Genetic Algorithm Optimization for a Surgical Ultrasonic Transducer

Daniel Porto , Aurélien Bourquard and Yves Perriard
Integrated Actuators Laboratory (LAI)
École Polytechnique Fédérale de Lausanne (EPFL)
CH-1015 Lausanne, Switzerland
Emails: daniel.porto@epfl.ch, yves.perriard@epfl.ch

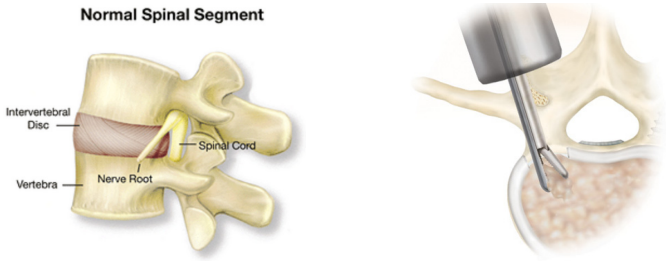

Fig. 1. Spinal Disc Segment and Disc Removal System

***Abstract*—An ultrasonic piezoelectric transducer to cut the human tissue or to remove the spinal disc is envisioned to improve efficiency and facilitate the surgeons work. Three genetic algorithms have been developed. The first one is based on conditional genetic operators, the second one is based on a specific crossover operator definition with a local search method, and the third is a combination of both the previous algorithms. The third algorithm is used to optimize the piezoelectric transducer. In less than 5000 simulations the optimizations give an amplitude of the cutting tip of about 4.8 microns.**

I. Introduction

Nowadays everyone knows someone who suffers from back pain. In most cases a surgical intervention has to be done and different manual methods are already used to cut the human tissue or to remove the spinal disc. Nevertheless, the surgeons are not always fully satisfied due to the tedious, time consuming and taxing of the existing devices. One interesting solution is the development of an ultrasonic piezoelectric transducer that could have an improved efficiency and facilitate the surgeons work.

II. Genetic Algorithms

Genetic algorithms (GA) repeatedly modify during successive generations a population of different sub-optimal solutions of a given problem. The selection process consists in keeping a number of individuals of the current generation, depending on some criteria, and produces children for the next generation. Over successive generations, the population evolves toward an optimal solution. The *fitness* function evaluates, for a given solution, its optimality degree; the evaluation of this function is essential to the selection process.

Interestingly enough that GA can be applied to particular optimization problems, in which some discontinuity or differentiability problems could appear.

A. Conditional genetic Operators

The working principle of conditional genetic operators [1] is to avoid using hard to find probabilistic parameters, and instead use conditioned operators by an objective measure of the population diversity, the *difference-degree* d_i.

A minimal *difference-degree* $D_s \in]0;1[$ is imposed at the beginning, that avoids a premature convergence and ensures some exploration of the domain search. The imposed diversity reduces progressively during the generations, that allows a more and more local search, however controlled.

The algorithm *upgr1* based on [1] is presented:

Step 1 Generate a population of N individuals and evaluate their *fitness*. Define a minimal D_s to impose.

Step 2 Select $\frac{N'}{2}$ pairs of parents ($N' \leq N$, N' even) with a selection method (as tournament). A same parent can be selected twice.

Step 3 Evaluate the *difference-degree* d_i between the two parents, for each pair i of parents.

Step 4 If $d_i < D_s$ for a given pair, the two parents suffer each one a mutation, and return to Step 3. Otherwise, continue to next step.

Step 5 Generate N' children from $\frac{N'}{2}$ pairs of parents by *crossover* (always applied). No more mutation operator is applied.

Step 6 Evaluate the *fitness* of the children.

Step 7 Reverse replacement: the best $R \cdot 100\%$ children replace the worst individuals of the current population.

Step 8 Relax the diversity constraint, with $D_s = \mu D_s$ and the cooling ratio $\mu \in]0;1[$. Return to Step 2.

One difficulty resides in the fact that [1] uses a binary representation for the individuals, and then the mutation, the *crossover* operators and *difference-degree* between two individuals are specifically defined for that representation.

Considering that we decide to keep the individuals representation with real values, the operators and the *difference-degree* measure defined in [1] are modified; and thus are not the same as defined in [1].

