Optimization of ultrasonic arrays design and setting using a differential evolution

Benoît Puel, Dominique Lesselier, Sylvain Chatillon, Pierre Calmon

Optimization of ultrasonic arrays design and setting using a Differential Evolution algorithm

Benoît Puel[a], Dominique Lesselier[b], Sylvain Chatillon[a], Pierre Calmon[a,*]

[a] *CEA, LIST, Department of Imaging & Simulation for Nondestructive Testing, F-91191, Gif-sur-Yvette cedex, France.*
[b] *Département de Recherche en Electromagnétisme, Laboratoire des Signaux et Systèmes (UMR8506, CNRS-Supélec-Univ. Paris Sud), 3 rue Joliot-Curie, 91192 Gif-sur-Yvette cedex, France.*

Abstract

The optimization of phased-array design and setting can be a delicate task when done *manually* through parametric studies. An optimization method based on an Evolutionary Algorithm and numerical simulation is proposed and evaluated. The Randomized Adaptive Differential Evolution has been adapted to meet the specificities of the addressed applications and in particular to solve multi-objective problems with the implementation of the concept of Pareto's optimal set of solutions. The algorithm has been implemented and connected to the ultrasonic simulation modules of the CIVA software used as forward model. The efficiency of the method is illustrated on two realistic cases of application: The optimization, considered as a mono-objective problem, of the position and delay laws of a flexible array inspecting a nozzle and the optimization of the design of a surrounded array and its delay laws, considered as a constrained bi-objective problem.

Keywords: ultrasonics, array, optimization, simulation, transducer

1. Introduction

Over the last decade, the use of ultrasonic arrays for Non Destructive Evaluation has continuously increased and phased-array techniques play now a major role in the field. Phased-arrays offer numerous advantages in terms of coverage, sensitivity and imaging capabilities. They allow to overpass some limitations of conventional ultrasonic inspections and to tackle increasingly complex inspections. The price to pay for these enhanced capabilities is the complexity increase of the design of the array (number and arrangement of elements) and the definition of the inspection setup (T-R functionalities, electronic delays, transducer displacements, etc.). Nowadays, efficient ultrasonic models are available which can help the expert in these tasks. Up to now this is done through computational parametric studies, which can be time-consuming, fastidious, and somehow delicate. Moreover, due to the necessarily small number of tested set of parameters such *manual* optimization may lead to one local optimum and miss a better global solution. The motivation of the present work is to propose, prototype and evaluate an automated optimization method based on the use of ultrasonic simulation tools connected to an Evolutionary Algorithm.

Evolutionary Algorithms (EAs) have been used by several authors to solve array optimization problems [1]. Chen *et al.* [2] used an Evolution Strategy to design a transducer by looking for the emission surface shape that minimizes the number of rings of an annular phased-array. This algorithm has been used to design an array in order to inspect titanium bullets [3]. Yang *et al.* [4] used a Genetic Algorithm to design a sparse array transducer. Gueudre *et al.* [5] used a Genetic Algorithm to characterize heterogeneous welds based on ultrasonic data inversion. EAs have also been applied in many other domains, e.g., optimization of focus pattern delay laws for ultrasound surgery using a Genetic Algorithm [6] or the retrieval of spheres buried in subsoil using a Differential Evolution [7]. In this work, among many available EAs, which makes non-trivial a choice, the so-called Randomized Adaptive Differential Evolution (RADE) [8] has been chosen in view of its good performance in solving engineering optimization problems and its small number of tuning parameters. RADE has been extended in order to solve multi-objective problems, as well as to handle constraints and criteria of accuracy on the variables. These algorithms, described in section 2, have been implemented and connected to the ultrasonic module of CIVA [9, 10]. This module allows to compute beams radiated by arrays in different materials and to simulate signals arising from defects in complex geometries [11]. Through this connection, it has been possible to evaluate the method on realistic NDE cases. In sections 3 and 4 we give two examples of such cases, illustrating respectively mono-objective and multi-objective problems. The first one concerns the optimization of the positioning of a flexible array on a complex part (a nozzle) and of the applied electronic delays. The second one (constrained bi-objective problem) concerns the design and set up of an array for the inner inspection of a pipe. A brief conclusion follows in section 5.

*Corresponding author
Email address: pierre.calmon@cea.fr (Pierre Calmon)

2. Optimization algorithm

The algorithms that have been used to optimize ultrasonic inspections are described in this section. Since RADE is able to solve only unconstrained mono-objective problems, extensions are proposed to handle constraints, accuracies, and multi-objective strategies.

2.1. Optimization problem

Whenever an engineering problem is to be solved by an EA, it should be transformed into an optimization problem that is given, in most general form, as

$$\begin{array}{lll} \text{Minimize} & f_l(X) & l = 1, \ldots, N_o \\ \text{Subject to} & g_m(X) \leq 0 & m = 1, \ldots, N_c \\ & x_j^L \leq x_j \leq x_j^U & j = 1, \ldots, N \end{array} \quad (1)$$

A candidate solution $X = [x_1; \ldots; x_N]$ is a vector of N variables tested to evaluate how well it solves the problem. The problem is defined with N_o objective functions f_l that have to be representative of the problem at hand; and N_c constraint functions g_m. The lower and upper bounds are respectively given by the vectors $X^L = [x_1^L; \ldots; x_N^L]$ and $X^U = [x_1^U; \ldots; x_N^U]$ and define the search range of all variables.

