Automatic Fitting of Cochlear Implants
with Evolutionary Algorithms

C. Bourgeois-République
Université de Bourgogne
LERSIA — BP 47870
2178 Dijon cedex, France
Bourgeois.Republique@Free.Fr

J.-J. Chabrier
Université de Bourgogne
LERSIA — BP 47870
21078 Dijon cedex, France
JJChab@u-bourgogne.fr

P. Collet
Université du Littoral
LIL — BP 719
62228 Calais cedex, France
Pierre.Collet@Univ-Littoral.Fr

ABSTRACT

This paper presents an optimisation algorithm designed to perform in-situ automatic fitting of cochlear implants.

All patients are different, which means that cochlear parametrisation is a difficult and long task, with results ranging from perfect blind speech recognition to patients who cannot make anything out of their implant and just turn it off.

The proposed method combines evolutionary algorithms and medical expertise to achieve autonomous interactive fitting through a Personal Digital Assistant (PDA).

Categories and Subject Descriptors
G.1.6 [Numerical Analysis]: Optimization;
G.3 [Probability and Statistics]: Stochastic processes

Keywords
Interactive evolutionary optimization, cochlear implants.

1. INTRODUCTION

Cochlear implants are electronic devices which attempt to give back auditory sensations to otherwise incurable deaf patients whose auditive nerve is however still functional.

The input is an acoustic signal that is obtained from an external microphone. The signal is processed through a microprocessor which turns it into electric impulses that are sent into an array of electrode which is inserted into the cochlea, in order to artificially stimulate the auditive nerve.

Many deaf people can benefit from cochlear implants. However, the history of the patient as well as the cause of deafness has a lot of influence on the good parametrisation of the cochlear implant. Many studies show that the brain area responsible for

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. SAC'04 March 14-17, 2004, Nicosia, Cyprus Copyright 2004 ACM 1-58113-812-1/03/04 ...$5.00.

stimulated, suggesting that an important reconfiguration of their auditory neural network has taken place.

All in all, it is very difficult to draw conclusions and develop a deterministic fitting methodology when patients with a similar background react very differently.

In 2003, around 50000 deaf people have been implanted with such devices around the world. Efficiency is quite variable, ranging from totally deaf patients that have fully re-covered their audition and are capable to follow telephone conversations and enjoy music, to others who hear strange sounds they can’t benefit from, up to a point where they prefer to switch off the implant. Success depends on many different factors (age of implantation, number of years of deafness before implantation, etc) but also implant parametrisation. The aim of this work is to develop an interactive software capable of helping both patients and practitioner in the parametrisation of the cochlear implant. Section 2 describes the current method used in cochlear implant fitting and the problems associated with it. Section 3 presents possible optimisation techniques while section 4 describes stochastic optimisation and evolutionary algorithms. Section 5 describes an implementation with the EASEA language. Finally, the new fitting procedure is presented in section 6 which makes use of a PDA. Section 7 presents conclusions and perspectives.

2. CURRENT WAY OF FITTING COCHLEAR IMPLANTS

The aim of cochlear implant fitting is to optimise an artificial electric stimulation of the auditory nerve in order to improve communication. Speech characteristics are put forward so as to improve speech recognition [5].

Right now, fitting takes place at hospital, in a quiet room. The practitioner connects the patient’s cochlear implant to a computer to modify the parameters depending on the sensations of the patient. Many factors influence the tuning, among which the patient’s general health state and age. The number of tested electrodes and the length of the fitting session depend very much on how long the patient is receptive and able to collaborate.

Depending on the medical team, fitting sessions are scheduled from one to seven times per year. One session takes between 60 and 90 minutes, and can be described as follows:

1. The patient tells the practitioner about the previous parametrisation of his implant.
2. The implant is linked to a computer and minimum and maximum thresholds are checked for each electrode.
3. Psychophysical tests are performed and implant parameters are modified according to the practitioner’s expertise.
4. The patient then tests or comments the new fitting. This procedure is repeated until the patient is satisfied or exhausted. Fitting a cochlear implant is a slow process. Steps 1 and 2 take about 30 minutes. Then, the time taken by steps 3 and 4 depends on the patience and collaboration of the patient.

Two to five parameters settings are tested by the patient in around 90 minutes. During step 3, the practitioner uses his expertise to submit new settings to the patient. Choices are deterministic, and only bear on a few parameters. No drastic changes are performed and the different settings submitted to the patient are only variations on the same original parameters, simply because the practitioner is a human being who would not be able to draw conclusions of tens or hundreds of radically different settings.