The *difference-degree* $d_i \in]0;1[$ between two individuals $\boldsymbol{X_1}$ and $\boldsymbol{X_2}$ (p the number of alleles, parameters or variables

of an individual, and $\boldsymbol{a}$ and $\boldsymbol{b}$ the lower and upper boundaries of $\boldsymbol{X}$) is:

$$d_i = \frac{1}{p} \sum_{i=1}^{p} \frac{X_{1,i} - X_{2,i}}{a_i - b_i} \tag{1}$$

The *crossover* operator used (as in [1]) is the *two-point crossover* consisting in the reversal of the alleles of the two individuals (with real values and not a single bit) between randomly chosen positions.

The mutation operator is chosen nearby of its binary counterpart (the *simple random*); thus, the following random value X_i' is added to one of the real variables i of the considered individual:

$$X_i' = X_i + \tau \cdot (a_i - b_i) \cdot 2^{-\lfloor r \cdot z \rfloor} \tag{2}$$

with τ that can equiprobable be ± 1, r a random value in $[0, 1[$ and z the accuracy (in power of two) that each real value is supposed to be (binary representation analogy).

B. Crossover and Local Search

The improvements (based on [2], with real values representation) done here concern the definition of a particular *crossover* operator, and the hybridization with a local search method. According to [2], these improvements avoid a premature convergence and, on the other hand, avoid a convergence too slow.

The *upgr2* algorithm, based on algorithm $GA(c_{r2}, l)$ in [2] (*crossover* rule n°2 with local search), is the following:

Step 1 Generate a population of N individuals and evaluate their *fitness*.

Step 2 Select $\frac{N'}{2}$ triplets of parents ($N' \leq N$, N' even) with a tournament selection with 2 players. A same parent can be selected twice in the same triplet.

Step 3 Generate N' children (N' even) from triplets of parents with help of a *crossover* operator defined below (always applied), each triplet of parents generates two children. Each variable of N' children has a probability P_m to be affected by the mutation operator.

Step 4 Evaluate the *fitness* of the children.

Step 5 Reverse replacement: the best $R \cdot 100\%$ children replace the worst individuals of the current population.

Step 6 Local search (optional). Return to Step 2.

The *crossover* operator is defined, for each variable i, with $\boldsymbol{X_1}$, $\boldsymbol{X_2}$ and $\boldsymbol{X_3}$ the three parents, $\boldsymbol{X_3}$ is the parent with the worst *fitness*, and $\boldsymbol{Y_1}$ and $\boldsymbol{Y_2}$ the two children:

$$Y_{1,i} = \frac{X_{1,i} + X_{2,i}}{2} + \sigma_i \left(\frac{X_{1,i} + X_{2,i}}{2} - X_{3,i} \right) \tag{3}$$

$$Y_{2,i} = X_{1,i} + \Phi_i (X_{2,i} - X_{3,i}) \tag{4}$$

with σ_i a uniform random number in $[0; 1]$ and Φ_i a uniform random number in $[0.4; 1]$ for each i.

The mutation operator, for a given variable i, is defined at generation k as:

$$X_i' = \begin{cases} X_i + \Delta(k, b_i - x_i) & \text{if } \tau = 0 \\ X_i - \Delta(k, x_i - a_i) & \text{if } \tau = 1 \end{cases} \tag{5}$$

where τ is a random bit that can take either 0 or 1, and

$$\Delta(k, y) = y \left(1 - r^{\left(1 - \frac{k}{T}\right)^b} \right) \tag{6}$$

where r is a random number in $[0, 1]$, T is the maximal number of individual generations and b is a specified parameter by the user that determines the non-uniformity degree in the search domain (typically, $b = 5$).

Local search works as follows: select an individual $\boldsymbol{Y}$, not the best individual $\boldsymbol{X_l}$, in the current population. Generate a new testing individual $\boldsymbol{X_n}$. If the new *fitness* $F_n <$ *fitness* F_h of the worst individual $\boldsymbol{X_h}$, then $\boldsymbol{X_h}$ and F_h are replaced respectively by $\boldsymbol{X_n}$ and F_n. Repeat the process l_r (local searches specified by user) times.

Each component of $\boldsymbol{X_n}$ is calculated as follows:

$$X_{n,i} = (1 - \gamma_i) X_{l,i} - \gamma_i Y_i \tag{7}$$

$\gamma_i \in [-0.5, 1.5]$ uniform random numbers for each i.