2.2. RADE

This algorithm is derived from differential evolution [12] (DE) with auto-regulation of one tuning parameter (the mutation factor). It makes a population of candidate solutions to converge to the optimum using three evolutionary operators: mutation, crossover, and selection. One loop of those operators is called a generation. The ith candidate solution of the kth generation is noted $X_i^{(k)}$ and its jth variable $x_{ij}^{(k)}$. The strategy used is denoted as DE/rand/1/bin [12].

The first step is the initialization of the population by a uniform distribution in the search domain defined by

$$x_{ij}^{(0)} = x_j^L + rand(x_j^U - x_j^L) \quad (2)$$

where *rand* is a random real between 0 and 1. Then, evolutionary operators are used to create new candidate solutions. Mutation creates new vectors ($V_i^{(k+1)} = [v_{i1}^{(k+1)}; \ldots; v_{iN}^{(k+1)}]$) by adding perturbations to a reference solution:

$$V_i^{(k+1)} = X_a^{(k)} + F_i(X_b^{(k)} - X_c^{(k)}) \quad (3)$$

where a, b and c are randomly chosen in the parent population such that $a \neq b \neq c \neq i$, and where F_i is the mutation factor, a self-adapted tuning parameter linked to the ith candidate solution of the population (X_i), as follows. F_i is initialized with a random value between 0.1 and 0.9; then, each five generations, F_i is reinitialized as well for half of the solutions $X_i^{(k)}$ with the lowest evolution of their objective function Γ (see [8] for more details):

$$\Gamma(X_i^{(k)}) = f(X_i^{(k-5)}) - f(X_i^{(k)}) \quad (4)$$

Crossover yields vectors $U_i^{(k+1)} = [u_{i1}^{(k+1)}; \ldots; u_{iN}^{(k+1)}]$ by combining variables between the new and parent solutions:

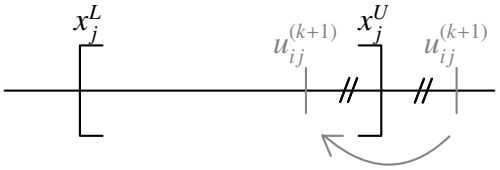

Figure 1: If a new variable $u_{ij}^{(k+1)}$ is generated out of the search range, it is put back inside by axis symmetry on the closest boundary.

$$u_{ij}^{(k+1)} = \begin{cases} x_{ij}^{(k)} & \text{if } rand > C_r \text{ or } j_i^{\text{rand}} = j \\ v_{ij}^{(k+1)} & \text{otherwise} \end{cases} \quad (5)$$

where j_i^{rand} is randomly chosen in interval $[1, N]$ for each solution i and $C_r \in [0.1; 0.7]$ is a user-defined parameter called crossover rate. Selection keeps solutions that improve in the objective function sense:

$$X_i^{(k+1)} = \begin{cases} U_i^{(k+1)} & \text{if } f(U_i^{(k+1)}) < f(X_i^{(k)}) \\ X_i^{(k)} & \text{otherwise} \end{cases} \quad (6)$$

This evolution is performed until stopping criteria are satisfied. Many [13] can be introduced but the most common one is a maximum number of generations (k_{max}), which will be the one chosen here.

2.3. Constraint handling

Two kinds of constraints have to be handled. First, since evolutionary operators provide diversity, some $U_i^{(k+1)}$ can be out of the search range defined by X^L and X^U. So, variables $u_{ij}^{(k+1)}$ are reintroduced by symmetry on the closest boundary (figure 1):

$$u_{ij}^{(k+1)} = \begin{cases} 2x_j^U - u_{ij}^{(k+1)} & \text{if } u_{ij}^{(k+1)} > x_j^U \\ 2x_j^L - u_{ij}^{(k+1)} & \text{if } u_{ij}^{(k+1)} < x_j^L \\ u_{ij}^{(k+1)} & \text{otherwise} \end{cases} \quad (7)$$

The other constraints are inequalities on some variables. The method used here is similar to the one in [14]. It is based on the notion of feasible solutions that respect all inequalities and non-feasible solutions that do not respect at least one. This information is given for a candidate solution X at the generation k by the constraint violation function ($g^{(k)}(X)$), defined by

$$g^{(k)}(X) = \sum_{m=1}^{n_c} \frac{\max(0, g_m(X))}{g_{m,Max}^{(k)}} \quad (8)$$

where $g_{m,Max}^{(k)}$ is the maximum value observed on the constraint g_m until generation k. Afterwards, to generate the least modifications in the algorithm, an equivalent objective function ($\eta^{(k)}(X)$) is defined by

$$\eta^{(k)}(X) = \begin{cases} 1 + \dfrac{g(X)}{n_c} & \text{if } g^{(k)}(X) > 0 \\ \dfrac{f(X) - f_{Min}^{(k)}}{f_{Max}^{(k)} - f_{Min}^{(k)}} & \text{otherwise} \end{cases} \quad (9)$$

where $f_{Min}^{(k)}$ and $f_{Max}^{(k)}$ are, respectively, the minimum and the maximum values observed on the objective function f until generation k.

