Genetic algorithme-based topology optimization: performance improvement through dynamic evolution of the population size

J. Denies[1], B. Dehez[1], F. Glineur[2] and H. Ben Ahmed[3]

[1] CEREM - Universitée catholique de Louvain, place du levant, 3, Louvain-la-Neuve, 1348, Belgium

[2] CORE - Universitée catholique de Louvain, Voie du Roman Pays, 34, Louvain-la-Neuve, 1348, Belgium

[3] SATIE laboratory - Ecole Normale Supérieure de Cachan Antenne de Bretagne, Campus de Ker Lann, 35170, Bruz, France

***Abstract*—Topological optimization tool using genetic algorithm as optimization algorithm are known as very expensive in computation time. In this paper, we study an approach to improve performance of topological optimization tool by introducing a dynamic variation of the population size of children during the process of optimization. This method allows to improve performance of each generation by adapting the number of children created and by introducing a coefficient of reproduction for each individual inside the population of parents. Through this coefficient of reproduction, the number of children assigns to each parent is calculated. The number of evaluations at each generation changes and the tool can saves evaluations in order to increase the number of iterations.**

***Index Terms*—topologyoptimization, design, inverse problem, Voronoï diagram, genetic algorithm**

I. Introduction

More and more often during the design process of a new device, optimization methods tale an importante place in order to help the designer to find the best solution, wether electromechanical or other. These methods differentiate themselves from the design variables on wich they are performed. There are three main categories of differents methods. The first one, called dimensionnal optimization, or the parametric optimization, uses design parameters to size a solution whose geometry has been predifined by the designer. The second one, called shape optimization, changes the boundary between each subdomain of material whose the topology is defined by the designer. And finally, the thirst method, called topology optimization, uses parameters describing the material distribution inside a design space. Unlike the two first methods, the third method does not need an initial solution defined by the designer. This tool is often used in a first step to produce a solution that will be next optimized through parametric or shape optimization methods.

There are differents kinds of topology optimization tool [1]-[3] according to the optimization algorithm used. The tool studied in this paper is based on a genetic algorithm [4]. This kind of tool has already been used successfully for mechanical and electromechanical problem [6]-[9]. However, as the genetic algorithm uses a lot of evaluations during the process of optimization, many studies aimed to decrease this timing cost. One possible approach consists in dynamically changing the size of the population during the optimization process to maximize the performance of each generation [10]-[14].

In this paper, we study another approach to adapt dynamically the size of a population. This is based on the evolution of the size of the population form generation to generation. Section II describes the topology optimization tool used as reference tool. Section III details all modifications made to the genetic algorithm in order to allow the dynamic variation of the population size. Section IV presents the study case selected and used for the evaluation of the proposed adaptation. The section V gives the results and proposes an analysis of the latter.

II. Topology optimization tool

Topology optimization tool uses a combination of three modules (Figure 1) : an optimization algorithm, a material distribution formalism and a module of evaluation. The optimization algorithm allows to modify the value of all optimization parameters according to some informations as fitness or gradient from previous solutions. The material distribution formalism allows to translate design parameters into a material distribution inside a design space. And finally, the evaluation module the is used to evaluate each solution, i.e. To compute the fitness or the gradient.

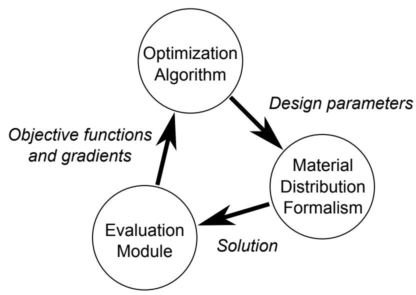

Figure 1. Three modules composing a topology optimization tool

Specified above in the introduction, a genetic algorithm is used as optimization algorithm. For the material distribution formalism and the module of evaluation, we choose respectively a Voronoï diagram [5] and a commercial software, Comsol©.

