Study of a topology optimization tool associating a hybrid algorithm with a Voronoï-based formalism for the design of electromagnetic devices

J. Denies, H. Ben Ahmed, F. Glineur and B. Dehez

Abstract—**This paper presents and compares three topology optimization tools differing from the optimization algorithms they implement. The first two are respectively based on a local optimization algorithm, i.e. a Nelder-Mead method, and a global one, i.e. a genetic algorithm. The third uses a hybrid algorithm associating the local and the global algorithms used in the first two. Each of them implements the same material distribution formalism, i.e. a dynamic formalism based on Voronoi cells. The results obtained on a practical study case, the design of an electromagnet, show the better performance of the hybrid algorithm as well in terms of speed of the convergence as in terms of quality of the final solution and therefore the interest of coupling the exploration abilty of the global search with the intensification abilty of the local search.**

Index Terms—**Topology optimization, hybrid algorithm, genetic algorithm, Nelder-Mead method, design, electromagnet**

I. Introduction

TOPOLOGY optimization is a particular domain of optimization bringing the possibility to design new devices without initial representation of the solution. This approach need to explore the very large domain of solutions by distributing some specified materials in a given design space in order to find the best solution according to objective functions.

The topology optimization tools generally combine three modules: an optimization algorithm, a material distribution formalism and an evaluation tool. The material distribution formalism links the design parameters manipulated by the optimization algorithm to the solution assessed by the evaluation tool.

Among the several existing distribution formalisms [1]-[5], the most used is probably the bit-array formalism [1], [2]. This is based on a static subdivision of the design space into small square cells. In [5]-[7], this formalism is compared to a formalism based on a dynamic subdivision of the design space with Voronoï cells [8]. These studies show that, when coupled with a genetic algorithm, this formalism allows to quickly find the best topology, but it has great difficulty to take advantage of its dynamic nature, i.e. its ability of changing the subdivision of the design space during the optimization, to bring fine improvements to the solution.

This behaviour can be explained by the particular nature of the genetic algorithm. Indeed, the latter is better suited to carry out global search than a local one. In this logic, and to make the most of the potential of the Voronoi based formalism, the research for the optimal solution can be implemented in two phases: a first phase of exploration conducted by a global search algorithm followed by a phase intensification led by a local search algorithm.

This constitutes the main contribution of this paper, namely, integrating a hybrid optimization algorithm with a Voronoï based material distribution formalism in a topology optimization tool. The hybrid algorithm includes a genetic algorithm [9] for the global search and Nelder-Mead method [10] for the local search. The tool is tested on a practical study case, the design of an electromagnet, and its effectiveness is assessed by comparing its results with those obtain with the tools using the genetic and the Nelder-Mead algorithms separately.

This paper is organized as follow. Section II gives a description of the material distribution formalism. Section III describes the global and local algorithms used and the way they are adaptated with the distribution formalism. Section IV presents the study case and first results. Section VI gives new results obtained with additional constraints close related to the industrial needs of the actual design. Finally, section VII concludes this paper.

II. Material Distribution Formalism

The material distribution formalism allows connecting the genotype, design parameters, with the phenotype, the solution. The Voronoï-based formalism use the structure of the Voronoï diagrams proposed by G. Voronoï [8]. This structure partitions an opened or a closed domain into small subdomains, named Voronoï cells. This division of space is performed by the use of a set of references points, the cell centers. Each point from the domain is associated to the nearest cell center. This allocation creates a cellular subdivision.

In practice, the domain is bounded to fit the design space and a material, chosen from a predifined material library, is associated to each reference point (Fig. 1). Two types of design parameters correspond to this sepcific material distribution formalism. The first includes continuous parameters for the

J. Denies is funded by a Belgian F.R.I.A. grant.

J. Denies and B. Dehez are with CEREM - Université Catholique de Louvain, Place du levant, 3, Louvain-la-Neuve, 1348 (Belgium) (e-mail: jonathan.denies@uclouvain.be and bruno.dehez@uclouvain.be).

F. Glineur is with CORE - Université Catholique de Louvain, Voie du Roman Pays, 34, Louvain-la-Neuve, 1348 (Belgium) (e-mail: francois.glineur@uclouvain.be).

H. Ben Ahmed is with SATIE laboratory - Ecole Normale Supérieure de Cachan antenne de Bretagne, Campus de Ker-Lann, 35170, Bruz (France) (e-mail: benahmed@bretagne.ens-cachan.fr).