This equivalent objective has interesting properties: if $0 \leq \eta^{(k)}(X) \leq 1$ then X is feasible; if $1 < \eta^{(k)}(X) \leq 2$ then X is non-feasible. Thus, it can replace the objective function for the selection (equation 6) while it respects Deb's proposition [15], as well as for the calculation (equation 4) of F_i since it does not depend on the order of magnitude due to normalization.

2.4. Accuracy handling

RADE had been designed to solve real-variable problems. Since it is interesting to take into account a precision step that determines the accuracy beyond which the solution is of no use to the end-user, a modification (accuracy handling) has been made as described now. The first task is to sample the search space with using the precision step. Then, whenever a new candidate solution ($U_i^{(k+1)}$) is created, the algorithm finds the closest solution on the chosen grid (called *formatted solution*) given by

$$\underline{u}_{ij}^{(k+1)} = x_j^L + x_j^P \text{floor}\left(\frac{u_{ij}^{(k+1)} - x_j^L}{x_j^P}\right) \tag{10}$$

where $\underline{U}_i^{(k+1)}$ is the formatted solution of $U_i^{(k+1)}$, x_j^P the precision of jth variable, and floor(a) computes the integer part of a.

Yet, one still has to care for diversity preservation since loss of diversity leads to being unable to generate useful new solutions. That is, the mutation operator (equation 3) cannot add a perturbation if $X_b^{(k)}$ and $X_c^{(k)}$ are too close or equal. So, if the formatted solution has already been computed, one is simply looking for the closest, non-computed solution on the grid within the neighborhood. In brief, computation time is spared since solutions too close to one another are not computed; and the optimal solution found makes sense, from the design point of view, since it has no unusable digits.

2.5. Multi-objective

The main difficulty faced with multi-objective problems is that two solutions cannot be easily compared, e.g., since one objective might be better reached and another one less or *vice-versa*. Then, so-called non-dominated sorting is often employed, as is implemented herein. The start point is that a solution X_1 dominates X_2 if $\mathcal{F}(X_1) \preceq \mathcal{F}(X_2)$ [16, chap. 8] where $\mathcal{F}(X) = [f_1(X), \ldots, f_{n_o}(X)]$, this being defined as

$$\mathcal{F}(X_1) \preceq \mathcal{F}(X_2) \Leftrightarrow \begin{cases} \forall i \in [1, n_o],\ f_i(X_1) \leq f_i(X_2) \\ \text{and } \mathcal{F}(X_1) \neq \mathcal{F}(X_2) \end{cases} \tag{11}$$

Obviously, solutions can be sorted out in order to produce sets of solutions with an attached rank. Rank one is defined by $\mathcal{S}_1$, which is the set of all solutions that are not dominated: the Pareto optimal set. Rank two is defined by $\mathcal{S}_2$, which is the set of all solutions only dominated by some solutions from $\mathcal{S}_1$. And so on for the next ranks. Obviously, $\mathcal{S}_1$ yields the best compromises between all objectives and is our main target. Figure 2 shows an example with five ranks of candidate solutions. The rank of a candidate solution X is noted as $\chi(X)$. This non-dominated sorting is performed on a database containing all feasible candidate solutions that have been evaluated since the initialization.

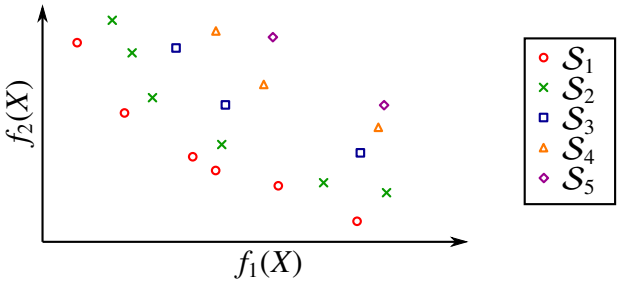

Figure 2: The Pareto optimal solutions (red circles) are all non-dominated solutions of all computed feasible candidate solutions ($\mathcal{S}_1 \cup \mathcal{S}_2 \cup \mathcal{S}_3 \cup \mathcal{S}_4 \cup \mathcal{S}_5$) in objective functions frame (f_1 and f_2).

Since candidate solutions can be ordered with non-dominated sorting, solutions from the same rank should be ordered so as to favor the most isolated ones, a good spread of solutions being aimed at in the Pareto optimal set. To achieve it, a diversity preservation method is applied, involving the normalized crowding-distance, as inspired from the crowding-distance method [17], properly modified to avoid infinite values. That is, the density of solutions surrounding X in its rank set is estimated via the average distance of two solutions on either side of X for each objective, and is denoted $\delta^{(k)}(X)$. The modification considers the distance between the first and last solutions of the rank to estimate the diversity of those solutions.