A. *The genetic algorithm*

The genetic algorithm is an algorithmic representation of the natural evolution described by Darwin's laws. The algorithm uses a set of N_s solutions, assimilated as a population of N_s individuals. The fitness of each individual represents its performance. This is used by the algorithm to evolve the population. At each iteration, called generation, a population of children is created from the main population. To keep the population size constant, only the best individuals of the main and the child population are preserved while the other are deleted.

The genetic code of the individual i in generation k is characterized by a vector of length N composed by a set of D discretes variables and a set of $N-D$ continuous variables :

$$\vec{X}_i^{(k)} = \left\{x_{i,1}^{(k)}, \ldots, x_{i,D}^{(k)}, x_{i,D+1}^{(k)}, \ldots, x_{i,N}^{(k)}\right\} \qquad (1)$$

For the same generation k, the matrix $\mathbf{X}^{(k)}$ corresponds to the population of N_s solutions where each line represents is a vector $\vec{X}_i^{(k)}$ characterizing/coding an individual.

The flowchart on Figure 2 shows the sequence of the implemented genetic algorithm. The genetic algorithm brings some change on the population by creating new solutions which will replace old worse solutions. These new individuals come from a genetic manipulation decomposed into three parts : selection of parents, reproduction of parents and elitist selection. After each part, a new population is created : a population of parents $\mathbf{X}^{(k)}$, a population of childs $\mathbf{X}^{(k)}$ and a population of elites $\mathbf{X}^{(k+1)}$ that will constitue the main population for the next generation.

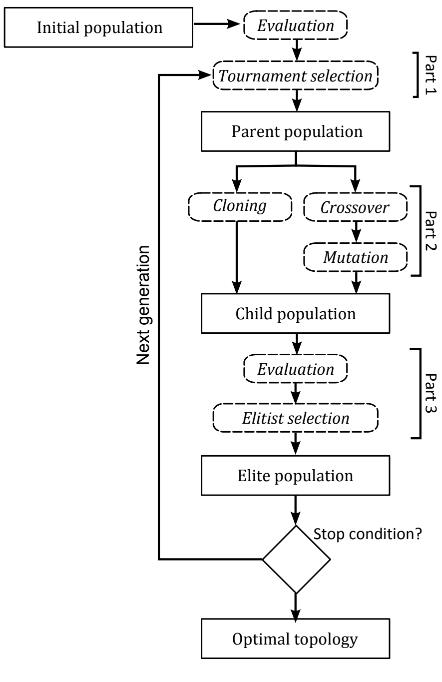

Figure 2. Genetic algorithm flowchart

The evolution of each population through the generation k, and the associated notations is illustrated on Figure 3. To simplify notation, next sections consider all parameters in generation k except when is mentionned.

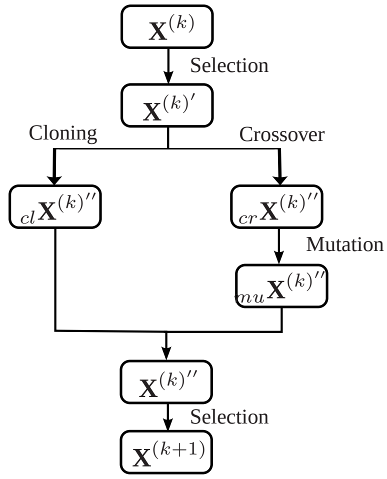

Figure 3. Matrix flux during the process of a generation

1) Part 1 : Selection of parents: To create the population of parents, $\mathbf{X}''$, the algorithm selects individuals inside the main population to compose an intermediate population, i.e. the parents population, $\mathbf{X}'$. In order to improve the current population, the genetic algorithm uses the fitness information to compute the probability of a solution to be selected for the next generation. Tournament is often used to select parents and create the parents population. If we need N_p individuals in the parents population, we must randomly select N_p pairs of individuals inside the main population with the possibility of repetition. For each pair, the best solution is selected for the parent population. This method gives a higher probability to the best solution to reproduce.

