14

Multi-Objective Genetic Algorithm to Automatically Estimating the Input Parameters of Formant-Based Speech Synthesizers

Fabíola Araújo, Jonathas Trindade, José Borges,
Aldebaro Klautau and Igor Couto
Federal University of Pará (UFPA)
Signal Processing Laboratory (LaPS)
Belém – PA
Brazil

1. Introduction

The Klatt synthesizer is considered one of the most important formant synthesis. Therefore, this chapter addresses the problem of automatic estimation of Klatt's synthesizer parameters in order to perform the imitation of voice (*utterance copy*), that is finding the parameters that causes the synthesizer to generate a voice that sounds close enough to the natural voice, so that the human ear does not notice the difference. Preliminary experimental results of a framework based on evolutionary computing, more specifically, in a kind of genetic algorithm (GA) called Multi-Objective Genetic Algorithms (MOGA), are presented. The task can be cast as a hard inverse problem, because it is not a simple task to extract the desired parameters automatically (Ding et al., 1997). Because of that, in spite of recent efforts (Breidegard & Balkenius, 2003; Heid & Hawkins, 1998), most studies using parametric synthesizers adopt a relatively time-consuming process (Klatt & Klatt, 1990) for utterance copy and end up using short speech segments (words or short sentences). GA was chosen to peform this task because they are known for their simplicity and elegance as robust search algorithms, as well as for their ability to find high-quality solutions quickly for difficult high-dimensional problems where traditional optimization methods may fail.

This chapter presents the application of GA to speech synthesis to solve the process of *utterance copy* (Borges et al., 2008). With this framework, we use several objective (fitness) functions and three possible ways of operating: *Interframe*, *Intraframe* and/or *knowledge-based* architectures with adaptive control of probabilities distribution and stopping criteria according to the convergence and number of generations. We also intend to fill a gap on the number of research efforts on developing automatic tools for dealing with formant synthesizers and help researchers to compare the performance of their solutions. The possibility of automatic analyzing speech corpora is very important to increase the knowledge about phonetic and phonological aspects of specific dialects, endangered language, spontaneous speech, etc. The next paragraphs provide a brief overview of the Klatt's speech synthesizer, the optimization problem and the approach using MOGA to solve this.

2. Speech synthesis

The voice synthesis consists on producing the human voice artificially, using the automatic generation of voice signal. Aspects as the naturalness or the intelligibility are considered when you evaluate the quality of the synthesized voice. Many researches on voice synthesis have been developed for decades and some headway has been achieved, nevertheless the quality of the terms about the naturalness of the voice produced still presents gaps, principally regarding the adaptations that the speaking can suffer considering the intonation and the emotiveness associated to the expressiveness of the content to be synthesized.

The efforts on producing the voice artificially started around the year of 1779 when the Russian professor Christian Kratzenstein, made an acoustic resonator similar to the vocal tract, where it was possible to produce the vowel sounds. At a later time, in 1791, Wolfgang von Kempelen created a machine where it was possible to produce simple sounds or combiners, and the difference was that the machine had a pressure chamber simulating the lungs, a kind of vibrating shaft that worked like the human vocal cords and a leather tube representing the vocal tract, allowing the emission of vowel and consonant sounds through the emission of its components. In 1800, Charles Wheatstones rebuild a new version of the Kempelen machine which possessed a more sophisticated mechanism and allowed the production of the vowels and great part of the consonants, including the nasal ones.

The researches continued, but with the objective of constructing electric synthesizers. In 1922, Stewart build a synthesizer composed by source that imitated the functionality of the lungs (excitation) and of the resonant circuits that molds the acoustic resonators of the vocal tract. With this machine it was possible the unique static generation of the vowel sounds with two formants. The first device considered a electric synthesizer was the VODER (*Voice Operating Demonstrator*) developed by Homer Dudley in 1939. It was composed by a bar to select the kind of voice (voiced or voiceless) a pedal to control the fundamental frequency and ten keys that controlled the artificial vocal tract. The basic structure of the VODER is very similar to the systems used on the model source-filter. Currently, the technology involving the voice synthesizers evolves and among these the synthesis that stand out are: by concatenation, articulatory, by formants (rules) and most recently based on Hidden Markov Models (HMM).

The speech synthesizer is the back-end of text-to-speech (TTS) systems (Allen et al., 1987). Synthesizers are also useful in speech analysis, such as in experiments about perception and production. Formant-based (Lalwani & Childers, 1991) is a parametric synthesis very eminent in many speech studies, especially linguistics, because most parameters of a formant synthesizer are closely related to physical parameters and have a high degree of interpretability, essential in studies of the acoustic correlates of voice quality, like male/female voice conversion and simulation of breathiness, roughness, and vocal fry.

3. Formant-based and Klatt's speech synthesizer

The techniques for voice synthesis can be divided in three classes: direct synthesis, the synthesis through the simulation of the vocal tract and the synthesis utilizing a model for the voice production (Styger & Keller, 1994). In the direct synthesis, the signal is generated through the direct manipulation of the waveforms. An example of this kind is the concatenative synthesis in which the sound units, like phonemes, are previously recorded and to produce a new sound, these recorded units are concatenated to compose words and

sentences. This way, in this category there is no necessity of knowing the mechanisms of voice production. The synthesis through the simulation of the vocal tract has the objective of producing the voice through the simulation of the physical behavior of the organs responsible for the production of the speech. The articulatory synthesis is an example of this category.

The synthesis based on a model for voice production consists on method that utilize the source-filter model (Lemmetty, 1999) which allows the modeling of the vocal tract through a linear filter, with a set of resonators that vary in time. The filter therefore is excited through a source, simulating the vibration of the vocal cords for voiced sounds or the comprehension of the vocal tract in the case of a noise. This way the sound is created in the vocal tract and irradiated through the lips. The synthesis by formants, or based on rules, is one of the most prominent techniques of this category, which is fundamented in a set of rules used to determine the necessary parameters to synthesize the speech through a synthesizer. In this synthesis there are two possible structures for a set of resonators: cascade or parallel, since the combination of the two architectures can be used for a better performance. Among the necessary parameters for the synthesizes based on rules, the fundamental frequency ($F0$), the excitation parameter (OQ), the excitation degree of the voice (VO), the frequency and amplification of the formants ($F1...F3$ e $A1...A3$), the frequency of an additional low frequency resonator (FN), the intensity of the low and high regions (ALF, AHF) stand out, among others.