Then the selection is designed for multi-objective problems to first favor the solution with the best rank, then with the highest normalized crowding-distance[17]. An equivalent objective function ($\eta(X)$) for constrained multi-objective optimization that makes sense to perform the self-adaptation of F_i, is then defined as

$$\eta^{(k)}(X) = \begin{cases} \chi_{Max}^{(k)} + \dfrac{g(X)}{n_c} & \text{if } g^{(k)}(X) > 0 \\ \chi^{(k)}(X) - \dfrac{\delta^{(k)}(X)}{n_o} & \text{otherwise} \end{cases} \tag{12}$$

where $\chi_{Max}^{(k)}$ is the total number of ranks at generation k.

3. One example of mono-objective problem

3.1. Problem description

This first example given here concerns the inspection of a nozzle with a 8×8 flexible array operating at 2 MHz. The sought defects are breaking cracks on the inner surface at the junction of the two pipes and oriented radially relatively to the secondary pipe as depicted in figure 3. The control is performed from the outside. The probe is scanned on the conical part of the secondary pipe and delays (computed by CIVA) are applied in order to focus the beam in the region of interest. The defects are

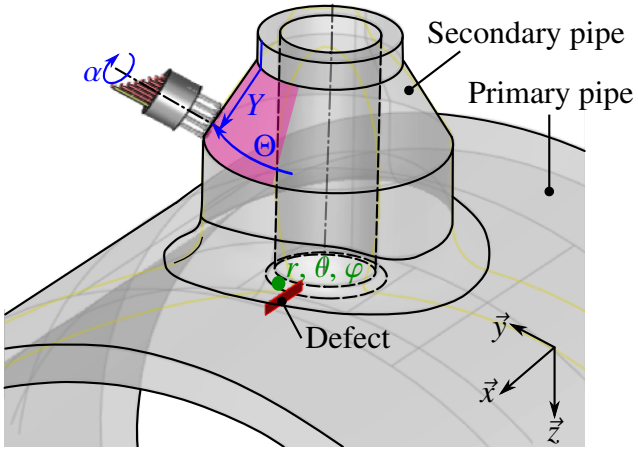

Figure 3: Schematic view of the nozzle inspection. The array probe is located on the outer surface of the secondary pipe. The sought defect is schematized at the junction of the two pipes. The focal point is materialized by the green dot.

name	description	interval	accuracy	unit
Position of the array on the nozzle				
Y	vertical position	$[30; 100]$	0.5	mm
Θ	angular position	$[0; 90]$	0.3	degree
α	probe rotation	$[0; 90]$	0.5	degree
Focal point coordinates relative to the probe center				
r	distance	$[100; 250]$	1	mm
θ	polar angle	$[-80; 80]$	0.3	degree
φ	azimuthal angle	$[-80; 80]$	0.3	degree

Table 1: Variables defined to optimize position and setting of a flexible array for a nozzle inspection.

detected thanks to the corner echo involving the reflexion on the inner surface of the nozzle. The objective of the optimization is to find both the position of the array and the focusing point (inputted in the delays computation) that maximize the amplitude of this echo for the different possible angular positions Θ of the defect. The problem is not trivial since this amplitude both depends on the deviation of the beam and on the angle of incidence on the defect. A *manual* optimization study had been previously performed by an expert for different possible angular positions of the defect. Solutions given by the presently proposed method can be compared to the solution obtained by the expert. Results presented here are obtained for a defect oriented along the primary pipe axis $\vec{x}$ ($\Theta = 0°$).

3.2. Variables and objective function

The variables of the problem are the position and orientation of the transducer and of the focal point witch is the input of the delay laws computation. They are listed in Table 1 with their search intervals and precision below which the information is considered useless.

Each candidate solution X generated by the algorithm is evaluated through one objective function f defined as

$$f(X) = -A_{max}(X) \tag{13}$$

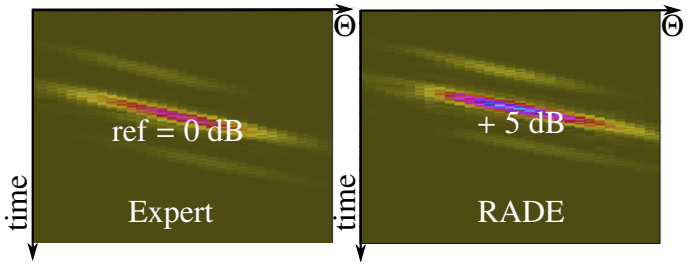

Figure 4: B-Scans comparison between solutions obtained by the expert and the RADE.

	Y	Θ	α	r	θ	φ
X_{exp}	91	49	20	153	44	0
X_{RADE}	94.5	54.6	22.5	188	39.4	0.4

Table 2: Parameters of the solutions obtained by the expert (X_{exp}) and the RADE (X_{RADE}).

where $A_{max}(X)$ is the amplitude of the corner echo computed by CIVA, the defect located at $\Theta = 0°$ being assumed to be planar with a height $H = 5$ mm.