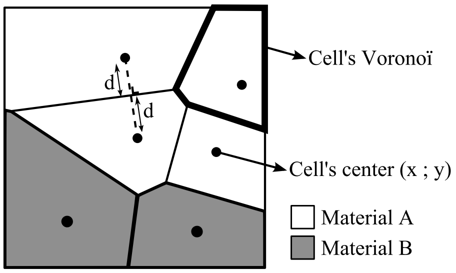

Fig. 1. Example of a closed domain subdivided into small Voronoï cells with a given distribution of two materials

cell position and the second discrete parameters to refer the material associated to each cell.

III. Optimization algorithms

The optimization algorithms can be classified into two main categories, local and global, that can be related to two aspects of the solution research, intensification and exploration respectively.

The local search algorithms look for better solution in the neighbours of an initial solution. Consequently, if the problem is not convex, the final solution may depend on the initial value assigned to the design parameters, i.e. the initial solution. However, with these methods, the convergence is very fast.

The global search algorithms look for a better solution by exploring all the solution space. This approach is ideal to find the basin of converence of the best solution but its convergence is often very slow.

The respective advantages of these two methods can be obtained by combining a global search algorithm with a local one.

In this paper, both methods are studied and their combination too, in order to present their respective performance.

A. Global optimization approach: a genetic algorithm

The genetic algorithms are an evolutionary algorithms using the law of evolution from Darwin to evolve a set of solutions [9]. It is considered that a good solution, in terms of the objectives functions, has more chance than a bad solution to survive inside the population and to reproduce. The main aim of this algorithm consists to quarry good information from some good solutions, parents, to create new solutions, children, hoping those children are better.

Fig. 2 presents a schematic representation of the population's evolution during the optimization process broken into 4 steps.

In the first step, the algorithm begins by computing a random initial population. Each solution is described by a vector, gathering all design parameters uses for the optimization. Design parameters, in this study, are positions and material definition of cells center from the Voronoï-based formalism.

In the second step, a parent's population is selected with repetition. To compute this selection, a method consists in considering a segment AB with a unit size and to decompose this segment into N_s pieces: $AC_1, C_1C_2, \cdots, CN_{s-2}CN_{s-1}, CN_{s-1}B$. N_s is the number of solutions within the population. Each piece's size is proportional to the performance of the solution. N_s points are placed on the main segment AB at regular intervals. For each point, the algorithm selects the corresponding solution and makes a copy inside the parent's population.

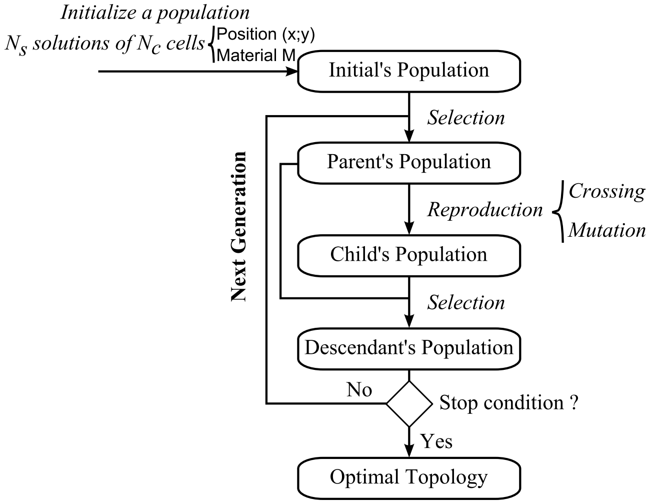

Fig. 2. Schematic representation of a cycle optimization for the genetic algorithm

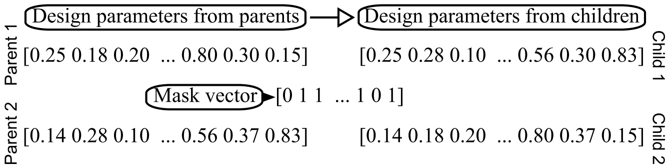

Fig. 3. Example of a uniform crossing operation

The third step consists to reproduce all parents by pairs following two reproduction mechanisms. The first is the crossing and second is the mutation. The crossing is implemented as a uniform mechanisms. It consists to define a random binary mask vector as same size than the numbers of design parameters. When a value of this vector is equal to 0, the child 1 keep the value from the parent 1 and the child 2 keep the value from the parent 2. And when the value is equal to 1, the operation is inversed (Fig. 3). The mutation consists to make some random change inside each child's vector. For each value from the vector, a random value between 0 and 1 is compared with a mutation rate defined. If the value is greater than the mutation rate, a new value is randomly chosen to replace the old value.

