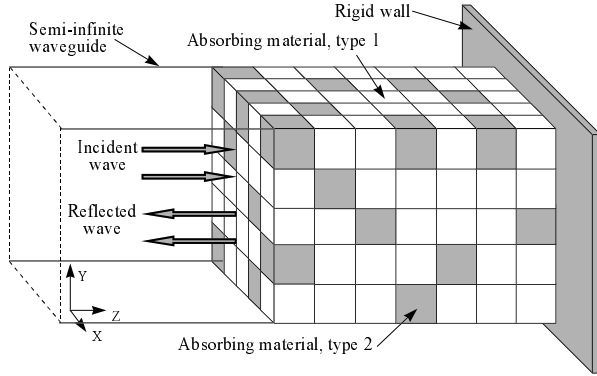


Fig. 1. Heterogeneous porous material made up from a network of homogeneous patches set on a rigid impervious backing and coupled to a semi-infinite waveguide.

rators for implicit constraint preservation are developed. The second concept consists of considering the natural data structure in operators design. This concept should help to answer the two following questions:

1. Is it better to preserve the whole 3-D structure, or only the average distribution along the principal direction?
2. Is it better to use a local search operator which is defined in the problem space, or one that is defined in the representation space, the latter being potentially highly disruptive in the problem space?

3 Evolutionary Operators for Specific Representations

3.1 Initialization Operator

Initializing a genetic code for the problem at hand requires a random string of length N containing exactly m_j characters of each category j , where $j = 1 \dots M$. The proposed method is described as follow:

```

Begin initialization
  Initialize  $M$  counters to zero
  For  $i = 1$  to  $N$ 
    Do
       $j \leftarrow$  integer random value  $\in \{1, \dots, M\}$ 
      If counter[ $j$ ]  $<$   $m_j$  (feasible value found)
        Increment counter[ $j$ ]
        (genetic code[ $i$ ])  $\leftarrow j$ 
      End if
    While feasible value not found
  End for
End initialization
  
```

The initialization operator does not have to be defined with consideration of the data structure, since the initial individuals are randomly distributed in the representation space.

3.2 Crossover Operators

The design of a crossover operator requires a choice on which informations must be preserved from parents to offsprings. This choice may leads to radically different operators for the same problem. For the problem at hand, two approaches are proposed with implicit constraint preservation. The first one preserves the positions where both parents share the same value, and the second one tries to preserves the longest sub-solution common to both parents. A third operator designed for one-dimensional information preservation is finally presented.

Identical Points Preservation Crossover. A first approach consists of preserving all the positions in the coding where both parents have the same material. For the other positions, any value is acceptable, as long as the constraints are respected. This procedure is termed Identical Points Preservation (IPP) crossover. Given two parents \mathbf{a} and \mathbf{b} , the offspring \mathbf{c} is created by the following procedure:

```

Begin IPP crossover
  Initialize  $M$  counters to zero
  For  $i = 1$  to  $N$ 
    If  $(a_i = b_i) : c_i \leftarrow a_i$  and Increment counter[ $a_i$ ]
  End for
  For  $i = 1$  to  $N$ 
    If  $(a_i \neq b_i)$ 
      Case (counter[ $a_i$ ] =  $m_{a_i}$  and counter[ $b_i$ ] <  $m_{b_i}$ ) :  $c_i \leftarrow b_i$ 
      Case (counter[ $a_i$ ] <  $m_{a_i}$  and counter[ $b_i$ ] =  $m_{b_i}$ ) :  $c_i \leftarrow a_i$ 
      Case (counter[ $a_i$ ] <  $m_{a_i}$  and counter[ $b_i$ ] <  $m_{b_i}$ ) : select  $c_i \in \{a_i, b_i\}$ 
      Case (counter[ $a_i$ ] =  $m_{a_i}$  and counter[ $b_i$ ] =  $m_{b_i}$ ) : select  $c_i \notin \{a_i, b_i\}$ 
      Increment counter[ $c_i$ ]
    End if
  End for
End IPP crossover

```

This operator introduces a high-level of randomization whenever few common building blocks are present between the two parents. This randomization vanishes when the parents become very similar. Because the IPP crossover preserves all the stable elementary positions, it is by the way a structure-preserving operators, since large blocks of common elements between both parents will be preserved. This principle, characteristics common to both parents are passed to the children, have been previously stated by Radcliffe [Rad91], and by Surry and Radcliffe [SR96] as the Random Respectful Recombination (RRR).