3.3. Results and discussion

For all the defect orientations (Θ), which have been investigated the solutions given by the automated method are, at least, as good as the solutions *manually* obtained by the expert in terms of signal magnitude. In the particular case $\Theta = 0°$, the amplitude of the corner echo has been improved of +5 dB. This improvement is illustrated in figure 4 by the two corresponding simulated Bscan. The parameters of the two solutions are listed in Table 2. The following values of tuning parameters have been used to perform the optimization: $Cr = 0.7$, $NP = 30$ and $k_{max} = 100$. In this example then 3030 candidate solutions have been generated and evaluated by the algorithm. The mean computation time required for one simulation being 42 seconds, the optimization process asked for about 36h total running time, on a standard PC. The evolution of the amplitude corresponding to the solution selected at each generation is reported in figure 5 for two versions of the algorithm: with and without accuracy handling. The amplitudes displayed are given in decibels, the 0 dB level being the *manual* solution. It can be seen that the accuracy handling significantly reduces the number of generations and therefore the corresponding computation time required to converge to the optimized solution: 33 generations and 12 hours *vs.* 50 generations et 18 hours.

4. One example of multi-objective problem

4.1. Problem description

The second example given in this contribution concerns the inner inspection of a stainless steel pipe by a surrounded array, depicted on figure 6. It is solved as a constrained multi-objective problem. The inner diameter of the pipe is 50 mm and its thickness 5 mm. The aim of the control is to detect cracks oriented along the pipe axis with 45° compressional waves. The

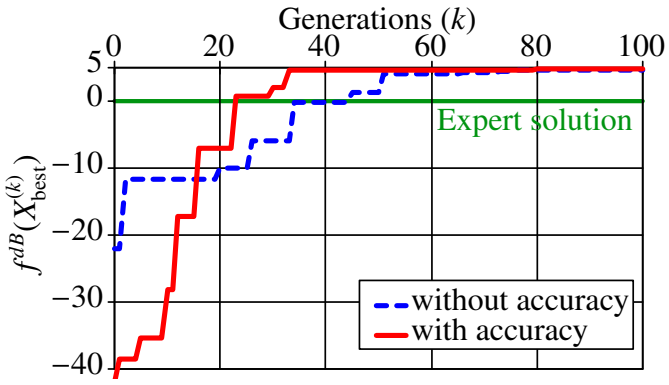

Figure 5: Evolution of the objective function (the maximum of the echo amplitude) during the optimization process with (in blue) and without (in red) accuracy handling. The manual solution corresponds to the 0 dB level.

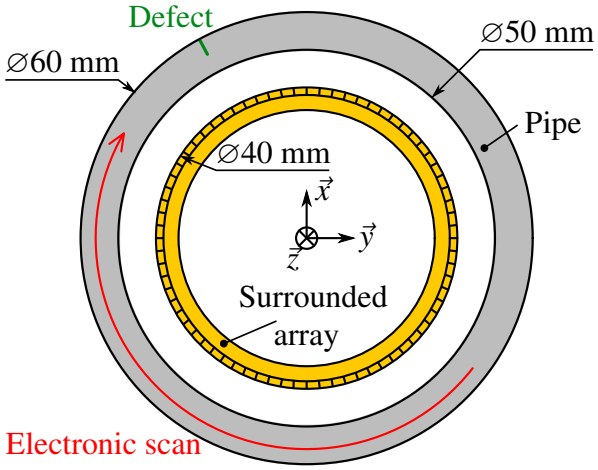

Figure 6: Schematic view of the pipe inspection problem.

array is moved along the axis $\vec{z}$ while electronic commutation is used to inspect the whole section of the pipe. The pipe is filled by water. The goal is to optimize the design of the array, and the electronic setting: the electronic commutation and applied delay laws. The number of available channels, and therefore the maximal number of elements of the array, is fixed at 256. The objective is to guarantee the good characteristics of the beam, direction (45° on the defect) and focalization on the outer surface, and to minimize grating-lobes.

4.2. Variables

The optimization concerns the design of the array (dimensions, arrangement and central frequency of the elements) the definition of the commuting aperture and the angular deviation defining the electronic delays. The variables, search intervals and accuracy are listed in Table 3.

4.3. Objective functions

The definition and evaluation of the objective functions are based on the computation by CIVA of the ultrasonic beams radiated by the array in the pipe and corresponding to the different

name	description	interval	acc.	unit
Design of the array				
l	length along $\vec{z}$	[1.0; 10.0]	0.5	mm
g	gap between elements	[0.1; 1.0]	0.1	mm
w	elements width	[0.2; 1.5]	0.1	mm
f	central frequency	[2.0; 10.0]	1.0	MHz
Electronic setting				
n	number of active commuting elements for the sequence	[3; 20]	1	–
θ	polar coordinate of the focal point on the outer surface	[45; 80]	1	degree

Table 3: Variables defined to optimize design and setting of an array for a pipe inspection.