The Klatt's synthesizer (Klatt & Klatt, 1990) is called a formant synthesizer because some of its most important parameters are the formant frequencies: the resonance frequencies of the vocal tract. Basically, the Klatt works as follows: for each frame (its duration is set by the user, often in the range from 5 to 10 milliseconds), a new set of parameters drives the synthesizer. The initial version of the Klatt was codified in FORTRAN and presented good results on simulations for the production of a variety o sounds generated by the human speech mechanism through the correct furnish of parameters of the source control and resonators. Other versions of this synthesizer were developed, and the KLSYN88 was chosen for this chapter, implemented on C language. The choice was made because its source code was donated to the Signal Processing Laboratory (LaPS - *Laboratório de Processamento de Sinais*) from UFPA by the Sensimetrics Enterprise (*http://http://www.sens.com/*, Visited on March, 2010.). Among the main differences between the KLSYN and the KLSYN88, the number of parameters stands out, because the KLSYN88 has 48 parameters. For a complete description of parameters of Klatt's speech synthesizer, the reader is referred to (Klatt & Klatt, 1990). In the latest versions of Klatt's, six parameters are not used anymore - they all are assumed to be zero, reducing our state space to 42 parameters. The problem to solve is: given an utterance to be synthesized, find for each frame a sensible set of parameters to drive the synthesizer. The number of parameters and their dynamic range make an exhaustive search unfeasible. GA was adopted as the main learning strategy.

4. Genetic algorithm

The GAs are mathematics algorithms from the Computational Intelligence area specifically the Evolutionary Computation (EC), where it searches Nature inspired techniques, the development of intelligent systems that imitates aspects from the human behavior, such as: evolution and adaptation. These possess a search technique and optimization based on the probability, inspired by the Darwinian principle of the evolution of the species, and on genetics where it utilizes the natural selection and the genetic reproduction through

the evolutionary operators of selection, crossover and mutation. This way, the most able individuals will have the chance of a longer longevity with higher probability of reproduction, perpetuating the genetic codes for the next generations.

Considering a problem in the GA process, this should be modeled through a mathematical function where the most apt individuals will have a greater or lower result, depending if the object is to maximize or minimize the function. In a population a lot of individuals can exist and each one of them corresponds to a possible solution of the mathematical function. If the function has three variables, for example, each one is represented by a chromosome and their concatenation composes an individual. A chromosome is composed by various characters (genes), each one of them are in a determined position (locus), with its determined value (allele).

The populations are evaluated periodically and it is verified in each one of them which individuals are more able, and these are selected for the crossover. After the crossover, each gene that composes the chromosome can suffer mutation. Following this phase of mutation, a new evaluation of the individuals is made and the ones with greater degree of fitness, that is the ones with the greatest value of the fitness function (performance function), will guarantee the survival for the next population. The genetic operators tend to generate solutions with greater values for the fitness function in which new generations are achieved. This way, the evolutive cycles are repeated until the stop criterion is achieved, it may be: the maximum number of generations, the optimization of the process of convergence or loss of the populational diversity with too similar individuals (do Couto & Borges, 2008).

In addition to the fitness function utilized to measure how much a particular solution will satisfy a condition, the GAs also need another objective function which is the optimization object, it can have a set of restrictions to the values of the variables that compose it. These two functions can be considered identical in optimization numerical problems (Coello et al., 2007).

The GAs present good results, when applied on complex problems that are characterized by:

- Having various parameters that need to be combined in search of the best solution;
- Problems with too many restrictions or conditions that cannot be modeled mathematically;
- Problems with a large search space.

On problems that the optimization with one objective is involved (mono-objective), the GA will try to find an optimal global solution that can be minimum or maximum. In this case, the solution minimize or maximize a function $f(x)$ where x is a vector of decision variables of dimension n, represented by $x = (x_1, ..., x_n)$ belonging to a Ω universe (Coello et al., 2007).

In optimizations with more than one objective function (multi-objective), the task will be the search of one or more optimal solutions, being that none of these can be said to be better than the others considering all of the objectives, because some solutions can bring conflicting scenarios.

5. Multi-objective Optimization Problem

An optimization problem is multi-objective (MOOP - Multi-objective Optimization Problem) when it has various functions that should be maximized and/or minimized simultaneously,

obeying a determined numbers of restrictions that any viable solution should obey. An MOOP problem can be characterized by the Equation 1 (Deb, 2001).

$$\left.\begin{array}{lll} \text{Maximize/Minimize} & f_m(x), & m = 1,2,...,M; \\ \text{subject to} & g_j(x) \geq 0 & j = 1,2,...,J; \\ & h_k(x) = 0, & k = 1,2,...,K; \\ & x_i^{(L)} \leq x_i \leq x_i^{(U)}, & i = 1,2,...,n. \end{array}\right\} \quad (1)$$

where $\boldsymbol{x}$ is a vector of n variables of decision $\boldsymbol{x} = (x_1, x_2, ..., x_n)^T$ that consist on a quantity of values to be chosen during the optimization problem. The limit restriction of the variables (x_i) restricts each variable of decision between the limit below $x_i^{(L)}$ and over $x_i^{(U)}$.These limits represent the space values of the variables of decision, or simply the space of decision. The terms $g_j(x)$ e $h_k(x)$ are functions of restriction and a solution x that can not satisfy all of the restrictions and the $2n$ limits will be considered a non factible solution. Otherwise, it is considered a factible solution. The set of all the possible solutions is denominated viable region, search space or simply S. The objective functions $f_1(x)$, $f_2(x)$, ..., $f_M(x)$, together, are the optimization object and can be maximized or minimized. In some cases a conversion of a maximization problem into a minimization problem may be necessary to avoid some conflicting situations.

Differently from a mono-objective problem in which only a function is optimized, and therefore, a single factible solution, on multi-objective problems there is not only one solution, but a set of them, because it is considered that there is not a single solution that satisfies the objective functions simultaneously, and that some solutions are better only on some objectives, and on others not. Even so, the set of solutions needs to be defined and for this the Optimality of Pareto Theory is used.

6. Dominance and optimal Pareto solutions

The terminology of Pareto establish that a vector of variables is considered optimum (x^*), if a non factible vector x exists in which the degradation of a criterion (value of the objective function) do not cause an improvement on at least another criterion, assuming in this case a minimization problem as example. Therefore, there are no solutions better than the others in all criterions but factible solutions (admissible) that sometimes will be better in some criterions, and sometimes they will not.