Longest Common Substring (LCS) Crossover. The second crossover approach is the preservation of an emerging partial ordering common to both parents, that is, a substring of length $\ell \leq N$. The proposed algorithm is described as follow:

1. Find the longest common substring between the two parents;
2. copy this substring into the offspring at the position indicated by the parent with highest fitness;
3. fill in the remaining positions with the variables given by the second parent in their relative order.

Optimal Crossover Strategy. A better approach is the design of an algorithm that selects the optimal operator between IPP and LCS in every situation. Since a common substring between two parents is likely to be significant only if it is longer than a certain threshold, an optimal crossover strategy would be to use LCS crossover if a long enough substring is found, and IPP crossover otherwise.

Layer crossover. The layer crossover ensures the preservation of only the average composition of each layer along the z axis, regardless of the value carried by the individual elements. This crossover is performed as follow, for the binary case:

1. Calculate for the two parents the number of additive elements $N_{add}(k)$ on each layer $k = 0, \dots, N_z - 1$.
2. Perform crossover over the $N_{add}(k)$'s from the two parents using binary selection between parent 1 and parent 2, i.e. choose either one or the other.
3. Repair the offspring in order to preserve the total number of additive elements: while $\sum_k N_{add}(k) > t$, pick a layer k randomly and decrease $N_{add}(k)$ by one, or do otherwise (increase it) if $\sum_k N_{add}(k) < t$.
4. Select randomly $N_{add}(k)$ positions on each layer k of the offspring and turn them to 1, turn the others to 0.

3.3 Mutation Operators

Any mutation operator that consist of a permutation of elements ensures constraint preservation for this problem. Three operators are suggested:

1. The *point* operator, which consists of choosing two random elements (points), and swapping their values. Since both elements may have the same value, a verification is done to avoid redundant solutions.
2. The *displacement* operator consists of choosing a substring between two randomly chosen points, cutting this string and reinserting it elsewhere.
3. The *inversion* operator consists of choosing a substring between two randomly chosen points, and inverting the order of the values in this substring.

The first of these operator works on a one-dimensional structure, and is therefore the most simple move that can be defined. The two other operators act on a two-dimensional representation which might be correlated or not to the problem-space.

Another mutation operator which works directly on the three-dimensional problem space is also proposed. This operator, the block mutation, produces an inversion of a 3-D block in the problem space. Two corners of a block are first selected (two triplets (i, j, k)), and a mirroring direction is chosen among the three possibilities. Elements into the block are then swapped pairwise in the selected direction. This operator does not break down heavily the natural data structure, compared to the chain mutation operator. Once again, null operations are detected in order to avoid useless reevaluations. A null operation arises whenever the block has a unit or null thickness in the mirroring direction.

4 Numerical Results

Computational experiments have been performed on 3 problems of various size. The first two cases address only the one-dimensional heterogeneity with 15 and 40 layers in the z direction. The first case is in a low frequency range, where high absorption values are very unlikely, and the second case in a mid-frequency range. The third case is a 3-D problem with 250 elements, $N_x = 5$, $N_y = 5$ and $N_z = 10$. Two materials are available with 200 elements of base and 50 elements of additive material. This case is also in a mid-frequency range. Since each evaluation requires the solution of a finite element problem, CPU time is a major limitation¹ and a maximum of 1000 evaluations have been allowed in each case. This represents the stopping criteria of the algorithm. The 3 cases are summarized on Table 1. For the cases 1 and 2, the optimal solution is known, and is compared with optimization results. However, for the case 3, the optimal solution is not known. In all the cases, results are averaged over five runs.