The fourth step consists to preserve a population of N_s solution for the next generation. A population of $2N_s$ solutions is obtained by mixing child's population with parent's population and only the N_s best solutions according to the evaluation function are selected.

This cycle is named "generation" by comparison with the natural evolution.

B. Local optimization approach: a Nelder-Mead method

Nelder-Mead is a derivative-free optimization (DFO) algorithm and is based on a simplex in the design parameters space to make some operations of local search. Each node of this simplex corresponds to a solution. The aim is to use the information from initial solution trying to create new better solutions. This algorithm is broken into 7 steps with some potential recurrence.

In the first step, the algorithm creates the initial simplex by creating $M+1$ solutions, with M equal to the dimension of the design parameters space. The simplex is defined by $\{p_1, p_2, \cdots p_{M+1}\}$ where p_i are design parameters vector. The value of $\{p_2, \cdots p_{M+1}\}$ is randomly chosen with the relation:

$$p_i = p_1 + \delta r \tag{1}$$

where δ is a control parameter and r is a random vector between $-\frac{1}{2}$ and $\frac{1}{2}$.

The second step consists in evaluating these solutions with the evaluation function $f(p)$ to obtain $f(p_1), f(p_2), \cdots, f(p_{M+1})$ and to rearrange the index i of f_i to respect the following inequality $f(p_1) < f(p_2) < \cdots < f(p_{M+1})$.

For the third step, the center of gravity vector p_0 is calculated from all point without p_{M+1}.

The fourth step consists in computing the reflection point according to the relation:

$$p_r = p_0 + \alpha (p_0 - p_{M+1}) \tag{2}$$

with α, the reflection factor $(0 < \alpha < 1)$ defined by the user. If the results respect the inequality $f(p_1) < f(p_r) < f(p_M)$, then p_r replaces p_{M+1} and go to step two.

In the fifth step, if the reflection point is the best, $f(p_r) < f(p_1)$, then an expanded point is computed by using the relation:

$$p_e = p_0 + \gamma (p_0 - p_{M+1}) \tag{3}$$

with γ, the expanded factor $(\gamma > 1)$ defined by user. If the expanded point is better than the reflection point, then the expanded point replace the worst point else the reflection point replaces the worst point. For both cases, go to step two.

The sixth step consists in changing the orientation of the exploration by a contraction. The contracted point is computed by:

$$p_c = p_{M+1} + \rho (p_0 - p_{M+1}) \tag{4}$$

with ρ, the contraction factor $(0 < \rho < 1)$, defined by the user. If the contraction point is better than the worst point, we obtain a new simplex by replace it and go to step two.

In the seventh step, for all but the best point, replaces points with the relation:

$$p_i = p_1 + \lambda (p_i - p_1) \text{ for all } i \in \{2, \cdots, M+1\} \tag{5}$$

with λ, the reduction factor $(\lambda < 1)$, and go to step two.

The algorithm finished when a stop criterion is satisfied. This stop criterion is chosen by the user and can contain multiple definitions, for example the number Q_0 of iterations of the optimization process or a final value for the evaluation function. A graphical representation of this method is presented on Fig. 4.

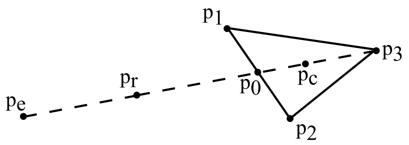

Fig. 4. Graphical representation, with $M = 2$, for the reflection point, extension point and contraction point.

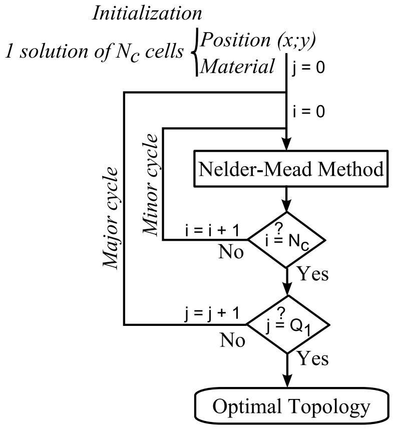

Fig. 5. Nelder-Mead algorithm uses in this paper with N_c, the number of cells and Q_1 the number of major cycles.