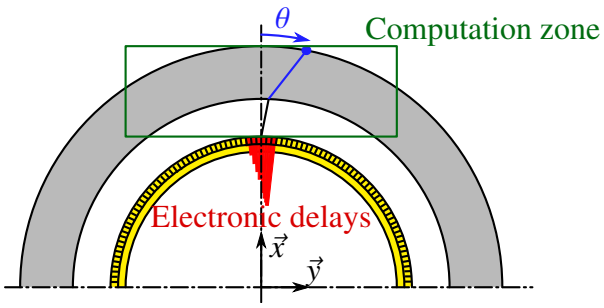

Figure 7: Sketch of the computation zone and array setting. The electronic delays applied to the rotating active aperture are plotted in red.

candidate solutions. More precisely, only the longitudinal component is calculated and the reflection on the outer surface is not taken into account. Computation parameters are sketched in figure 7. The computation zone is a rectangular grid of 161×61 points, distant of 0.2 mm along both axis. The steel is assumed to be isotropic with compressional wave velocity of 5650 m/s and negligible attenuation. The sound wave velocity in water is set at 1483 m/s in water. For each deviation angle θ and active aperture, the delay law applied to the array is computed by using the geometrical approach available in the CIVA package. Two objective functions are defined: (f_1) ensures a beam orientation as close as possible to 45° on the defect, and (f_2) maximizes the ratio between *useful* amplitude distribution inside the beam and *unwished* amplitude distribution outside the beam. The later mainly corresponds to the possible apparition of the grating lobe, which is to be avoided.

4.3.1. Beam orientation

The objective function f_1 is defined as the difference between 45° and the average orientation in the beam. This average orientation is evaluated on a set of pixel A_i of highest amplitudes.

$$f_1(X) = \frac{1}{N} \sum_{i=1}^{N} |45° - \alpha_i(X)| \qquad (14)$$

The orientation (angle α_i) assigned to one pixel A_i is issued

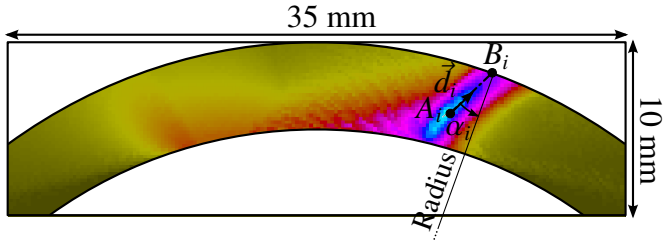

Figure 8: Definition of the parameters related to the objective function f_1, sketched on the map of the computed ultrasonic field corresponding to one candidate solution.

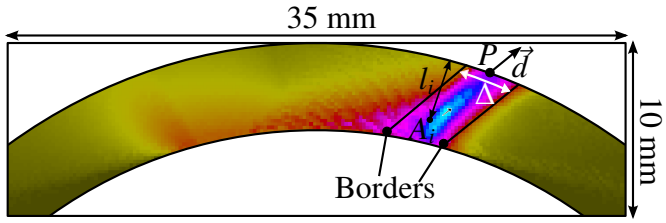

Figure 9: Definition of the parameters related to the objective function f_2, illustrated on the map of the computed ultrasonic field corresponding to one candidate solution.

from the local direction of propagation $\vec{d}_i$ output of the CIVA computation. α_i measures the angle between $\vec{d}_i$ and the normal of the outer surface at B_i (see figure 8).

4.3.2. Amplitude distribution

The spatial extension of the *useful* beam is defined for each candidate solution from the −6 dB width Δ and the direction of propagation $\vec{d}$ at the focal point F (see figure 9), which is an output of the field computation. The second objective function f_2, related to the amplitude distribution, is expressed as

$$f_2(X) = -\frac{\mathrm{Amp}^{\mathrm{in}}(X)}{\mathrm{Amp}^{\mathrm{out}}(X)} \tag{15}$$

where $\mathrm{Amp}^{\mathrm{out}}(X)$ is the average amplitude of all the pixels laying outside the *useful* beam:

$$\mathrm{Amp}^{\mathrm{out}}(X) = \langle \mathrm{Amp}_i^{\mathrm{out}} \rangle_{\mathrm{out}} \tag{16}$$

and $\mathrm{Amp}^{\mathrm{in}}(X)$ is the weighted average amplitude of all the pixels laying inside the *useful* beam:

$$\mathrm{Amp}^{\mathrm{in}}(X) = \langle w_i \mathrm{Amp}_i^{\mathrm{in}} \rangle_{\mathrm{in}} \tag{17}$$

The weights w_i are introduced in order to favor solutions with highest amplitude on the outer surface. w_i decreases with increasing distance to the outer surface l_i.