2) Part 2 : Reproduction: In order to preserve the existing population, a branch of the reproduction consist in cloning all individuals from the parent population, ${}_{cl}\mathbf{X}'' = \mathbf{X}'$. One other method, not illustrated in the Figure 2, consists in preserving the gene pool directly from the main population, ${}_{cl}\mathbf{X}'' = \mathbf{X}$.

However, the main part of the reproduction is made in parallel, on the second branch of the reproduction. This part is decomposed into two consecutives steps : the crossover and the mutation. The creation of a new solution, a child, is made by crossing the gene of two parents (2). This operation is performed with a probability p_c otherwisethe child is created by copying of a randomly selected parent. To implement this probability, an uniform continuous random vector is used: $R_i \sim U_c(0,1)$.

$${}_{cr}X_i'' = \begin{cases} Crossover\left(X_r', X_s'\right) & \text{If } R_i < p_c \\ Copy\left(X_r' \text{ or } X_s'\right) & \text{If } R_i \geq p_c \end{cases} \qquad (2)$$

with $r, s \sim U_d(0, N_s)$. To perform the crossover, a binary random mask, $\vec{m} \in \mathbb{1}^N$, is created in order to specify which parts from genetic information of the firts parent, $\vec{X}_r'$: $\{x_{r,1}, x_{r,2}, \ldots x_{r,N}\}$, and the second parent, $\vec{X}_s'$: $\{x_{s,1}, x_{s,2}, \ldots x_{s,N}\}$, will be combined:

$$Crossover :_{cr} x_{i,n} \quad = \begin{cases} x_{r,n} & \text{If } m_n = 0 \\ x_{s,n} & \text{If } m_n = 1 \end{cases} \quad (3)$$
$$\text{with } n = 1, 2, \ldots, N$$

After this first step, the mutation operation is performed to introduce some diversity inside the gene pool of the new population. This operation is applied to each individual, with a mutation probability, p_m, that determine if a gene is changed:

$$Mutation :_{mu} x_{i,n} \quad = \begin{cases} {}_{cr}x_{i,n} & \text{If } r_n < p_m \\ \alpha_n & \text{If } r_n > p_m \end{cases} \quad (4)$$
$$\text{with } n = 1, 2, \ldots, N$$

where α is an uniform random value that can take two value. If n is less than D then $\alpha_n \sim U_d\,(min, max)$ or else $\alpha_n \sim U_c\,(0, 1)$.

3) Part 3 : Selection for descendants: Next to the reproduction step, the population is composed of $2N_s$ individuals. An eltist selection consisting in selecting the N_s best solutions is therefore performed to compose the new population that will go through the next iteration.

B. The Voronoï formalism

The material distribution formalism involves two steps. The first one aimes at discretizing the design space into small cells. The second one consists in filling each cells with materials. In the case of a Voronoï formalism, the discretization of the design space is performed by a Voronoï diagram. A Voronoï diagram (Figure 4) is composed by a set of cells, each is defined by a reference point, named the Voronoï center.

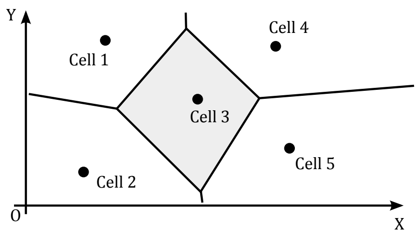

Figure 4. Diagram of Voronoï cells

Let $\vec{P}$ the set of Voronoï center in the Euclidean space with $V\,(P_i)$ the associated cell i. For all point x_e from the Euclidean space, there is one closest point P_k such as the point $x_e \in V\,(P_k)$. The Voronoï diagram is then borned inside a design space and a material is associated to each cell. Figure 5 gives a representation of the Voronoï formalism.