Unfortunately, implant fitting is nothing but a deterministic process, as all patients are different. Their auditory neurons react in a way that is difficult to predict, as they have not been stimulated equally if the patient is pre- or post-lingually deaf, or whether deafness occurred 2 or 20 years before implantation or due to meningitis or from traumatic causes.

The great number of parameters, the disparity of patients, the difficulty to communicate with the patient, the time taken by the process of modifying a single parameter all are obstacles to methods based on medical expertise, which explain why it is difficult for experts to rapidly find an optimum fitting for a given patient.

3. PROPOSED SOLUTION

The fitting method presented in this paper is based on a stochastic optimisation technique that also takes into account the expertise of the practitioner. The optimization software takes place into a PDA that the patient can carry with him. This feature is very important, as it is nearly impossible to realistically simulate such environments as a restaurant, a cinema or a train station in the practitioner’s small room.

The PDA can connect to the cochlear implant, and the patient can start a fitting session with the PDA wherever and whenever he feels like it.

3.1 Parameters classification

Parameters fall into two categories: determinant parameters and refinement parameters.

3.1.1 Determinant parameters

Determinant parameters are those whose modification can radically change the auditory perception of the patient.

- On MXM implants[7], with which this work is realised, three parameters are determinant:
  - activation or deactivation of an electrode,
  - offset of an electrode,
  - bandwidth covered by an electrode.

- Each of these three parameters apply to all of the fifteen electrodes of the implant. Offset and bandwidth are integers within [549;7263] and [122;7324] respectively, while the activation or deactivation of an electrode is a boolean.

3.1.2 Refinement parameters

The following five parameters allow to finely tune the rough settings obtained with the determinant parameters:
- Microphone sensitivity,
- Gain,
- Volume,
- Adaptation Energy,
- Intensity.

3.2 Human-based stochastic optimisation

Determinant parameters are difficult to optimise, as the practitioner’s expertise is not of much help: depth of electrodes insertion in the cochlea, deficient auditory neurons, protein form auditory neural network of the patient, etc. make it very difficult for the practitioner to find deterministically an optimal fitting for a particular patient.

This somewhat tosses this problem into an NP-complete class of complexity, for which the only known way of finding the optimal fitting is to explore exhaustively all the possible values of the parameters.

Since a fitting procedure that would need thousands of tests is impossible to conduct, the human practitioner practically tests only a few combinations based on his expertise, which he later optimises by tuning deterministically refinement parameters. This turns this procedure into a stochastic optimisation, for which humans are not well prepared: Stochastic optimisation relies on the quality of the search space exploration and on the ability to exploit all the data that is produced by the tests. Humans, on the contrary are known to be bad at producing random sequences — meaning that it is very likely that the fitting procedure is biased by the practitioner — and at drawing clear conclusions if too many variables are involved: the practitioner is simply overwhelmed with data.

3.3 Computer-based stochastic optimisation

Such stochastic optimisation is a domain where computers are efficient, since they are good at producing random sequences and at extracting significant features from a large volume of data.

Moreover, the expertise of the practitioner can be used, provided that the ergodicity of the algorithm is maintained (i.e.: the guarantee that all points of the search space are attainable). Such optimisation methods are used for complex problems where all else fails (optimisation of non-derivable functions).

4. EVOLUTIONARY ALGORITHMS

Among the numerous stochastic methods that are available, evolutionary algorithms seem well suited for this specific problem. They rely on the evolution of a population of potential solutions which try to maximise a fitness function that can be seen as an evaluation of the solution[3].

Evolutionary Algorithms use vocabulary borrowed to genetics and to Darwin’s theory on natural selection. Potential solutions are called “individuals,” that are coded by their chromosome, “ itself made of the different parameters of the solution which are called “genes.” A set of individuals — called a population — is evolved, following steps found in natural evolution, i.e. reproduction
4.1 Description of an Evolutionary Algorithm

Before the algorithm can be executed, one needs to define the genome of an individual. In the present case, the three determinant parameters of the 15 electrodes are those that will be evolved by the evolutionary algorithm. Therefore, the genome consists in 45 integer values, i.e. 45 genes.

A typical evolutionary algorithm follows:

1. Initialisation of the population — i.e. creation of n individuals with random values within defined limits for each of the 45 genes of the individuals.
2. Evaluation of the initial population. This evaluated population is called the parent population. In our case evaluation is done by the patient, who rates individuals depending on the quality of the fitting.
3. In order to create children, reproduction takes place among some parents, chosen with a bias towards the best individuals.