C. Combination of Both Improvements

Given that the two improvements are of concern of each separate part of the GA (the conditional operators application rules, and the crossover operator with an added local search strategy), the combination of the two improvements gives a more efficient algorithm than the two improvements taken separately. Moreover, that algorithm generalizes the selection method in the sense that a method among others (tournament, wheel, etc.) can be specified by the user.

Thus, the algorithm *upg12* arises in the following manner:

Step 1 Generate a population of N individuals and evaluate the *fitness* of each individual.

Step 2 Select $\frac{N'}{2}$ triplets of parents ($N' \leq N$, N' even) with a chosen selection method (eg. tournament). A same parent can be selected twice in the same triplet.

Step 3 Evaluate the triplets $d_{i,mean}$, defined as the mean of the *difference-degrees* $d_{i,12}$, $d_{i,23}$ and $d_{i,31}$ of the parents taken two by two of each triplet.

Step 4 If $d_{i,mean} < D_{s,mean}$ for a given triplet , the three parents suffer each one a mutation following Eq. (2) (does not depend on any probabilistic parameter to be specified) and back to Step 3. Else, next step.

Step 5 Generate N' children (N' even) from the triplets of parents with the *crossover* operator (always applied) defined by Eq. (3) & (4), each triplet of parents generates two children. No more mutation operator is any applied. Let us note that the crossover operator, that needs the parents *fitness* as inputs, uses here the fitness of the parents evaluated before any mutation at Step 4; indeed, this approximation is useful to avoid a very large number of objective function evaluations and does not spoil the algorithm results in a significant manner.

Step 6 Evaluate the children *fitness*.

Step 7 Reverse replacement: the $R \cdot 100\%$ best children replace the worst individuals of the current population.

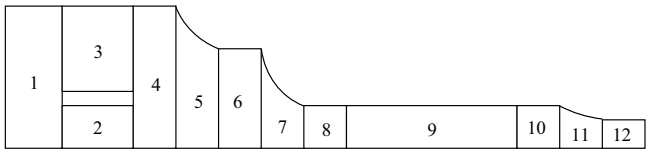

Fig. 2. 2D Axi-symmetric Ultrasonic Transducer Model

Step 8 Local search (optional). Return to Step 2.

In this algorithm, the imposed *difference-degree* $D_{s,mean}$ depends on other parameters than a constant factor multiplication μ. Indeed, a correspondence between $D_{s,mean}$ and the mutation operator defined at Eq. (5) is set up. More precisely, the imposed mean *difference-degree* $d_{i,mean}$ in triplet of parents has to be of the same order as a variable (or allele) mutation of one of the three parents according to Eq. (5). Thus $D_{s,mean}$ is given by:

$$D_{s,mean} = \frac{1}{3p} \cdot \left(1 - 0.5^{\left(1-\frac{k}{T}\right)^b}\right),$$

with p the number of variables of an individual.

The present solution, although it seems to have a superfluous complexity degree, reveals to be more efficient than other possibilities considered with the *difference-degree* D_s evolving with the cooling rate μ.

III. Transducer Optimization

The GA optimization routines are developed with MATLAB software combined with finite element (FE) simulations in ANSYS used to maximize the vibrational displacement amplitude of the cutting tip at the resonance frequency f_r and the vibrational displacement along the actuator.

A. Transducer Model

An ultrasonic transducer model (Figure 2) composed of a piezoelectric stack (3), eight mechanical transmission parts (1, 2, 4, 6, 8, 9, 10, 12) and three exponential horns (5, 7, 11) for the movement amplification is defined.

B. Optimization Parameters

The transducer has eleven geometric parameters to be optimized, that are diameters OD_i or lengths L_i , cf. values in square brackets in Table I.

The materials for that case are chosen but could also be optimized. The piezoelectric stack is composed by eight discs of PZ-28 material. All other metal parts are made of titanium TI-6Al-4V-STA. The resonance frequency f_r is fixed at 22.5 kHz and the applied voltage V_a to the piezoelectric stack is 100 V.