The multi-objectives optimization algorithms are based on the domination concept and on its searches, in which two solutions are compared to verify if a relationship of dominance is established one over the other. Considering a problem with M objective functions, where $M > 1$, the solution $x^{(1)}$ dominates the other solution $x^{(2)}$ if the two following conditions are met (Deb, 2001):

1. The solution $x^{(1)}$ is not worse than $x^{(2)}$ in all of the objectives, or $f_i(x^{(1)})$ **not** $\prec f_i(x^{(1)})$ for all $j = 1,2,...,M$ objectives;
2. The solution $x^{(1)}$ is narrowly better than $x^{(2)}$ in at least one objective, or $f_j(x^1) \succ f_i(x^{(2)})$ to at least one $j \in 1,2,...,M$.

where it is considered that the operator $\prec$ denotes the worst and the operator $\succ$ denotes the better. If any of these conditions above is violated, the solution $x^{(1)}$ do not dominates the solution $x^{(2)}$. If $x^{(1)}$ dominates $x^{(2)}$ ($x^{(1)} \succ x^{(2)}$) it is possible to affirm that:

- $x^{(2)}$ is dominated by $x^{(1)}$;
- $x^{(1)}$ is not dominated by $x^{(2)}$;
- $x^{(1)}$ is not worse than $x^{(2)}$.

From this analysis considering the concept of optimality mentioned previously, a set denominated optimal solutions of Pareto is made. These solutions are considered as admissible or efficient, being their set represented by $\bar{P}^*$. The correspondent vectors to these solutions are denominated non-dominated. The aggregation of various non-dominated vectors composes the Pareto front (Coello et al., 2007).

The concept of dominance can be applied to define sets of optimal local and global solutions. The optimal local set of Pareto is defined when, for each x element belonging to the $\bar{P}$ set, an y solution does not exist on its neighborhood to dominate another element of the $\bar{P}$ set characterizing the belonging solutions to $\bar{P}$ with a optimal local set of Pareto. If a solution does not exist in the research space that dominates any other member in the set $\bar{P}$ constitutes an optimal global set of Pareto.

In the presence of multiple optimal solutions of Pareto, it is hard to choose a single solution with no additional information about the problem. Because of that, it is important to find as many optimal solutions of Pareto as possible, obeying the following objectives:

1. Guide the search as close as possible to the global optimal region of Pareto and;
2. Keep the populational diversity in Pareto optimal front.

7. Non-Dominated Sorting Genetic Algorithm II

The NSGA-II (*Non-Dominated Sorting Genetic Algorithm II*) is a Multi-Objective Evolutionary Algorithm (MOEA) based on the *a posteriori* technique of search with emphasis in the search for diverse solutions with the goal to generate different elements in the optimal set of Pareto. The process of decision by a solution is made after (*a posteriori*) the realization of complete search by optimal solutions.

This method was proposed in (Deb et al., 2000) as a modification of the original algorithm mentioned in (Srinivas & Deb, 1994). The main characteristics are the elitism, the ranking attribution and the crowding distance. The elitism is used as a mechanism for the preservation and usability of the best solutions found previously on posterior generations. Through the ranking, the algorithm is achieves the ordering of the non-dominated solutions of the population. The crowding distance uses an operator of selection by tournament to preserve the diversity between the non-dominated solutions in the posterior execution stages to obtain a good spread of the solutions.

In the NSGA-II, the population Q_t is created from the parent population P_t, where both have N individuals and are combined to form together the population R_t, size $2N$. After this junction, it is performed an ordering of the best solutions to classify all the population R_t. Even though it requires a greater computational effort, the algorithm allows the checking of a

non global domination between the populations P_t and Q_t. With the ending of the ordering of the non-dominated solutions, the new set P_t is created and filled by solutions with different non-dominated fronts $(F_1, F_2, ..., F_n)$. The filling starts with the best non-dominated solution from the first front, following the subsequent ones. As only N solutions can be inserted in the new population, the rest of the solutions is simply cast-off. Each F_i set must be inserted in its totality in the new population, and when $|P_{t+1}| + |F_i| > N$ the algorithm introduces a method called crowding distance, where the most disperse solutions are preferred from the F_i set and the other ones are cast-off. The daughter population Q_{t+1} is created from P_{t+1} using the operators of selection by tournament, crossover and mutation. The Figure 1 shows a sequence of the process of the NSGA-II.

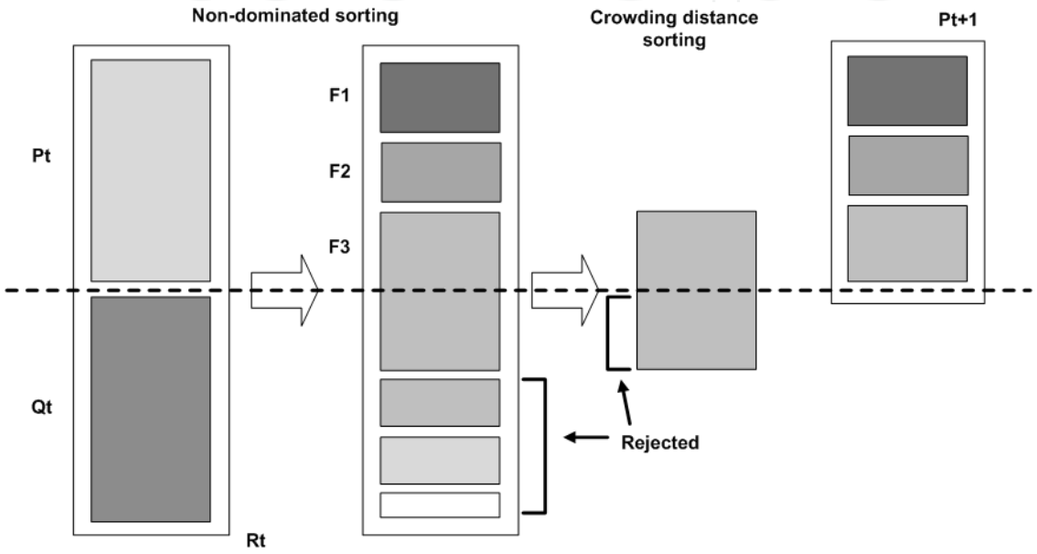


Fig. 1. Diagram that shows the way in which the NSGA-II works - Adapted from (Coello et al., 2007).