4.4. Constraints

One constraint function is introduced to guarantee that the number of elements of the array is lower or equal to 256. The expression of the constraint is:

$$g_1(X) = \frac{40\pi}{g(X) + w(X)} - 256 \leq 0 \tag{18}$$

name	variables						objectives	
	l	g	w	f	n	θ	f_1	f_2
sol 1	3.5	0.3	0.2	7	14	67	1°	11 dB
sol 2	4.5	0.3	0.2	8	16	63	4°	14 dB
sol 3	4	0.3	0.2	10	13	55	8°	16 dB

Table 4: Three interesting solutions from the Pareto front.

4.5. Results and discussion

The following values of tuning parameters have been used to perform the optimization: $Cr = 0.7$, $NP = 25$, and $k_{max} = 50$. 1530 candidate solutions have been generated among which 1422 were feasible. The evaluation of feasible solutions (implying a field computation) took 50h total running time on a standard PC. The Pareto optimal sets obtained at generations 0, 10, 20 and 50 are depicted in figure 10. At the 50th generation, the Pareto set is composed of 94 solutions corresponding to different compromises between f_1 and f_2. Along the front, f_1 and f_2 range respectively from 1° to 25° and from 11 dB to 19 dB. All the solutions have approximately the same values of array length ($l = 4 \pm 0.5$ mm), element width ($w = 0.2$ mm) and gap between elements ($g = 0.3$ mm). Unsurprisingly, these values correspond to the allowed maximum number of elements (256) and minimum width of the elements. The three parameters that mostly influence the compromise between the two objectives are the number of elements in the active aperture n, the frequency f, and the deviation angle θ. We give three representative solutions of the Pareto optimal front corresponding to reasonable compromises between the two objectives in the Table 4. The computed fields associated to these three solutions can be seen in figure 11. Solution 1 favors the beam orientation while solution 3 maximizes the focalization on the outer surface, the solution 2 being an intermediate solution. This example illustrates the interest of the algorithm which proposes in such multi-objective problem the best set of compromises among with the expert can select the final solution.

5. Conclusion

A method based on the use of a Differential Evolution algorithm, the Randomized Adaptive Differential Evolution (RADE) has been proposed to optimize array design and setting. The RADE has been adapted to meet the specificities of the addressed NDT applications and in particular to: i) manage the constraints on the variables in the problem: bounding domain and inequality constraints, ii) consider the finite precision of the variables, iii) solve multi-objective problems with the implementation of the concept of Pareto's optimal set of solutions. The algorithm has been implemented and connected to the ultrasonic simulation modules of the CIVA software platform. For one given solution (expressed as a set of parameters input into the simulation) the value of the objective function is inferred from the result of the corresponding simulation. The method has been evaluated on two realistic cases of application considered respectively as one mono-objective problem and one constrained multi-objective problem: the optimization of the

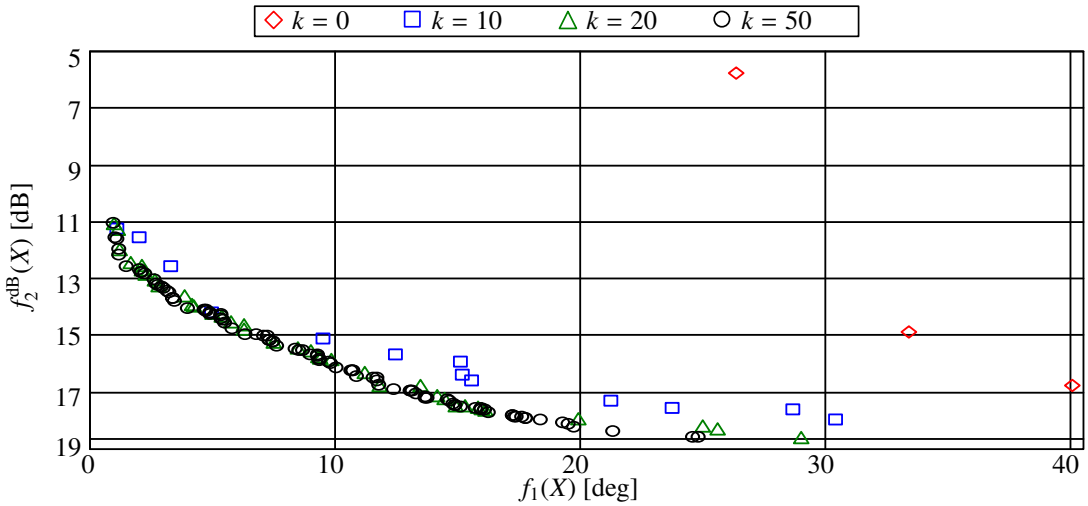

Figure 10: Pareto optimal sets obtained each ten generations during the optimization of the surrounded array, where $f_2^{dB}(X) = 20log(-f_2(X))$.