To adapt this algorithm with the Voronoï-based formalism, the cell position only are defined as design parameters. The cell material is kept unchanged. The main idea is to move locally one cell at a time to find a better solution. This approach requires less parameter for the Nelder-Mead method than move all cells together. The value of Q_0 is fixed to 4. A cycle of optimization evolves each cell individually. This is the minor cycle. A major cycle is used to execute Q_1 times the Nelder-Mead method for each cell after each minor cycle to take into consideration the high degree of relation between all cells (Fig. 5).

With this algorithm, design parameters are not restricted to a close domain. The position of a cell can exit the design space. In this case, the cell no longer participates to the solution.

C. Hybrid algorithm

In a hybrid optimization, there are two main steps: exploration and intensification. The first explore at lower cost the research space to find an optimal initial solution for an intensification exploration. The hybrid algorithm presented here combines a genetic algorithm with a Nelder-Mead method. A combination between two different algorithms needs to establish some adaptations, five in our case (Fig. 6).

First, the Nelder-Mead method works only with the best solution from the genetic algorithm. At each new generation, a number Q_1 of cycles are executed. The value of Q_1 changes with the generation number:

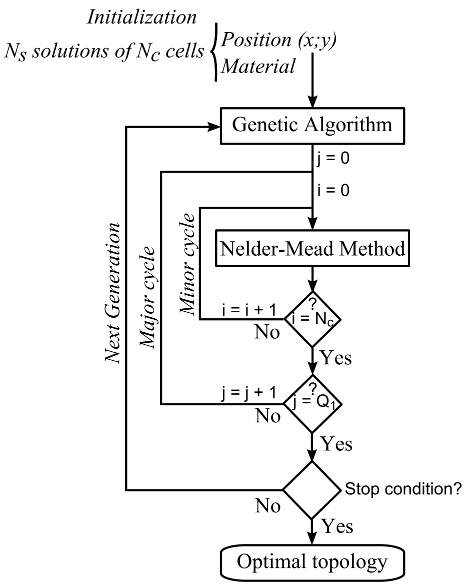

Fig. 6. Hybrid algorithm uses in this paper with N_c, the number of cells and Q_1 the number of major cycles.

$$Q_1 = \text{generation number} - 1 \tag{6}$$

Second, in order to eliminate all redundancies, the design parameters are distributed between both algorithms. The genetic algorithm, to maximize his exploration, works on the cell material while the Nelder-Mead method uses the cell position to change the discretization of the design space and also the position of the boundary to make an intensive search.

Third, the genetic algorithm works only with cell material for mutation but can work with all design parameters for the crossing operations to accelerate the exploration.

Fourth, the contribution of both algorithms is unbalanced during the process of optimization. At the beginning, we choose to favor to the exploration process. The contribution of the exploration decreases with the number of generations and by contrast, the contribution of the intensive research increases.

Fifth, the value of δ changes with the generation number following the relation:

$$\delta = 0.2\left(1 - \frac{i-1}{G}\right) \text{ with } i \in \{1, 2, \cdots, G\} \tag{7}$$

with i the generation number and G the maximum number of generations. This choice confines the intensification search.

IV. Study case

The study case consists in designing a cylindrical electromagnet in order to maximize the attraction density force σ applied on a magnetic core in the case of stable position:

$$\sigma = \frac{F_z}{A_{core}} = \frac{F_z}{\pi R_1^2} \tag{8}$$

with R_1 the radius of the core and F_z the axial force calculated by the Maxwell tensor. Fig. 7 presents the main characterisitcs of the study case, i.e. the design space and the dimensions.