To verify the crowding distance, first is calculated the average distance of the two points, both sides of these points, considering all of the objectives. The quantity d_i serves as a estimation of the size of the biggest cuboid that includes the i point without the inclusion of any other point of the population, being called crowding distance. In the Figure 2, the distance from the i-th solution in its Pareto front (filled points) is the average lateral length from the cuboid drew by the dashed lines.

The operator that do the crowding comparison incorporate a modification in the selection method by tournament that considers the crowding of the solution (crowded tournament selection operator). So, the solution i is considered a winner in the tournament by a solution j, if it obeys the following restrictions:

1. The i has the best rank of non-dominance in the population;
2. If both solutions are in the same level, but i has a distance bigger than j ($d_i > d_j$);

Considering two solutions in different levels of non-dominance, the chosen points are the ones with lower level. If both points belong to the same front, then it is chosen localized points in a region with a less number of points, so, solutions with bigger crowding distances.

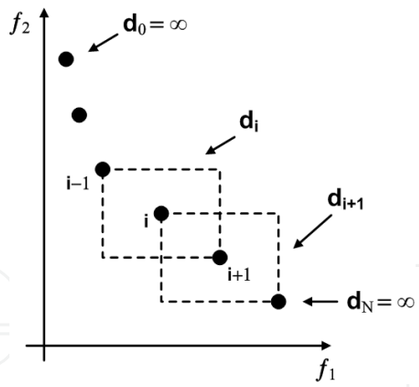


Fig. 2. A graphical illustration of crowding distance.

8. Automatically learning the input parameters

The present chapter has the objective of resolving the issue to estimate the values of the input parameters of a formant synthesizer, as the Klatt for example, aiming to mimicking the human voice. This problem is considered difficult since the parameters specific the temporization of the source and the dynamic values for all the filters. Depending on the quantity of the parameters involved in a possibility of possible combinations can be to big and not viable of being made manually because each parameter has a vast interval of reasonable values. According to Figure 3, it is necessary to estimate initial values for the input parameters of the synthesizer, submitting to the synthesis and then evaluate the synthesized voice through a comparison mechanism with target voice. After the verification, the values of the parameters must be adjusted, that is, new re-estimated values are given as input bringing the synthesis of the voice and a posterior comparison, until the generated voice is as close as possible from the target.

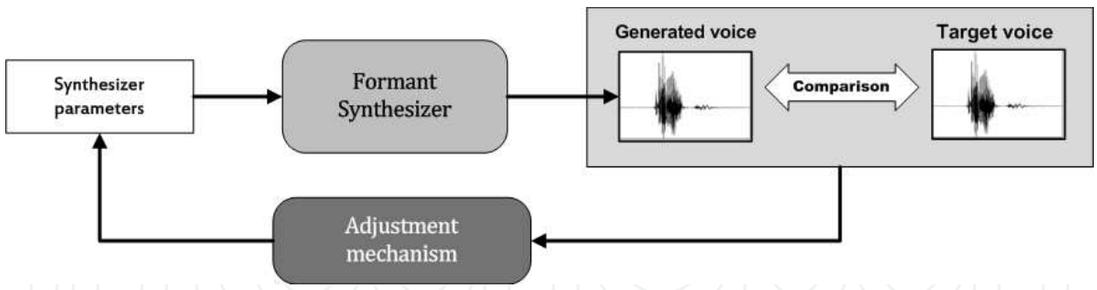


Fig. 3. General problem description.

The Klatt synthesizer is the most utilized among the synthesizers by formants, that is why it was chosen as object of this chapter. Besides that, even not being the focus of this study, the Klatt can be used in TTS systems because it requires low computational cost to produce the voice in high degree of intelligibility, but generally it is hard to reproduce the exact voice signal sound emitted by a human speaker (de Oliveira Imbiriba, 2008).

However, another problem appears in consequence of the option of the formant synthesis that consist in extracting the values of the Klatt's parameters from a voice. These parameters can be generated through the TTS systems, as the Dectalk (Hallahan, 1995), but specifically, to a

single speaker. Some tools and techniques that utilize the signal processing appeared to try to extract them of voice and not having them from text files, but the results were not satisfied.

Considering the complexity of the problem, the proposal is to utilize this type of model to estimate automatically the parameters of a formant synthesizer, developing mechanisms of comparison from voices (synthesized and target) e of adjustments of the re-estimated parameters, attaching this methodology to a technique of extraction of the parameters from the voice in which minimizes the degradation of the synthesized voice.

9. GASpeech framework

With the objective of automatizing the imitation of the natural voice (utterance copy), it was developed in LaPS a methodology that uses MOGA. The methodology called *GASpeech* was adapted from NSGA-II algorithm (Deb et al., 2000) and utilizes three architectures, described later.

As illustrated in Figure 4, the *GASpeech* starts with the input text file and as exit there is the synthesized voice. The rectangles represent programs or scripts and the rounded rectangles correspond to files. First, the text files are submitted to *Dectalk* (Bickley & Bruckert, 2002) where it is a TTS system produced by Fonix Corporation. The generated voices by it possess high intelligibility, but are configured to a single male announcer (Paul). A demo version of this TTS was provided to LaPS for academical purposes. The *Dectalk* generate an exit achieve having 18 parameters in which they are mapped to the 13 parameters of the input file from *HLSyn* through the script *DEC2HLSyn*. The *HLSyn* is utilized to generate the input file of the Klatt synthesizer (version KLSYN88), having the 48 necessary parameters to the voice synthesization. But, of the 48 parameters only 42 are utilized because in this chapter the parallel resonators bank is not considered because of its values being always zero.

In possess of the files having the target voice and the corresponding values from Klatt's parameters, the simulation starts in the *GASpeech*. The population is initialized randomically and each individual is a vector composed by 42 parameters according to the motives exposed previously. The initial population is evaluated taking in consideration the objective functions that can be: spectral distortion (SD), mean squared error (MSE) and cross correlation (CC). After the evaluation, a rank is assigned to each individual. Individuals with best ranks are selected to suffer crossover and mutation. As result, a new population is generated and this one will take all the evaluative process and the genetic operators until the total number of generations is achieved or another stop criterion is fulfilled (Figure 5).