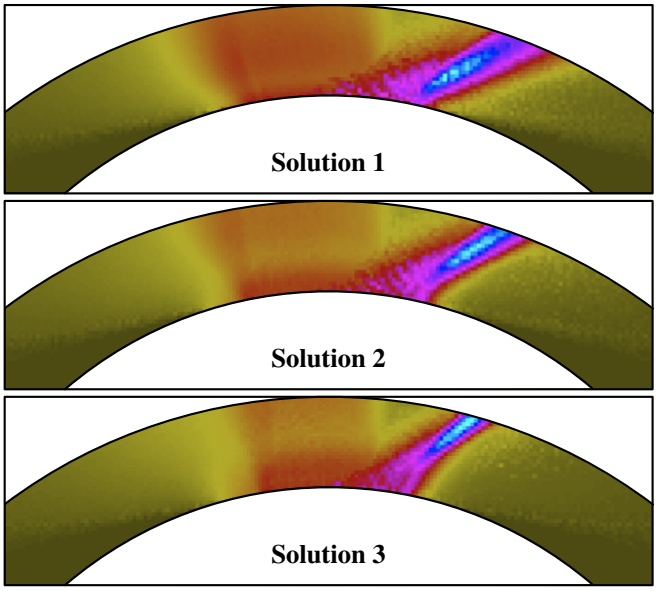

Figure 11: Maps of the computed ultrasonic field corresponding to the three solutions of Table 4.

positioning and electronic delays applied to a flexible array inspecting a nozzle and the optimization of the array design and electronic delays for a pipe inspection. The results confirmed the ability of the algorithm to converge towards the desired solution within a reasonable time. The comparison with *manual* reference solutions when available highlighted the equivalent or better quality of the solution provided by the algorithm.

Acknowledgement

This work has been supported by the Région Ile-de-France in the framework of the CAPVERS project.

References

[1] P. Rocca, M. Benedetti, M. Donelli, D. Franceschini, A. Massa, Evolutionary optimization as applied to inverse scattering problems, Inverse Problems 25 (12) (2009) 123003.

[2] S. Chen, S. Razzaqi, V. Lupien, An evolution strategy for improving the design of phased array transducers, in: IEEE Congress on Evolutionary Computation, 2006, pp. 2859-2863.

[3] V. Lupien, W. Hassan, P. Dumas, Improved titanium billet inspection sensitivity through optimized phased array design, part I: Design technique, modeling and simulation, in: D. O. Thompson, D. E. Chimenti (Eds.), Review of Progress in QNDE, Vol. 820, AIP, 2006, pp. 853-860.

[4] P. Yang, B. Chen, K.-R. Shi, A novel method to design sparse linear arrays for ultrasonic phased array, Ultrasonics 44 (2006) e717-e721.

[5] C. Gueudre, L. Le Marrec, J. Moysan, B. Chassignole, Direct model optimisation for data inversion. Application to ultrasonic characterisation of heterogeneous welds, NDT & E International 42 (2009) 47-55.

[6] M. Lu, M. Wan, F. Xu, X. Wang, X. Chang, Design and experiment of 256-element ultrasound phased array for noninvasive focused ultrasound surgery, Ultrasonics 44 (2006) e325-e330.

[7] A. Bréard, G. Perrusson, D. Lesselier, Hybrid differential evolution and retrieval of buried spheres in subsoil, Geoscience and Remote Sensing Letters, IEEE Geoscience Remote Sensing Letter (4) (2008) 788-792.

[8] A. Nobakhti, H. Wang, A simple self-adaptive differential evolution algorithm with application on the alstom gasifier, Applied Soft Computing 8 (1) (2008) 350-370.

[9] Details may be found at http://www-civa.cea.fr

[10] S. Mahaut, M. Darmon, S. Chatillon, F. Jenson, P. Calmon, Recent advances and current trends of ultrasonic modelling in CIVA, Insight - Non-Destructive Testing and Condition Monitoring 51 (2) (2009) 78-81.
[11] P. Benoist, C. Poidevin, O. Roy, Simulation and applications of NDT ultrasonic array techniques, Proc. Of the ASME Pressure Vessel and Piping conf. 2009, Vol5, (2010) 211-217
[12] K. V. Price, R. Storn, J. Lampinen, Differential Evolution: A Practical Approach To Global Optimization, Springer-Verlag Berlin and Heidelberg GmbH & Co. K, 2005.
[13] K. Zielinski, P. Weitkemper, R. Laur, R. L. K.-D. Kammeyer, K. Zielinski, Examination of stopping criteria for differential evolution based on a power allocation problem, in: 10th International Conference on Optimization of Electrical and Electronic Equipment, Brasov, Romania, 2006.
[14] V. Huang, A. Qin, P. Suganthan, Self-adaptive differential evolution algorithm for constrained real-parameter optimization, in: IEEE Congress on Evolutionary Computation, 2006, pp. 17-24.
[15] K. Deb, An efficient constraint handling method for genetic algorithms, Computer Methods in Applied Mechanics and Engineering, Vol. 186, 2000, pp. 311-338.
[16] T. S. Angell, A. Kirsch, Optimization Methods in Electromagnetic Radiation, Springer-Verlag New York, 2004.
[17] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: Nsga-II, IEEE Transactions on Evolutionary Computation 6 (2) (2002) 182-197.
[18] N. Gengembre, A. Lhémery, Pencil method in elastodynamics: application to ultrasonic field computation, Ultrasonics 38 (1-8) (2000) 495-499.