4.1 Constraint Preservation Operators

Mutation Operators. Constraint-preserving mutation operators have been evaluated with a simplified EA, with a population of size 2, binary tournament selection, no crossover, an elitist survival of the best solution in the current population, and a probability of mutation $p_m = 1$. Results are presented on Figure 2a to 2c for the 3 problems. In all the cases, the point operator gives

Table 1. Description of the three test cases

Case no	1	2	3
Material no	Number of elements		
1	5	20	200
2	5	20	50
3	5	0	0
Total	15	40	250
Number of solutions	7.57×10^5	1.38×10^{11}	1.35×10^{53}

¹ The resolution for $N = 250$ takes about 1 minute of CPU on a Pentium 400 MHz running under Linux 2.0, and the complexity of finite elements problems is $O(N^3)$.

better results than any other, but the difference is larger for the 250 variables problem. These differences are explained by representation issues. For the 15 and 40 variables cases, there is a perfect correlation between the problem space and the representation space. For the third case, the problem space is uncorrelated with the representation, and local search operators defined in one space are highly disruptive in the other. The point operator is not affected by this problem.

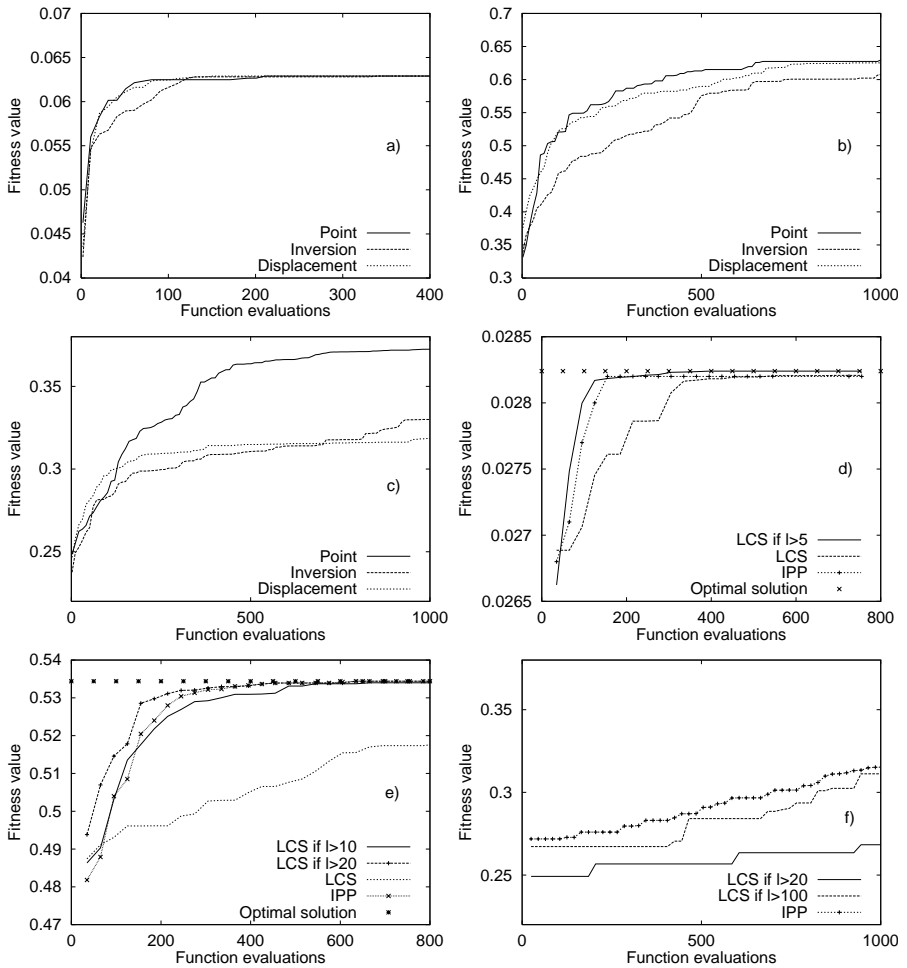


Fig. 2. Comparison of the three mutation operators, for a) 15 variables problem, b) 40 variables, and c) 250 variables. Comparison of the crossover strategies for d) 15 variables problem, e) 40 variables, and f) 250 variables.