The possible architectures are: *Intraframe*, *Interframe*, *Knowledge-based* or a combination of the last two. Considering that a voice file is composed by various frames, in the *Intraframe* methodology, it is believed that each frame is a conventional problem of GA. So, for example, as the target sentence has the duration of one second and each frame of 10 milliseconds (no superposition) , then 100 problems of GA are solved independently. To start the simulation, the population of the first frame is obtained randomically and the user has the option of utilizing a more adaptive model for the crossover and the mutation or operate them with a fix value. In the *Interframe* methodology, the best individuals from the last population from frame t (obtained $rank = 1$) are copied to the frame $t + 1$. Considering that it may exist a big quantity of able individuals, only 10% of the population can be copied to a following frame and the other individuals are initialized randomically (Borges et al., 2008).

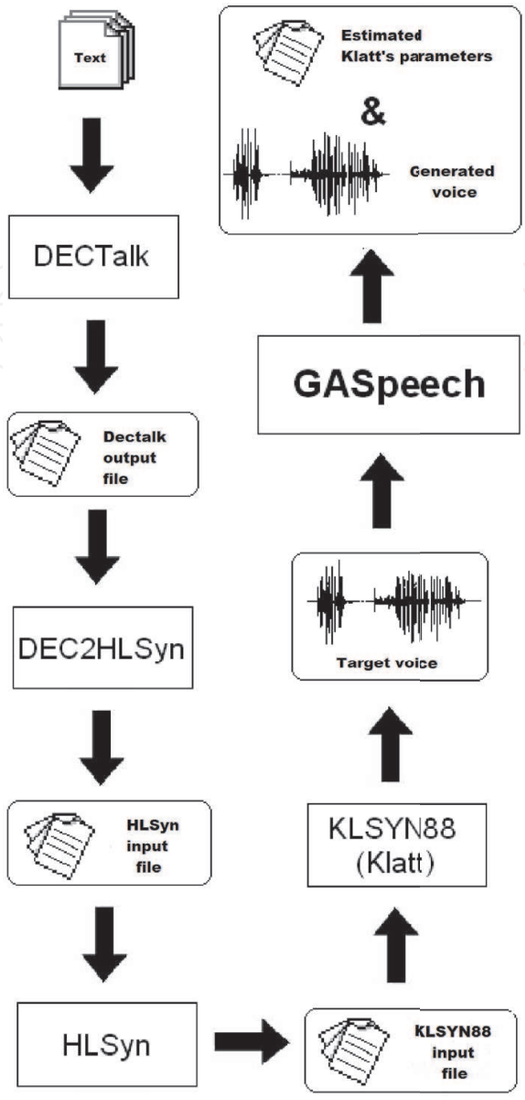


Fig. 4. *GASpeech's* methodology.

In the *Knowledge-based* architecture, for each frame, $N-1$ individuals from the population are initialized randomically and the last individual is inserted through correct values of the Klatt, applying a random variation. The initial idea consists in that this known individual was extracted from the estimations made in tools such as Praat (Boersma & Weenink, Visited on June, 2011.) and Winsnoori (Laprie, 2010), but these tools do not utilize the same version as the Klatt adopted in this chapter, making it necessary therefore the development of a mapping between the different versions. This architecture also can be utilized in conjunction with a *Interframe*. In this case, besides the insertion of an individual partially known in the population initialized randomically, the best individuals from the previous frame population can be copied to a initial population of the following frame. This way, it is tried to keep a previous knowledge in which is widespread to the following populations, lowering this way the quantity of necessary generations to find the correct value of the Klatt's parameters in each frame.

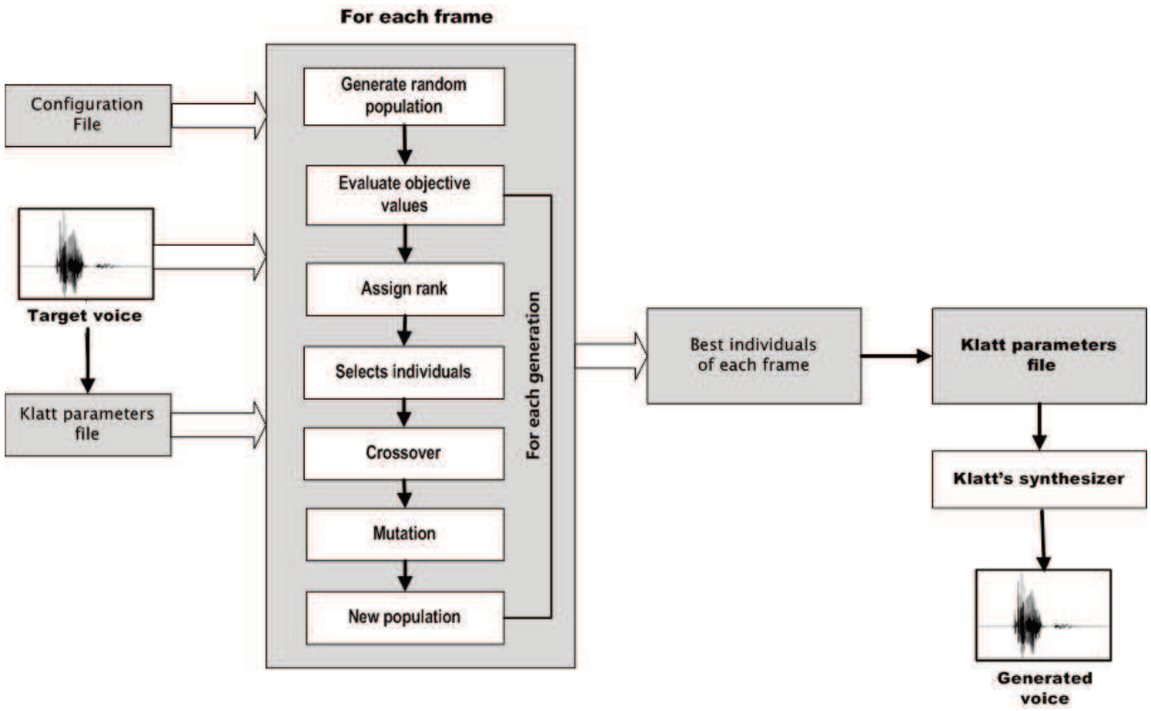


Fig. 5. Functional scheme of *GASpeech*.

The stop criterion defined were three, being them:

- **Convergence:** the simulation is finished when the convergence is obtained, being the convergence parameter (Δ) configured by the user, it can be the SD, the MSE and/or the CC delay.
- **The maximum number of generations:** This criterion is used on traditional GAs and finishes the simulation when the number of generations (*ngen*) is achieved, being this the configured value by the user.
- **The number of generations in evolution:** In this criterion, when the frame achieves the percentage (*ngenwevolve*) of the maximum number of generations with no evolution, the simulation stops. This value is configured by the user and takes in consideration the diversity degree, because when the individuals are the same or too similar, this aspect is not being obeyed.