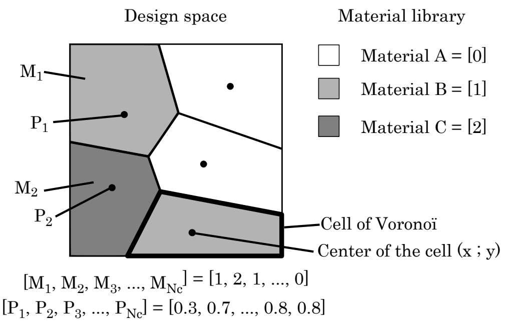

Figure 5. Voronoï formalism

Each cell is defined by a set of three parameters (M, x, y) with M the material and (x, y) the position of the Voronoï center. Each solution is therfore characterized by $3N_cel$ parameters describing the material distribution.

In this paper, we take into account the graphical aspect of the problem to performs the crossover. The graphical crossover uses a graphical pivot in order to select which Voronoï center are selected to the new solution (Figure 6). The graphical pivot implemented here is a randomly chosen circle in the design space. Voronoï centers outside and inside the circle respectively from parent 1 and from parent 2 are selected to perform the crossover. The graphical approach proved to be effective for topology optimization problem [16].

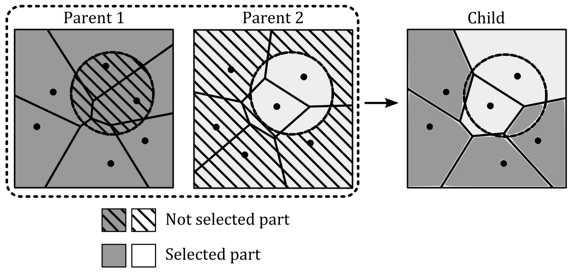

Figure 6. Diagram of Voronoï cells

C. The evaluation tool

The third module is the evaluation tool. It allows to compute the fitnesses of all solutions produced by the genetic algorithm. In this paper, it is composed by a finite element commercial software: $Comsol^{\copyright}$.

The strategy used to evaluate the solutions with the FEM sofware consists in creating a mesh adapted to the topology, i.e. adapted to the discretization defined by the Voronoï cells. This avoids problem with the degradation of the solution that occurs when the material distribution is projected on a fixed mesh (Figure 7).

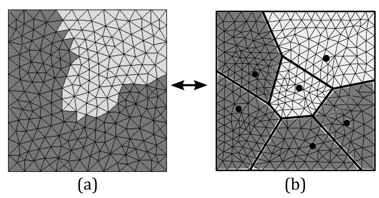

Figure 7. Material projection on a a mesh (a) and on an adapted mesh (b)

Automatic mesh adds an additional time cost, t_m, for all evaluations. However, if the time to compute the fitness, t_f, is mush greater than the time cost to create a mesh :

$$t_f \gg t_m, \quad (5)$$

the total cost of all evaluations does not really change.

III. Dynamic variation in population size

The aim of the genetic algorithm described in section II is to produce at each generation children improving the main population. The creation of each child is based on a random selection of parent with respect to fitnesses and the population size of each generation of children is defined as constant during all the optimization process. However, there is no possibility to know the best size of the population. A large population gives to the algorithm a better global search, the exploration while a small population gives to the algorithm a better local search, the intensification. During the optimization process , the algorithm must use exploration and intensification step when it is needed. With a static population, it is not possible to switch between these two steps and it is hard to find the best population size without some tests since it strongly depends on the problem studied.

With dynamic variation in population size, it is possible to change the number of individuals inside the population and thus to evolve between intensification ant exploration. The method proposed in this paper consists in assigning to each individual a coefficient of reproduction, $c_{r,i}$.

Each individual inside the main population is used once in the parents population. Pairs of parent are created, by random selection with respect to their rank r_i in the population. The probability p_i of selecting the parent i is calculated as follow (6) and (7),

$$t_i = \frac{1}{r_i} \tag{6}$$

$$p_i = 100 \times \frac{t_i}{\sum_{j=1}^{N_p} t_j} \tag{7}$$

Each pair of parents, i and j, creates through reproduction $c_{r,i} + c_{r,j}$ solutions. The total size of the population next the reproduction is therefor $\sum c_{r,i}$. The way of defining the reproduction coefficients is based on the following general idea. On the one hand, when a lot of children improve the main population, the population size has to increase to further favour the exploration. On the other hand, when a few of children improve the main population its size has to decrease for helping intensification.