Crossover Operators. The two constraint preservation operators have also been compared for the three cases, using an EA with a population size of 25, point mutation with $p_m = 1/2$, probability of crossover $p_c = 1$, binary tournament selection and the elitist survival of the best 5 individuals. Results are presented on Figure 2d for the case 1, and 2e for case 2. These two cases show that the preservation of common positions (IPP) is always better than the preservation of the greatest common substring (LCS) when considered alone. However, a slight improvement is obtained if the LCS operator is employed only if a chain having a certain threshold length is found. For case 3, results presented on Figure 2f shows that the IPP operator is always better. This can also be explained by the disruption problem.

4.2 Integration of Data Structures

This section considers only the 250 variables problem, since the effect of the data structure is present only for 3-D distributions. All the experiments are based on the same EA with a population size of 25, binary tournament selection, elitist survival of the best 5 solutions, a maximum number of 1000 function evaluations, and variable probabilities of crossover and mutation. In the first 4 series of experiments, the various crossover and mutation operators are compared. The next 4 series compare the effect of the rate of application of a same operator. The 8 test cases are described on Table 2. This table also gives the best fitness values obtained on average after 1000 evaluations.

An interesting result is that in all cases except case 7, the use of a crossover operator brings no improvement compared to the use of mutation alone. The best results are even obtained with the most simple operator, the point mutation. This suggests that improvements come mainly from small local perturbations rather than dramatic changes or recombinations. Comparing the two crossovers alone (case 2), it is observed that the preservation of average content of each layer gives better results than the preservation of individual positions

Table 2. Summary of cases settings and results ($p_m = 1$ in all cases except 2).

Case	Mutation	Crossover	p_c	Fitness	Case	Mutation	Crossover	p_c	Fitness
1	Point	No	-	0.349	5	Point	Positions	0.0	0.357
	Chain		-	0.322				0.5	0.347
	Block		-	0.336				1.0	0.316
2	No	Positions	1.0	0.282	6	Point	Layers	0.0	0.357
		Layers	1.0	0.307				0.5	0.328
								1.0	0.278
3	Point	Positions	1.0	0.285	7	Block	Positions	0.0	0.329
		Layers	1.0	0.324				0.5	0.343
								1.0	0.335
4	Block	Positions	1.0	0.325	8	Block	Layers	0.0	0.329
		Layers	1.0	0.286				0.5	0.309
								1.0	0.302

(IPP). This suggests that the IPP crossover alone can hardly introduce new and useful information without the help of a mutation operator. The hypothesis is confirmed by the observation of cases 5 to 8. In these cases, the use of a mutation operator together with the two crossovers gives the advantage to the position-preservation crossover, compared to the layer crossover. This means that there is some interest in preserving the 3-D distribution rather than only a one-dimensional distribution, as long as the evolutionary operators are able to introduce a sufficient amount of diversification.

Another aspect of the problem is the correlation between the various operators in the same algorithm. Case 7 is the only one where an improvement is brought in by the crossover operator. It is also the only one where both structure-preserving operators are employed. This suggests that care should be taken in matching the crossover and mutation operators, the best results are obtained when both are defined with respect to the same kind of information structure.