An individual in the *GASpeech* is composed by a vector of parameters, and in each frame, a single individual must be choose to compose the file with various frames to be synthesized in the end. As the multi-objective optimization can find more than one factible solution, the software is configured to choose the optimal solution of Pareto with lowest value of SD. The fact that the choice befall on the spectral distortion is because this function represents a little better the quality of the generated voice signal, among the other functions. This way, the best individual is the one in which the spectral distortion is lower or equal to 1. If it does not find individuals with this characteristic, the process of decision by the best is used according to the native NSGA-II, based on the elitism, ranking and crowding distance.

On the traditional GAs, the values of the crossover and mutation probabilities are fix, predefined before the initial execution of the algorithm. However, these options can be

inefficient since there is a great chance to take the algorithm to minimum places. With it, (Ho et al., 1999) proposed an heuristic, so the parameters could have their values adapted, although controlled. This strategy aims to vary the probabilities mentioned starting with high values and decaying to lower values, considering this way that in the beginning there is little information about the dominion of the problem and a bigger diversity of the population is supposed to exist. In the end of the optimization process, there is some knowledge about the domain and the best solutions must be explored. In the *GASpeech*, if the options of the mutation and crossover probabilities utilized are adaptable, the initial values of the probabilities are lowered according to Equations 2 and 3.

$$p_m^{n+1} = p_m^n - p_m^n x \delta_m \tag{2}$$

$$p_c^{n+1} = p_c^n - p_c^n x \delta_c \tag{3}$$

where δ_m e δ_c are the decreased rates for the mutation and the crossover, respectively, considering a initial value configured for the probabilities of crossover and mutations (p_m^0 e p_c^0) and minimum values that they can assume ($min(p_m)$ and $min(p_c)$).

As mentioned before, the *GASpeech* works with multi-objective optimization and three objective functions are utilized. These are: SD, MSE and CC delay. It was considered that the lower the value of the three objective functions, better is the individual, so, a way of lowering the values of the functions is search.

The SD is calculated through a FFT routine (*Fast Fourier Transform*) that has as objective evaluate the distortion between the synthesized spectrum ($H(f)$) and the target ($S(f)$). The equation is given by:

$$SD = \sqrt{\frac{1}{f2 - f1} \int_{f1}^{f2} \left[20 \log_{10} \frac{|H(f)|}{|S(f)|} \right]^2 df} \tag{4}$$

The MSE is a manner of quantifying the estimated value from the real one (Imbens et al., 2005). The calculation is made through the Mean Squared Error and how it is desired to minimize the error, the Equation 5 must be minimized.

$$MSE = \frac{1}{n} \sum_{j=1}^{n} (\theta_t(j) - \theta_s(j))^2 \tag{5}$$

where n is the number of samples per frame, $\theta_a(j)$ and $\theta_s(j)$ are, respectively, the index samples j of each frame from the waveforms of the target and synthesized voices.

The delay in the CC can be calculated in the following form: consider two sequences $x(i)$ and $y(i)$ where $i = 0, 1, 2...N - 1$. The normalized cross correlation r in the delay d is defined as:

$$r(d) = \frac{\sum_i [(x_i - \overline{x})(y_{i-d} - \overline{y})]}{\sqrt{\sum_i (x_i - \overline{x})^2} \sqrt{\sum_i (y_{i-d} - \overline{y})^2}} \tag{6}$$

where $\overline{x}$ and $\overline{y}$ are mean from the x and y series, respectively.

Considering the delay in the CC the third objective of the *GASpeech*, it is tried to minimize the delay d for which the function r is maximum, where the signals x and y (Equation 6) are

frames of the original and synthesized voices. The justification to this fact is that when r is maximum to $d = 0$, it means that the signal has maximum correlation in the moment that there is no delay, then the peaks of these signals tend to be aligned.

10. Experiments

The experiments that are made aim the target-voice generated from the Klatt synthesizer version KLSYN88 where it utilizes 48 parameters. The acquisition of the target voice to the various speech sentences was made from a Dectalk TTS system, to a single speaker, Paul. The sentences were processed one by one, as shown on Figure 6, considering the frequency of 11025 Hz.

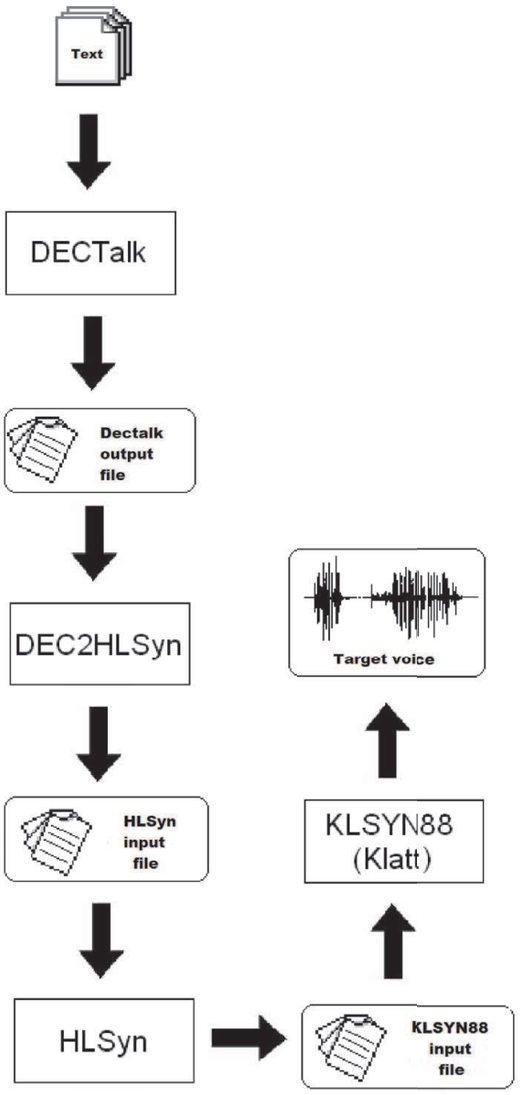


Fig. 6. Preparation of the voice files.

For the experimental effects, nine sentences were chosen considering the variation by phonetic transcription. Each one of them was labeled as shown on Table 1. To evaluate the generated voices it was utilized SD, MSE and CC metrics.