Two genetic operators[2] implement reproduction:

- Recombination or crossover aims at creating new individuals out of the genes of the parents. Genes of either one parent or the other are selected to compose a new individual. Recombination allows to explore a basin of attraction made of the pool of genes of the parents, in what can be considered as a local search.
- A Mutation operator is then applied on the created children, in order to introduce new gene values. With recombination only, new values cannot appear for a specific gene, because values are selected among the parents. The algorithm would be limited to the pool of genes contained by all initial parents.

The aim of the mutation operator is to guarantee ergodicity with the aim of finding gene values that are better than those contained in the original gene pool.

The reproduction step is repeated until a specified number of children have been created, to constitute the offspring population.
4. The offspring population is then evaluated by the fitness function (the patient in this implementation).
5. A selection favouring the best individuals is operated among the parents and offspring populations, in order to reduce the total number of individuals to the initial population size. This reduced population will be the parent population of the next generation.
6. The algorithm then loops back to step 3 unless a stopping criterion has been met (the patient is satisfied or has had enough) in which case, the best individual of the population is the result of the algorithm.

4.2 Why Evolutionary Algorithms rather than other stochastic optimisation techniques?

Evolutionary algorithms are known to widely explore the search space, thanks to the fact that the population is made of several individuals. Other techniques, such as simulated annealing, are equivalent to optimising only one individual, meaning that only a small part of the search space is covered.

Therefore, evolutionary algorithms are less easily trapped into local optima (cf. Fig. 1), which is a very interesting feature in the problem of optimising cochlear implants fitting. What is sought with this automatic optimisation algorithm is to avoid the main obvious local optimum in the center of the figure that would not allow the patient to reach the necessary level to understand blind speech.

Another advantage is that evolutionary algorithms do not need a formal mathematical representation of the problem to optimise: they only need a fitness value for a proposed individual, which can be given interactively by the patient.

5. IMPLEMENTATION WITH THE EASEA SPECIFICATION LANGUAGE

It is unfortunately not easy to write an evolutionary algorithm. The algorithm needs to manipulate populations of individuals, with refined selection and reduction algorithms that would take a lot of time to write and debug. It was therefore decided to use EASEA (acronym of EAsy Specification of Evolutionary Algorithms), a high level language dedicated to evolutionary algorithms[1] that is written in standard C++ (necessary for the PDA implementation).

5.1 Genome structure

An important section of an EASEA program is the genome structure that contains the parameters to be optimised.

Even though only certain parameters will be optimised, all parameters are inserted into the genome, to keep track of the refinement parameters tuned by the practitioner. An electrode can be defined by the following structure:

```c++
ElectrodeStruct {
    bool bActive; // electrode activation
    int nMin; // minimum threshold
    int nMax; // maximum threshold
    int nOffset; // Frequency offset
    int nBandwidth; // Bandwidth
}
```

The complete genome contains 15 electrodes (as MXM cochlear implants use up to 15 electrodes) plus the rest of the parameters that do not depend on electrodes. An individual of the evolutionary algorithm is an instance of this structure:
5.2 Genome initialisation

An initialisation function is used to create the individuals of the first generation:

```cpp
GenomeClass::Initialiser {
    int CurrentOffset=549;
    for (int i=0;i<Nb_Electrodes;i++) {
        Electrode[i].bActive=tosscoin(0.8);
        Electrode[i].nOffset=CurrentOffset;
        Electrode[i].nBandwidth=random(0,log(i)*1000);
        CurrentOffset+=Electrode[i].nBandwidth;
    }
}
```

The tosscoin(r) function returns randomly 1 or 0, with a bias of r. If r=0.8, there are 80% chances that the result is true. This initialisation function creates different individuals, with stochastic parameters that benefit from the practitioner’s expertise: the bandwidth follows a logarithmic law.

5.3 Recombination

The crossover function creates two children from two parents. In fact, the function is given four individuals as parameters: two parents and two clones of the parents (child1 and child2). The crossover function must modify in order to turn them into real children. A standard monopoint crossover is used below for the sake of simplicity, although the real one contains many subtleties that implement some of the practitioner’s expertise.

```cpp
GenomeClass::crossover {
    int cross = random(0,Nb_Electrodes - 1);
    for (int i=0; i<cross; i++) {
        swap(child1.Electrode[i].bActive, child2.Electrode[i].bActive);
        swap(child1.Electrode[i].nOffset, child2.Electrode[i].nOffset);
        swap(child1.Electrode[i].nBandwidth, child2.Electrode[i].nBandwidth);
    }
    smoothen(&child1, cross);
    smoothen(&child2, cross);
}
```

The swap() function is written in the User Functions section of the EASEA source file, as well as the important smoothen() function, that takes care of possible gaps or overlaps that will appear at the level of the crossover point. The parameters of the electrodes around the gap will be slightly changed. This can be considered as some sort of mutation, although a quite minimal one.