We consider that to improve the main population, a child must have a better fitness than the median value of the main population. In other words, if the child is better than 50% of the main population, the population is improved.

Each parent is rewarded when its children improve the main population and punished when no children improve the main population during two generations. The reward and the punition consist respectively in incrementing and decrementing the reproduction step for the next reproduction step.

Each new solution have a coefficient of reproduction initialized to $c_{r,i} = 1$ and the best solution have a minimal value fixed to $c_{r,best} = 2$.

IV. Study case

To study the performance of the dynamic population size method, we applied it on a study case presenting the following characteristics :

- having a low cost in time computing;
- having a physical problem (electromagnetic);
- having a know global solution.

The study case we opted for is an electromagnetic inverse problem. With this, the global solution can be imposed, the problem is a physical problem, and with some hypothesis, the time computing can be low.

More precisely, it concerns the material quality measurement. In this field, electromagnetic sensors, like the one illustrated on Figure 8, are used to detect holes or cracks inside materials by observin the response of the sensor to an AC excitation. To decreasing the time cost of one evaluation, some simplification are made. The first consists in reducing the initial 3D problem to a 2D problem. Second one, the current is a DC-current and finally, we can obtain a map of the magnetic flux density inside the design space.

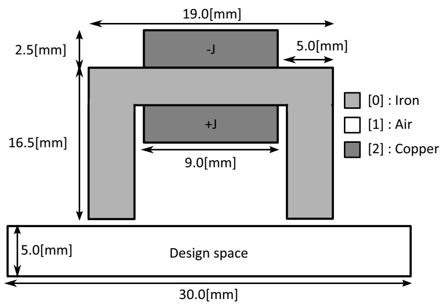

Figure 8. Problem characterization

The last simplification consists in searching the material distribution in the design space (Figure 8) that reproduces as best the magnetic field distribution produced by the sensor on an a priori known distribution of material instead of the response of the sensor to a specific excitation.

The problem has therefore one fitness and no constraints. The fitness is the difference between the target magnetic field distribution, $B(\xi_1, \xi_2)$ and the one produced by the material distribution produced by the optimization algorithm $B'(\xi_1, \xi_2)$:

$$f\left(\vec{P}, \vec{M}\right) = \int\int \frac{|B - B'|^2}{B} d\xi_1 d\xi_2. \tag{8}$$

This we can write in a discreet version with 50×50 measure points, $\xi'_{1,\psi} = \frac{0.03\psi}{50}$ and $\xi'_{2,\phi} = \frac{0.005\phi}{50}$:

$$\sum_{\psi=1}^{50}\sum_{\phi=1}^{50} \left| \frac{B\left(\xi'_{1,\psi}, \xi'_{2,\phi}\right) - B'\left(\xi'_{1,\psi}, \xi'_{2,\phi}\right)}{B\left(\xi'_{1,\psi}, \xi'_{2,\phi}\right)} \right|^2 \tag{9}$$

To perform the material distribution, ony two materials are used in the design space: air and iron. Parameters are defined by :

$$\vec{M} \in 1^{N_{cel}} \; and \; \vec{P} \in \Re^{2N_{cel}}_{\{0;1\}} \tag{10}$$

The reference solution is illustrated on Figure 9. It includes one crack and one hole. The target field distribution $B(\xi_1, \xi_2)$ produced by the reference solution is shown on Figure 10.