Label	Description
p1007	You don't belong in professional baseball.
p1010	We'll pay you back if you'll let us.
p1013	Draw each graph on a new axis.
p1016	They assume no burglar will ever enter here.
p1032	The wagons were burning fiercely.
p1036	He had four extra eggs for breakfast.
p1069	He recognized his jacket and trousers .
p1074	Our aim must be to learn as much as we teach.
p1159	Blockade is one answer offered by experts.

Table 1. Sentences used.

The experiments made possess as configuration the combinations of the following options:

- **Three objectives**: SD, MSE and CC simultaneously, as objective functions.
- **Two types of architecture**: *Interframe* and the one combined with the Knowledge-based architecture, since the *Intraframe* architecture was less efficient further to the ones mentioned.
- **10 levels of complexity**: the individuals were composed according to the combinations specified on Table 2.

Label	Description
FeB	Formants and bandwidths.
FeBF0	FeB and F0.
20par	FeB and parameters FNP BNP FNZ BNZ A2F A3F A4F A5F A6F AB.
20parF0	FeBF0 and parameters FNP BNP FNZ BNZ A2F A3F A4F A5F A6F AB.
23par	20par and parameters B2F B3F B4F.
23parF0	20parF0 and parameters B2F B3F B4F.
25par	23par and parameters B5F B6F.
25parF0	23parF0 and parameters B5F B6F.
27par	25par and parameters DF1 DB1.
27parF0	25parF0 and parameters DF1 DB1.

Table 2. Levels of complexity.

To initialize a simulation it is necessary a input file in which is generated by the *GASpeech* itself, having the specified configurations on Figure 7. In the example, it is utilized only three Klatt's parameters (F1, F2 and F3) being necessary to inform the value zone that each one of them can receive.

When initializing the simulations it was needed to indicate through the command line the following options:

- **−I <file_name>**: the file of parameters to be passed to the *GASpeech*.

```
80              Number of individuals (multiple of 4)
100             Number of generations
1               Number of  goals
0               Number of constrains
3               Number of real variables (Klatt parameters)
180     1300    Range of values of the variables (F1)
550     3000    Range of values of the variables (F2)
1200    4800    Range of values of the variables (F3)
0.8             Crossover probability
0.024           Mutation probability
8               Distribution index for crossover
10              Distribution index for mutation
0               Number of binary variables
```

Fig. 7. GASpeech's configuration file.

- −**T <file_name>.raw**: audio file (target voice) in the RAW format.
- −**O <file_name>.raw**: name of the output file where its generated in the RAW format too, grouping the best individuals of each frame.
- −**C <value>**: stop criterion based on the informed value.
- −**i <value>**: choose by the *Interframe* methodology with a percentage referring to the best individuals of each frame that will be copied to the next frame.
- −**a**: option to do the adaptation of the values related to the crossover and mutation probabilities.

The utilized values to the parameters during the simulations are described on Table 3.

Parameters	Value
Number of generations (*ngen*)	1000
Population size	200
p_c^0	0.9
p_m^0	0.5
δ_c	0.01
δ_m	0.03
$min(p_c)$	0.1
$min(p_m)$	0.1
Δ	0
$ngenwevolve$	0.3

Table 3. Parameters used in *GASpeech*.

The simulations considered three objectives (SD, MSE e CC), adaptations of crossover and mutations probabilities, *Interframe* architecture isolated and then combined with *Knowledge-based*.

The best results were obtained when it was considered only the formants and the bandwidth (*FeB* – 10 parameters). The *Interframe* methodology combined with the *Knowledge-based*

architecture showed slightly better results, being able to find the reasonable solutions in the previous frame, transferring to the next frame. This caused the increase of the investigation power (exploitation) and lowered the quantity of utilized generations to find the correct value of the Klatt parameters to each frame, because of the almost correct values passed through an individual of the population.

The simulations involving 20, 25 and 27 parameters presents an intelligible generated voice, to all the sentences mentioned, considering an subjective evaluation. But, from the simulations with more than 27 parameters, the quality of the voice decays considerably. This degradation still is most evident when the F0 parameter is considered (fundamental frequency). The combination of the *Interframe* architecture with the *Knowledge-based*, brought little improvement regarding the obtained results, reducing only the quantity of utilized generations, until the achievement of the generated voice.

The Table 4 below shows the values of the SD, MSE, and CC obtained to two of the sentences mentioned before (*p*1007 e *p*1010), considering only the FeB, 20, 25 e 27 parameters with the *Interframe* and this architecture combined with the *Knowledge-based*. The values of the metrics indicate that the MSE and the CC presents little variance between the generated files with a good quality of voice and the ones with a degraded voice, except when the voice quality is very bad as in p1007_27par, p1007_27parK, p1010_27par and p1010_27parK. In these cases, the CC values are negative characterizing a delay between target and synthesized voices.

Label	SD	MSE	CC	Subjective
p1007_FeB	0.3176	0	0.0061	Good
p1007_FeBK	0.2271	0	0.0060	Good
p1007_20par	0.7124	0	0.0059	Good
p1007_20parK	0.7415	0	0.0063	Good
p1007_25par	0.6737	0	0.0059	Reasonable
p1007_25parK	0.6146	0	0.0058	Reasonable
p1007_27par	3.2883	0.0084	-0.0223	Bad
p1007_27parK	2.7798	0.0090	-0.0298	Bad
p1010_FeB	0.2991	0	0.0037	Good
p1010_FeBK	0.2671	0	0.0037	Good
p1010_20par	0.6346	0	0.0035	Good
p1010_20parK	0.6534	0	0.0037	Good
p1010_25par	0.6584	0	0.0037	Reasonable
p1010_25parK	0.6363	0	0.0038	Reasonable
p1010_27par	3.3749	0.0098	-0.0168	Bad
p1010_27parK	3.0881	0.0115	-0.0171	Bad

Table 4. SD, MSE and CC values of generated voices.

The SD when evaluated in the file as a whole do not present coherent values according to the values that you can see in *p*1007_20*par* and *p*1007_20*parK* when compared to *p*1007_25*par* and *p*1007_25*parK* because the generated files with *Knowledge-based* architecture are a little better than those generated only by *Interframe* and therefore should have a lower value for SD. However, when the SD frame value per frame is considered (Figures 8 - 11), its behavior can

be observed with more detail, with the possibility of identifying which frames were generated with the values of the Klatt parameters too different when compared to the target.