IV. Results and Discussion

In less than 5000 function evaluations (ANSYS simulations), at the resonant frequency f_r of 22.5 kHz, the GA optimizations give an amplitude of the cutting tip of about 4.8 μm and the total length of the transducer is about 340 mm. The used algorithm is *upg12*.

TABLE I
Parameters Dimensions to be Optimized

i	L_i [mm]	OD_i [mm]	ID_i [mm]	Material
1	[5; 50]	15	0	TI-6Al-4V-STA
2	L_3	6	ID_1	TI-6Al-4V-STA
3	24.9	OD_1	7	PZ-28
4	[2; 50]	OD_3	ID_2	TI-6Al-4V-STA
5	[5; 50]	-	ID_4	TI-6Al-4V-STA
6	[5; 50]	$[OD_8; OD_4]$	ID_5	TI-6Al-4V-STA
7	[5; 50]	-	ID_6	TI-6Al-4V-STA
8	[5; 50]	6.35	ID_7	TI-6Al-4V-STA
9	[5; 200]	OD_8	ID_8	TI-6Al-4V-STA
10	[5; 50]	OD_9	ID_9	TI-6Al-4V-STA
11	[5; 50]	-	ID_{10}	TI-6Al-4V-STA
12	[5; 40]	3.4	ID_{11}	TI-6Al-4V-STA
f_r = 25 kHz ; V_a = 100 V				

Fig. 3. 3D View of the Optimized Transducer

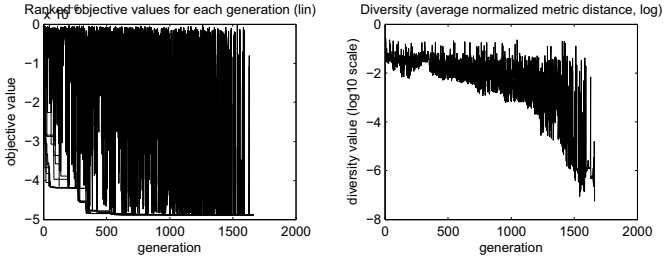

Fig. 4. Diversity and Objective Values

A 3D view of the optimized transducer is shown in Figure 3 with the piezoelectric stack represented in green color.

The optimized transducer parameters are summarized in Table II.

TABLE II
Optimized Parameters Results

L_1 = 45.89 [mm]	L_4 = 9.49 [mm]	L_5 = 13.20 [mm]
L_6 = 28.79 [mm]	OD_6 = 7.55 [mm]	L_7 = 38.86 [mm]
L_8 = 31.23 [mm]	L_9 = 43.05 [mm]	L_{10} = 35.20 [mm]
L_{11} = 30.94 [mm]	L_{12} = 39.28 [mm]	

The objective function, corresponding to the vibrational amplitude displacement calculated in ANSYS, evolution and the diversity through the generations are shown in Figure 4.

The number of generations (1663) depends on the specified or desired parameters for each algorithm (number of parents, children, local search), and is different from the number of function evaluations (5000).

At the resonant frequency f_r of 22.5 kHz, the harmonic displacement in the transducer nodes in the y-axis direction is shown in Figure 5. The amplitude of the cutting tip is about 4.8 μm.

The transducer has three nodes where there is no vibrational amplitude and that are very useful to hold the transducer in