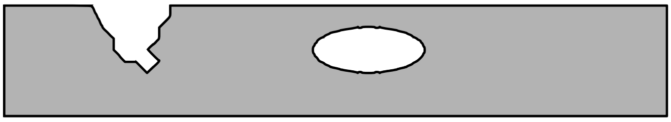

Figure 9. Material distribution of the target solution

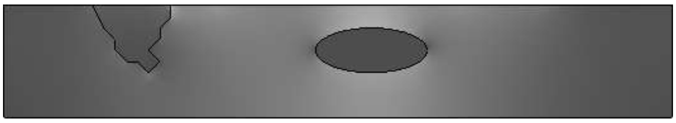

Figure 10. Target magnetic flux density field (norm) with a grayscale density illustration. Dark color corresponds to the maximum value

V. Results

Two sizes of populations, 10 and 100 individuals, were used to shows the behavior of the algorithm during the opimization process, both for 20.000 evaluations. The choice of these value for the population size is important because we know that for this problem, 10 individuals gives better results than 100 individuals in static mode. However, this result comes from empirical tests and we do not know exactly which value is finally the best. All solutions are compared through the number of evaluations because of the variability in the number of generations. For both configurations, the mutation probability, the crossover probability and the initial number of Voronoï cells are respectively set to $\frac{100}{N_{cel}}\%$, 90% and 100 cells.

Figure 11 shows the volution of the median value of the children population obtained from 5 independant runs executed with dynamic and static population size for an initial population size of 10 and 100 individuals. The graphic shows the slightly better performance of the dynamic mode compared to the static mode in the case of 10 and 100 individuals. The exploration observed at the beginning of the optimization process seems boosted while the intensification appearing at the end of the optimization not. Figure 11 highlight the static and dynamic mode with 10 individuals are better than with 100 individuals. But in the case of dynamic population size with 100 individuals, the evolution of the best solution during the process of optimization is almost the same than the evolution of the best solution for the case of static population with 10 individuals. Statistical values are noted in Table I.

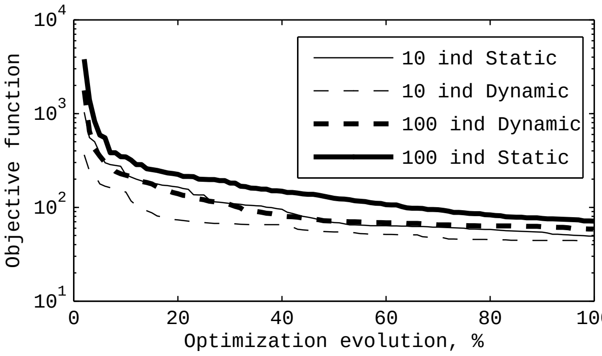

Figure 11. Graphic of convergence with median value from all results : a comparison between static population size and dynamic population size.

Table I
Statistical values of all simulations

	static	dynamic	static	dynamic
Population size	10	10	100	100
Minimum	34	39 (+14%)	62	41 (-33.9%)
Median	49	44 (-10.3%)	71	57 (-19.8%))
Maximum	115	60 (-47.8%))	117	125(+8.5%)

Figure 12 confirms that small populations are better than big populations for this problem. With an initial children population size set to 100 individuals, the ppulation size decreases continuously. The minimum value obtained is 2 individuals and the optimization finish with a children population size set to 11 individuals.

With an initial population of 10 individuals, Figure 13 shows that the population size evovles slightly, staying always between 2 and 15 individuals. The strategy of evolution of the children population size does not allow population size lower than 2 because the minimal reproduction coefficient of the best solution is always set to 2.

The evolutions of the population size observed on Figure 11 and 12 tend to demonstrate the ability of the proposed methode to evolve towards a population size better suites to the adressed problem. These evolutions also suggest that the dynamic of the population size evolution is not high enough. Indeed, over 20.000 evalutions, the population size just reached population of about ten individuals.