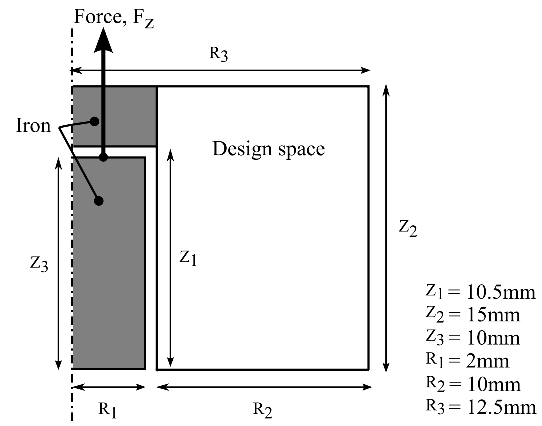

Fig. 7. Axisymmetric electromagnet configuration and design space definition

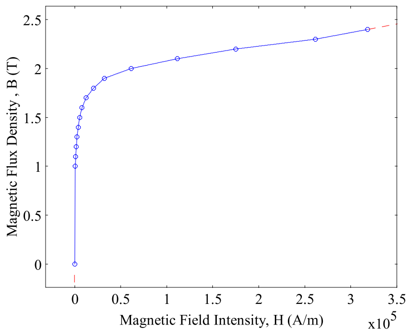

Fig. 8. Magnetic saturation characteristic for the iron used in the topological optimization (Source COMSOL 3.5a)

The materials available in the design space for optimization are iron, copper and air. The iron magnetic characteristic is presented on Fig. 8 and the copper is supposed crossed by a constant current density $J = 10^7 (Am^{-2})$. Finally, the optimization problem can be defined as follow:

$$\begin{array}{l} \max\sigma\left(x_i, y_i, m_i\right) \text{ for } i \in \{1, \cdots, N_s N_c\} \\ P < 10W \end{array} \tag{9}$$

with x_i, y_i, m_i, respectively the x- and y-position and the material of the i^{th} cell, and P, the total joules losses. The tool is coupled with the finite-elements software COMSOL 3.5a [11] to evaluate solutions. In order to exploit all information from Voronoï-based formalism, a new mesh is created for all new solutions.

The theoretical average force density σ_T is given by (10).

$$\sigma_T = \frac{F_Z}{A_{core}} \approx \frac{B^2}{2\mu_0} \tag{10}$$

with μ_0 the permeability of free space and B the norm of the magnetic flux on the superior face of the core.

V. Results and discussions

The results presented in this section, the final design and the convergence curve, are obtained with a number of evaluations fixed to 20.000 and a number of cells equal to 100.

A. Genetic algorithm

The genetic algorithm needs some configuration parameters: the mutation rate, the crossover rate, the population size and the number of generation. The probability to change the material of a cell, τ, is set proportional to the number of parameters $\left(\tau = \frac{2}{N_c}\right)$. The crossing probability is fixed to 90%. The population size is fixed to 100 solutions evolving during 200 generations. The final result is presented on Fig. 9. The maximal force obtained in the stable position, when the core is up, is $1.19\left(Nmm^{-2}\right)$.

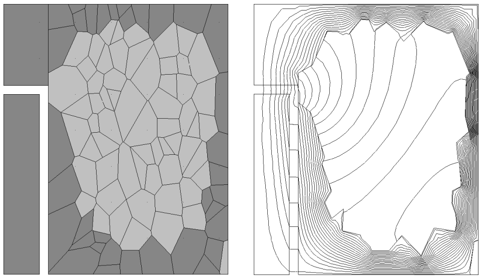

Fig. 9. Design of the final result for the genetic algorithm (left) and the corresponding isovalue for magnetic potential (right).

B. Nelder-Mead

The configuration parameters, α, γ, ρ and λ, from the Nelder-Mead method are respectively fixed to 1, 2, 0.5 and 2. The final result is presented in Fig. 10. The maximal average density force obtained is $1.01\left(Nmm^{-2}\right)$.

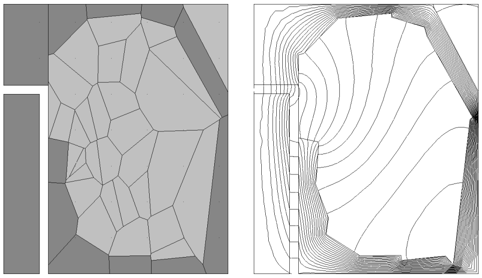

Fig. 10. Design of the final result for the Nelder-Mead algorithm (left) and the corresponding isovalue for magnetic potential (right).

C. Hybrid algorithm

All the configuration parameters are identical to the parameters from the genetic algorithm and Nelder-Mead case. The final result is presented in Fig. 11. The maximal average density force obtained is $1.21\left(Nmm^{-2}\right)$.

Fig. 12 presents the magnetic flux map from the hybrid solution. The size of the magnetic circuit is optimized to maximize the magnetic flux with saturation.

D. Discussion

The final designs of the three solutions are more or less identical. In general, the coil must be the widest possible but all algorithms have added a piece of iron alongside the iron core in order to redirect the magnetic flux. This case comes when the iron can be in saturation.