5.4 Mutation

Mutation is the genetic operator that ensures ergodicity. The mutation function takes one individual, and modifies one parameter at random.

If the modified parameter is the bActive parameter of an electrode, the offset and bandwidth of the previous and next active electrodes are modified by the smoothen() function so as to take into account the fact that an electrode has appeared or disappeared:

```cpp
GenomeClass::mutation {
    int ElectrActivatMutation=random(0,Nb_Electrodes-1);
    if (tosscoin(.01)){
        flip(Electrode[ElectrActivatMutation].bActivation);
        smoothen(&this, ElectrActivatMutation); return;
    }
    for (int i=1; i<Nb_Electrodes; i++) {
        if (tosscoin(1.0/15.0)) {
            mutateOffsetAndBandwidth(&this, i);
            smoothen(&this, i);return;
        }
    }
}
```

The mutateOffsetAndBandwidth() function modifies both offset and bandwidth of the chosen electrode. It is called with a probability of 1/15, so that in average, one electrode is always mutated.

5.5 Evaluation

Evaluation of the individual is done by the patient, thanks to two inputs: a quantitative input (a mark ranging from 0 to 10) and a qualitative input (checkbox among 8 possibilities qualifying the synthetic sound: echo, white noise, hollow, far away, screechy, resonating, sharp, sibilant). These words were chosen because they are commonly used by implanted people to describe the quality of the sound they hear.

This information will be used soon to help choosing among individuals for reproduction: the idea is that the selector can help avoiding two “screechy” individuals.

```cpp
GenomeClass::evaluation {
    int Score=getScore(); // Calls the PDA’s GUI
    SaveIndividual(&this, score);
    return Score;
}
```

The genotype of all individuals and their evaluation is stored in a database implemented in ACCESS, because this software is available on the computers of the hospital.
The database is used for statistics, based on My-SQL requests, to extract information that is of interest to expert practitioners. Exploitation of the data will also help adjusting the parameters of the evolutionary algorithm.

5.6 EASEA default execution parameters

Here are the default parameters used for the evolution:

\begin{verbatim}
\text{Default run parameters :}
\text{Number of generations} : 1000
\text{Mutation probability} : 0.1
\text{Crossover probability} : 1
\text{Population size} : 10
\text{Selection operator} : Tournament
\text{Offspring population size} : 5
\text{Replacement strategy} : Plus
\text{Discarding operator} : Tournament
\text{Evaluator goal} : Maximise
\text{Elitism} : On
\end{verbatim}

10 different parameter sets are initially created, and five new individuals are coined at each generation. The crossover function is called every time, while mutation occurs only every ten children.

6. NEW FITTING PROCEDURE

Thanks to the presented software, the new fitting procedure is the following for a newly implanted patient (cf. fig. 3):

1. The first fitting session takes place with the practitioner, who determines with the patient the minimum and maximum stimulation values for each electrode.
2. The PDA is connected to the cochlear implant and the practitioner explains how the fitting software works.
3. The practitioner can then leave the patient alone with the PDA until some interesting results are obtained.
4. The practitioner downloads the results from the PDA and performs a fine tuning of the different parameters.

The advantages of this fitting procedure are numerous:

- Evolutionary algorithms are less likely to get trapped in a local optimum than a human practitioner.
- Patients can leave with the PDA and bring it in noisy environments that are very difficult to reproduce.

Using a PDA[4] is a great asset as it gives the patient the possibility to tune the implant in his everyday environment.

Communication between the PDA and the practitioner’s computer is done with the bluetooth[6] or wifi[8] protocol. Both methods can be used because the data that needs to be transferred is very small.

7. CONCLUSION AND PERSPECTIVES

Output of the evolutionary algorithm is currently extensively tested (bugs are not allowed in this domain, as stimulating an electrode with a too high current may damage remaining neurons and lead to a deafness that would be irrecoverable) while tests on willing patients should begin within a couple of weeks.

The proposed method being at least ten times faster than the current one, many more settings can be tested, therefore maximising the possibility to find the basin of attraction of the global optimum, or at least of a local optimum that would allow the patient to communicate and understand blind speech.

The setup presented in this paper is still a prototype, and many things can be done to optimise the software once the data collected in the database can be exploited. The MXM company is very helpful and interested by the results, as there is right now no other automatic fitting system on the market.

8. REFERENCES