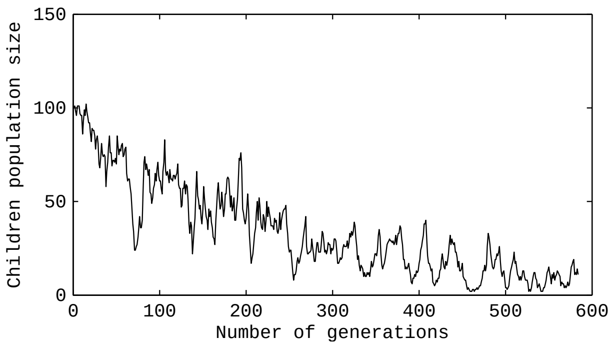

Figure 12. Evolution of the children population size during the process of optimization with an initial value of 100 individuals for the best run

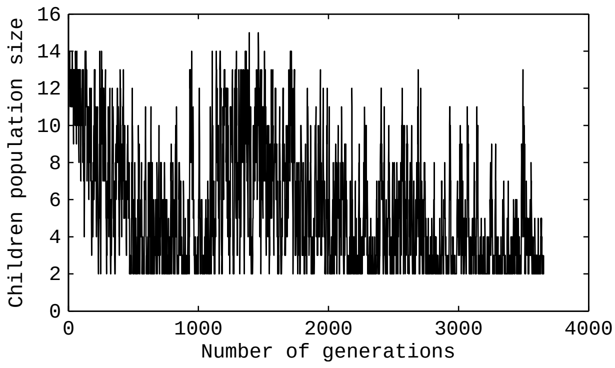

Figure 13. Evolution of the children population size during the process of optimization with an initial value of 100 individuals for the best run

Figures 14 and 15 shows the final material distribution of the best solution for the four cases studied. We can see that all solutions have converged to a local optimum. Two characteristics are nevertheless identifiable: the cracks is more larger and an additional crack appear on the right side of the design space. With 10 individuals, the intensification step is favored over the case with 100 individuals. The main outcome is a better drawn hole.

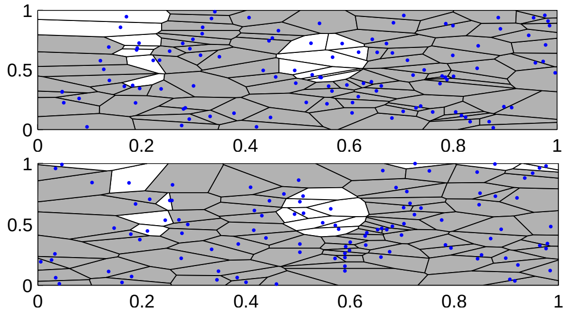

Figure 14. Best material distribution with an initial population size of 10 individuals : static (top) and dynamic (bottom) population size

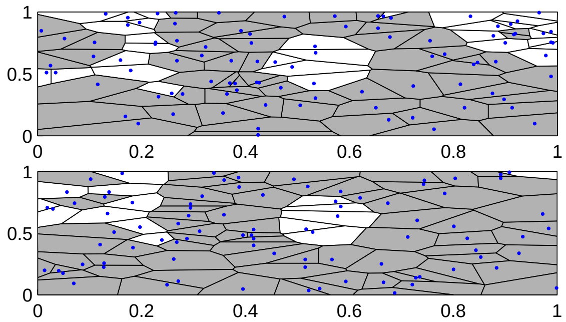

Figure 15. Best material distribution with an initial population size of 100 individuals : static (top) and dynamic (bottom) population size

VI. Conclusions

With optimization algorithms working with populations of solutions, it is often hard to define wich population size is the best to solve the problem. For helping the optimization algorithm in the exploration step, the population size must have the same order of magnitude than the number of parameters. But this value can be change during the optimization process while it evolves to intensification. It is therefore important to implement a strategy of evolution of the population size to adapt it to the needs of the algorithm.

In this paper, we introduced a coefficient of reproduction for each individual inside the main population that gives the possibility to control the size of the population of children at each generation. Each parent creates a number of children depending on a coefficient of reproduction. This coefficient change at each generation, according to the performance of children. With the strategy proposed, the evolution of the number of individuals is inevitable when the algorithm converge to the final solution. If we consider that it is more and more hard to find a better soluiton, the coefficient will tend to decrease. The difficulty to find a solution is measured by the coefficient of reproduction and the most effective reaction seems to decrease the size of the population.