5 Physical Significance

A comparison of the solutions found by evolutionary optimization for the 250 variables problem with other types of solutions is given on Table 3. The best solution found in all the optimization runs limited to 1000 evaluations, and the average of the 10 best runs are compared with a classically known solution, which consists of stacking the materials in increasing order of fluid flow resistance. This is equivalent to placing all the 50 additive elements in the first two layers. It is observed from these data that the classical solution gives better results than a random distribution of the two phases. However, the solutions found by the EAs give significant improvements. It should be noted that although the limit of 1000 evaluations represents a reasonable computational cost, it does not guarantee optimality. Solutions found after 6000 evaluations show some improvement, but with a computational cost 6 times higher. The standard error decreases dramatically between 1000 and 6000 evaluations, thus indicating that the algorithm tends to reach some limiting value after 6000 evaluations.

The physical distribution of the additive elements is illustrated for two cases. Figure 3 shows a “good” solution with a fitness value of 0.343, and the best solution with a fitness of 0.380. The base material elements are represented on these two figures by the black lozenges, and the additive elements by the circles.

Table 3. Comparison of optimization results and standard solutions.

Description	Average	Standard error
Best solution found after 1000 evaluations	0.3647	-
Average of 10 best solutions after 1000 evaluations	0.3565	0.0048
Classical solution (separated phases)	0.3295	-
Random solution (average of 50)	0.2339	0.0165
Best random solution among 50	0.2710	-
Best solution found after 6000 evaluations	0.3803	-
Average of 10 best solutions after 6000 evaluations	0.3795	0.0007

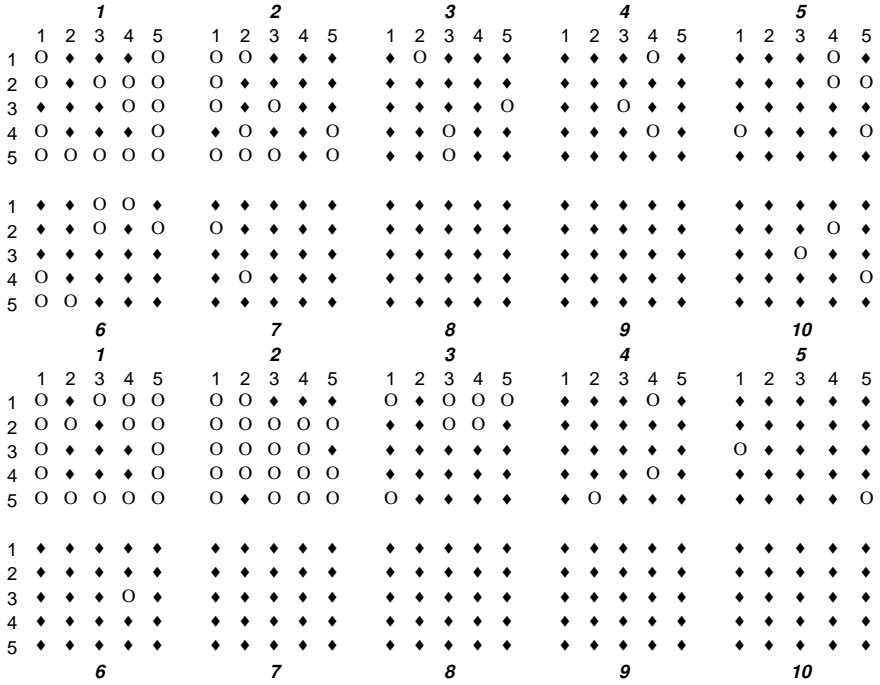


Fig. 3. Distribution of additive elements for a moderately good solution (fitness=0.343) on top, and the best solution found (fitness=0.380) on bottom.

The layer 1 corresponds to the (leftmost) incidence layer, and the layer 10 is backed by the rigid wall. Comparing these cases, it seems that the more the additive elements are pushed to the left, the better is the solution. The best case would be with all the additive elements in the first two layers. However, data on Table 3 show that this is not true. There must exist a non-trivial solution which maximizes the sound absorption coefficient, and which cannot be manually extrapolated from past experience.