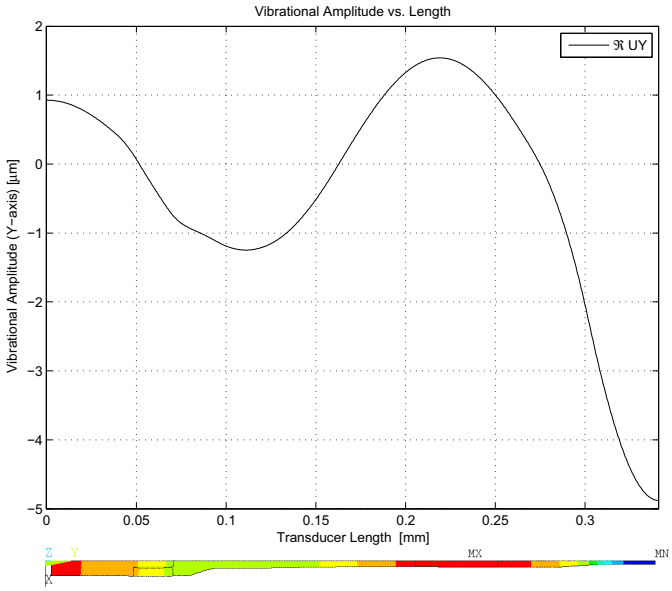

Fig. 5. Real Amplitude Displacement in Transducer Nodes

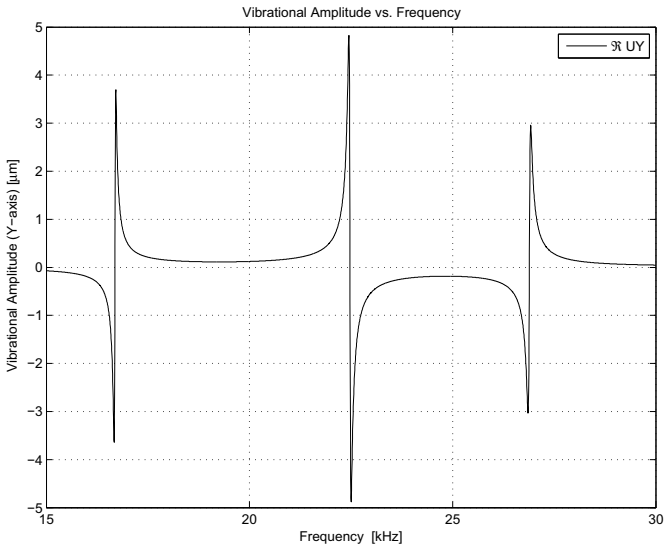

Fig. 6. Harmonic Frequency Response

ones hands. The position of nodes in the transducer can only be seen in Figure 5.

It is also relevant to verify the resonant modes of the transducer near the resonant frequency to avoid disturbing modes. The frequency analysis between 15 kHz and 30 kHz is shown in Figure 6.

V. Conclusion

In this paper, three genetic algorithms have been presented. An ultrasonic transducer model has been defined and optimized with GA and FE simulations.

In a previous work [3], an optimization of a piezoelectric transducer has been done with a pseudo-gradient method based on an analytical developed model. When comparing the vibration amplitude with the two methods (pseudo-gradient and genetic algorithms), the results are almost the same. The advantage of the GA resides in the low number of simulations needed to obtain good results. Which makes optimizations of full FE models truly envisioned.

References

[1] R.-L. Wang and K. Okazaki, "An improved genetic algorithm with conditional genetic operators and its application to set-covering problem," *Soft Comput.*, vol. 11, pp. 687–694, 2007.

[2] M. Ali and P. Kaelo, "Integrated crossover rules in real coded genetic algorithms," *Eur. J. Oper. Res.*, vol. 176, pp. 60–76, 2007.

[3] J. Murphy, D. Porto, and Y. Perriard, "Ultrasonic transducer model for optimization of a spinal tissue ablation system," in *Industry Applications Conference, 2006. 41st IAS Annual Meeting. Conference Record of the 2006 IEEE*, vol. 1, Oct. 2006, pp. 379–384.

[4] A. Lemonge and H. Barbosa, "An adaptive penalty scheme for genetic algorithms in structural optimization," *Int. J. Numer. Methods Eng.*, vol. 59, pp. 703–736, 2004.

[5] H. Maaranen, K. Miettinen, and A. Penttinen, "On initial populations of a genetic algorithm for continuous optimization problems," *J. of Global Optim.*, vol. 37, pp. 405–436, 2007.

[6] Matlab, *Genetic Algorithm and Direct Search Toolbox 2, User's Guide*, 2007, www.mathworks.com.

[7] C. Moon, J. Kim, G. Choi, and Y. Seo, "An efficient genetic algorithm for the traveling salesman problem with precedence constraints," *Eur. J. Oper. Res.*, vol. 140, pp. 606–617, 2002.

[8] T. Schaffter, "Optimisation d'un moteur synchrone à l'aide d'algorithmes génétiques," 2007, projet de Semestre, EPFL.

[9] L. Snyder and M. Daskin, "A random-key genetic algorithm for the generalized traveling salesman problem," *Eur. J. Oper. Res.*, vol. 174, pp. 38–53, 2006.

[10] *ANSYS Academic Research 11 version, ANSYS Inc.*, 2007.