In ordre to evaluate the ability of this strategy to dynamically adapting the population size, we applied it on a study case for different initial population size both with static and dynamic population size. We observed a decreasing evolution of the children population size during the optimization. With this strategy, both configurations give better results.

For future work, it is also possible to change the main population size to favor a set of best solutions during the reproduction in order to help the exploration or the intensification step. Effectively, with a big population size, best individuals will have a high probability to reproduce with worst individuals and with a small population reproduction between solution with a high level of similarity is favored.

References

[1] M.P. Bendsoe, N. Kikuchi, "Generating optimal topologies in structural design using a homogenization method," Comput. Methods Appl. Mech. Eng. 71 (1988) 197-224.

[2] C. D. Chapman, K. Saitou and M. J. Jakiela, "Genetic algorithms as an approach to configuration and topology design. ASME Journal ofMechanical Design," 116:1005-1012, 1994.

[3] K. Tai, S. Wang, S. Akhtar and J. Prasad, "Structural topology optimization using a genetic algorithm and a morphological representation of geometry," High Performance Computation for Engineered Systems (HPCES), 2003.

[4] K. Deb, S. Agrawal, A. Pratab, T. Meyarivan, "A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multiobjective Optimization : NSGA II," Proc. PPSN VI, September 2000, pp. 849-858.

[5] S.S. Skiena, "The Algorithm Design Manual," London: Springer London, 2008.

[6] E. Jensen, "topological structural design using genetic algorithms," PhD thesis, Purdue University, November 1992.

[7] C.Kane and M. Schoenauer, "Genetic operators for two-dimensionnal shape optimization," Artificial Evolution, Springer, pp. 355-369, 1996.

[8] Denies J., Ben Ahmed H., Dehez B., "Design of a ferrofluid micropump using a topological optimization method," Proceedings of the 2009 Electromotion, (2009).

[9] J. Denies, B. Dehez, and H. B. Ahmed, "Simulated annealing and genetic algorithms in topology optimization tools : a comparison through the design of a switched reluctance machine," In Proceedings of the International Symposium on Power Electronics, Electrical Drives, Automation and Motion - Speedam 2010, Pisa, Italy, pages 1247-1252. IEEE Press, Juny 2010.

[10] J. Arabas, Z. Michalewicz and J. Mulawka, "GAVaPS - a Genetic Algorithm with Varying Population," Proceedings of the First IEEE Conference on Evolutionary Computation, 1994. IEEE World Congress on Computational Intelligence, 73-78 vol.1, 27-29 Jun 1994.

[11] G. Harik and F. Lobo, "A parameter-less genetic algorithm," in: W. Banzhaf, J. Daida, A.E. Eiben, M.H. Garzon, V. Honava, M. Jakiela, R.E. Smith (Eds.), GECCO-99: Procesedings of the Genetic and Evolutionary Computation Conference, Margan Kaufmann, San Francisco, CA, 1999, pp. 258-267.

[12] F.G. Lobo, "The parameter-less genetic algorithm: Rational and automated parameter selection for simplified genetic algorithm operation," Doctoral dissertation, Universidade Nova de Lisboa, Lisboa, 2000.

[13] F. G. Lobo and D. E. Goldberg, "The parameter-less genetic algorithm in practice," Information Sciences, Vo. 167, 1-4, pp:217-232, 2004.

[14] X.H. Shi, Y.C. Liang, H.P. Lee and L.M. Wang, "An improved GA and a novel PSO-GA-based hybrid algorithm," Information Processing Letters, v.93 N.5, P.255-261, March, 2005

[15] J. Denies, H. Ben Ahmed and B. Dehez, "Topology Optimization for Solving Electromagnetic Inverse Problem," In: 1st International ICST Workshop on New Computational Methods for Inverse Problems, NCMIP'11, May 16, 2011, Cachan, France, 5 p.

[16] C. Kane, "Genetic algorithms and topology optimization (in French)," Doctor of Philosophy Thesis, Ecole Polytechnique, September, 1995