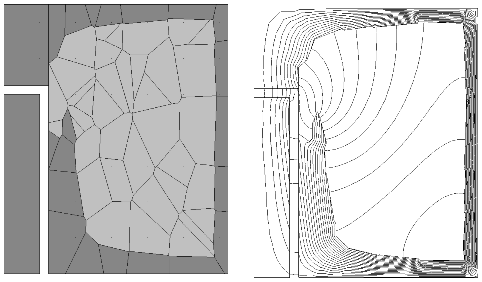

Fig. 11. Design of the final result for the hybrid algorithm (left) and the corresponding isovalue for magnetic potential (right).

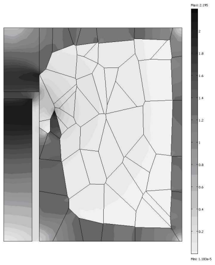

Fig. 12. Magnetic flux map, B (T) with a maximum of $2.195T$ (dark gray) and a minimum of $1.1810^{-5}T$ (light gray).

The genetic algorithm gives a solution near the global optimum. However, after 20.000 evaluations, the geometry of the notch is not correctly set out, demonstrating the slow convergence of this algorithm.

The Nelder-Mead method gives a local optimum near the global optimum. He failed to remove a cell of copper near the iron core. The geometry of the notch is well defined.

The hybrid algorithm presents the best solution. The value of the force is the best and the final design is the most easily exploitable to create the device.

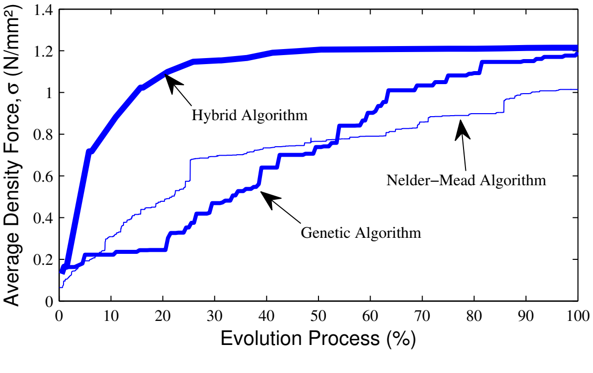

Fig. 13. Evolution of the average density force during the process

The convergence graphs of Fig. 13 confirm these observations. The genetic algorithm has a slow convergence but the final result is nearest of the best topology. The Nelder-Mead method, as for it, converges quickly at the beginning but cannot find the best topology and stay limited to a lower average density force than with the genetic algorithm. The convergence of the hybrid algorithm shows that it takes benefit of the genetic algorithm to start the Nelder-Mead method with the best topology and to converge quickly towards the best solution.

VI. A NEW STUDY CASE

In practice, the design of an electromagnetic device would change with the function to maximize or the constraints to satsify. This section aims at illustrating the advantages offer by topology optimization tools for performing these changes efficiently.

The device to optimize is the same as in the previous section but two new constraints are added. The first constraint is linked to the product of the supply current, $I(A)$, by the number of spires, $n(t)$. This product is fixed a priori to 1.000 At and the current density , $J(A/m^2)$, must respect the following inequality:

$$A_{copper} J \leq nI \tag{11}$$

with A_{copper}, the area of the copper section (m^2). The second constraint is linked to the value of the current density. It must guarantee that the Joule losses P_{Joule} stay under 10 W:

$$\frac{J^2 V}{\sigma_{copper}} \leq P_{Joule} \tag{12}$$

with V, the volume of the copper (m^3) and σ_{copper} the electric conductivity of the copper. Finals result for both new algorithms are presented in the Fig. 14. The density current value J is 1.20 and $1.13(Amm^{-2})$ and the maximal average density force obtained is 1.20 and $1.29(Nmm^{-2})$ respectively.

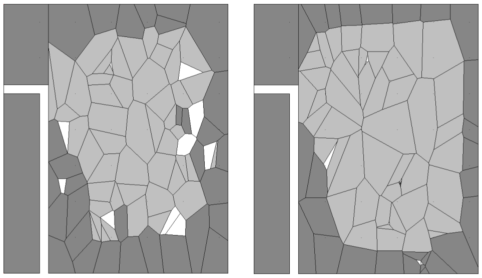

Fig. 14. Design of the final result for the ampere-turn constraint (left) and for the Joule losses constraint (rigth).