6 Discussion and Conclusions

This paper presented an original design problem which can hardly be dealt with using general-purpose algorithms. Even though the case is naturally formulated in a binary, or low-cardinality characters string, classical genetic algorithms are of limited use, due to the constraints and the particular data structure. The comparison of the constraint-preserving mutation operators have shown that in all the cases, an operator working on single elements gives better results than operators working on longer strings. This effect is clearly more pronounced for three-dimensional cases, due to the disruption between problem space and representation space. The same effect has been observed for crossover operators.

In all the cases, but more significantly with a 3-D distribution, preserving the individual positions is better than preserving a substring.

Evolutionary operators have been developed with respect to the specific data structure of the problem. A mutation operator acting on a block of elements defined in the problem space gives better results than one working on a string in the representation space. However, the best results are still obtained using a mutation working on a pair of single elements. In the same way, two crossover operators have been proposed, the first one preserving the individual positions where both parents share the same value, thus preserving their spatial distribution, and the second preserving only the average content of each layer in the main direction. Results have shown that the preservation of the 3-D structure has a positive effect, as long as a sufficient amount of new and relevant information is introduced by a suitable mutation operator. These results also suggests that the correlation between operators employed in the same EA has a great relevance.

The global results show that evolutionary optimization brings significant improvements in the quality of solutions compared to the solution classically known by practitioners and also compared to random solutions, in spite of the fact that the allowed computational resources are not sufficient to ensures optimality. This point enlightens an interesting feature of EAs: even when no guarantee of optimality can be stated, evolutionary optimization often give truly interesting solutions from an engineering point of view, compared to usual practices.

It is also interesting to note that mixing up more than one crossover operator has given in some cases better results than the use of any operator alone. This suggests that a possible improvement would be to use a self-adaptive algorithm, where many different crossover and mutation operators are available, since they all have good and bad points. The probability of use of each one can be coded in the individuals, in a similar way as mutation amplitudes are coded in the self-adaptive evolution strategies [Bäc96], with the updating of the probabilities based on previous performances.

References

- All93. J.F. Allard. *Propagation of sound in Porous Media: Modeling sound absorbing materials*. Chapman & Hall, London, 1993.
- APA96. N. Atalla, R. Panneton, and J.F. Allard. Sound absorption by non homogeneous thin porous layers. *Acta Acoustica*, 83:891–896, 1996.
- Bäc96. T. Bäck. *Evolutionary Algorithms in Theory and Practice*. Oxford University Press, 1996.
- Cul98. J.C. Culberson. On the futility of blind search: an algorithmic view of “no free lunch”. *Evolutionary Computation*, 6(2):109–127, 1998.
- FM91. B.R. Fox and M.B. McMahon. Genetic operators for sequencing problems. In G.J.E. Rawlins, editor, *Foundations of Genetic Algorithms*, pages 284–300, 1991.
- Hol75. J.H. Holland. *Adaptation in Natural and Artificial Systems*. MIT Press, 1975.
- KS95. C. Kane and M. Schoenauer. Genetic operators for two-dimensional shape optimization. In J.-M. Alliot, E. Lutton, E. Ronald, M. Schoenauer, and D. Snyers, editors, *Artificial Evolution*, 1995.

- RA98. A. Ratle and N. Atalla. Evolutionary optimization strategies for the combinatorial design of heterogeneous materials. *J. Evolutionary Optimization*, 1(1):77–88, 1998.
- Rad91. N.J. Radcliffe. Forma analysis and random respectful recombination. In R.K. Belew and L.B. Booker, editors, *Proceedings of the First International Conference on Genetic Algorithms*, pages 222–229, 1991.
- SR96. P.D. Surry and N.J. Radcliffe. Formal algorithms + formal representations = search strategies. In H.-M. Voigt, W. Ebeling, I. Rechenberg, and H.-P. Schwefel, editors, *Parallel Problem Solving from Nature IV*, pages 366–375, 1996.
- WM97. D.H. Wolpert and W.G. MacReady. No-free-lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1(1):67–82, 1997.