VII. CONCLUSION

In optimization, global and local search algorithms present different advantages. The first is well suited for approaching the global optimum but it can have some difficulty to improve the quality of the final solution. The second approach converges faster to a solution that can be a local optimum. The hybrid methods can mix advantages from both algorithms by combining them in one tool.

The hybrid algorithms, combining global and local search, are not new in optimization but this paper presents the possibility to use this approach with Voronoï-based formalism in a topology optimization tool. The results obtained with this tool show the effectiveness of hybrid algorithm compared to simple algorithm as well in terms of quality of the final solution as in terms of rate of convergence.

The hybrid algorithm proposed in this paper could be more deeply study to improve, notably, the coupling between global search and local search.

REFERENCES

[1] C. Kane et M. Schoenauer, "Genetic operators for two-dimentional shape optimization," Artificial Evolution, Springer, 1996, pp. 355-369.

[2] K. Tai and T.H. Chee, "Genetic algorithm with structural morphology representation for topology design optimization," Mechanics in Design-Proceedings of the 2nd International Conference on Mechanics in Design, Nottingham, 1998, pp. 827-36.

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

[4] M. Schoenauer, "Shape representations and evolution schemes," in Proceedings of the 5th Annual Conference on Evolutionary Programming, MIT Press, vol. 5, 1996, pp. 121-129

[5] M. Schoenauer, L. Kallel, and F. Jouve, "Mechanical inclusions identification by evolutionary computation," European Journal of Finite Elements, vol 5, no 5-6, 1996, pp. 619-648

[6] B. Dehez, J. Denies, H. Ben Ahmed, "Design of electromagnetic actuators using optimizing material distribution methods," in Proceedings of Int. Conf. on Electrical Machines (ICEM08), 2008, Vilamoura, Portugal, ISBN: 978-1-4244-1736-0

[7] J. Denies, B. Dehez, F. Glineur and H. Ben Ahmed, "Impact of the material distribution formalism on the effciency of evolutionary methods for topology optimization", in Recent Advances in Optimization and its Applications in Engineering, Springer Verlag, 2010

[8] S. S. Skiena, "Voronoi Diagrams," in The Algorithm Design Manual, New York: Springer-Verlag, 1997, pp. 358-360

[9] D. E. Goldberg, E. David, "Genetic Algorithms in Search Optimization and Machine Learning," Addison-Wesley Reading Menlo Park, 1989

[10] J.A. Nelder et R. Mead, "A simplex method for function minimization," The computer journal, vol. 7, 1965, p. 308.

[11] COMSOL Multiphysics 3.5a®, http://www.comsol.com

BIOGRAPHIES

Jonathan DENIES Jonathan DENIES was born in Tournai, Belgium, in 1984. He received the degree in electromechanical engineering from the Université Catholique de Louvain (UCL), Louvain-la-Neuve, Belgium. Since 2008, he was PhD student in the UCL. His research interests are in the field of topological optimization of electromagnetic devices by material distribution.

Bruno DEHEZ was born in Uccle, Belgium, in 1975. He received the degree in electromechanical engineering and the Ph.D. degree from the Université catholique de Louvain (UCL), Louvain-la-Neuve, Belgium, in 1998 and 2004, respectively. He is currently associate professor in the Centre for Research in Mechatronics (CEREM) at UCL. His research interests are in the field of design and optimization of dedicated actuator design.

Hamid BEN AHMED received the PhD degree and HDR degree in 1994, and 2006 respectively. From 1994 to 1997, he was a Research Associate in the SATIE laboratory. Since 1997, he was an Assistant Professor in the Mecatronique Department of Ecole Normale Supérieure de Cachan, Britttany Branch, France. His research interest include the design, modelling and optimisation of novel topologies of electromagnetic actuators and generators, and the optimisation of renewable energy systems. Phone: (33) 2 99 05 93 30, Fax: (33) 2 99 05 93 28 e-mail: benahmed@bretagne.ens-cachan.fr

François GLINEUR received engineering degrees from Faculté Polytechnique de Mons (Belgium) and Ecole supérieure d'électricité (France) in 1997, and a PhD in applied sciences from Faculté Polytechnique de Mons in 2001. Since 2002, he is an associate professor at Université catholique de Louvain, member of both the mathematical engineering departement and the Center for Operations Research and Econometrics. His research interests revolve mainly around convex optimization, including models, algorithms and applications.