In the following Figures the behavior of the SD value can be observed when the quantity of estimated parameters grows. For each sentence (*p*1007 and *p*1010), simulations were performed using 10, 20, 25 and 27 parameters. In Figures 8 and 9, SD values for each frame is shown using only the *Interframe* architecture and this combined with *Knowledge-based*,

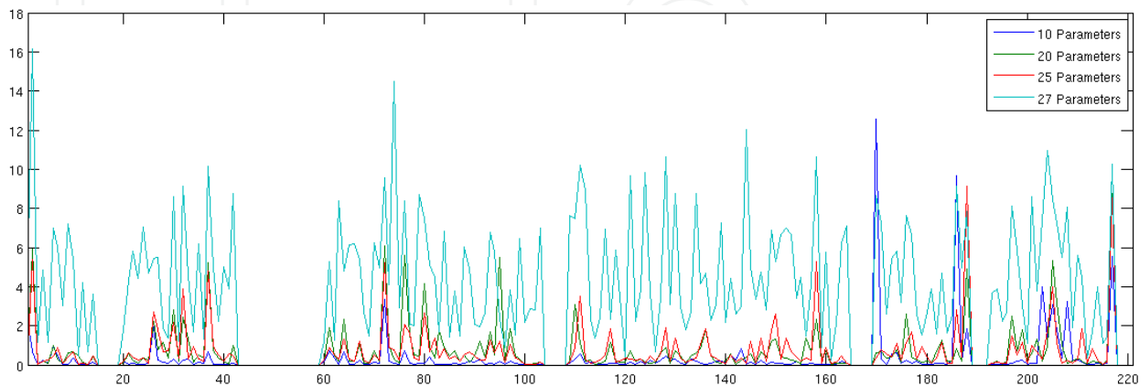


Fig. 8. Spectral Distortion for p1007 sentence with *Interframe* methodology.

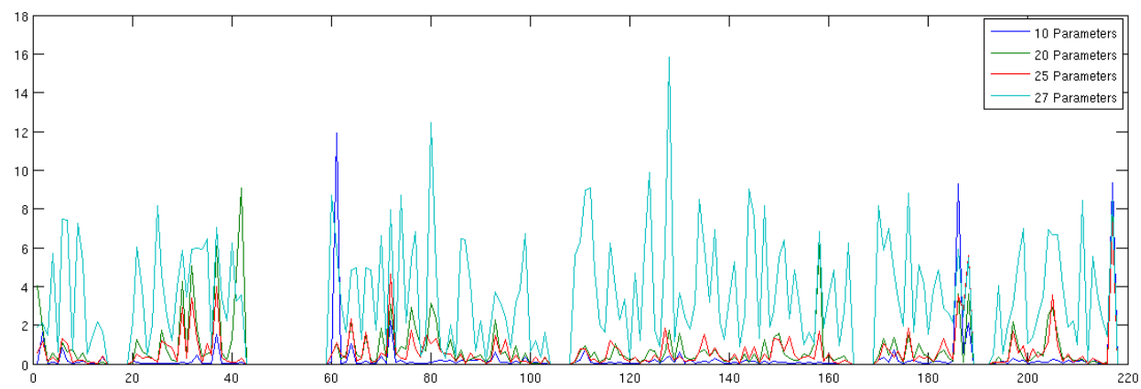


Fig. 9. Spectral Distortion for p1007 sentence with *Knowledge-based* methodology.

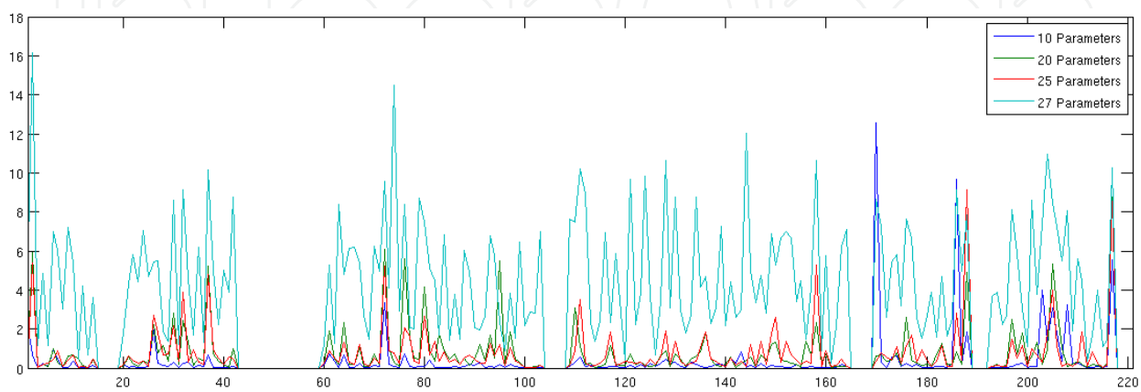


Fig. 10. Spectral Distortion for p1010 sentence with *Interframe* methodology.

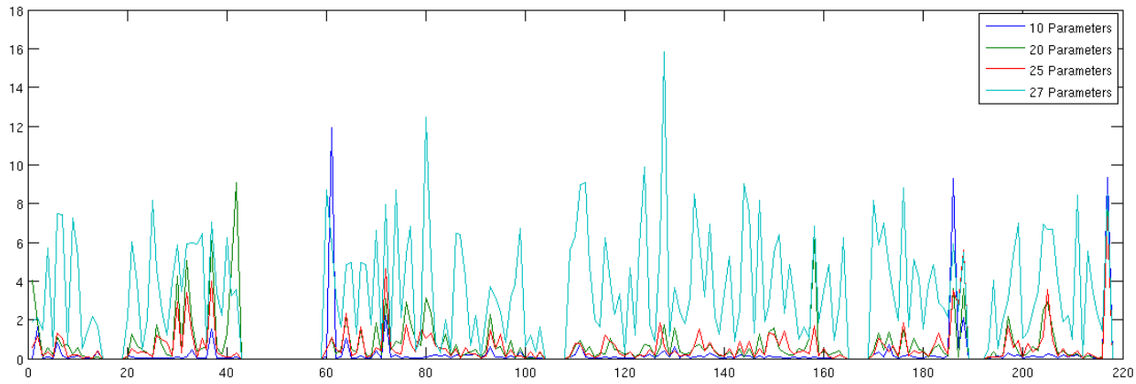


Fig. 11. Spectral Distortion for p1010 sentence with *Knowledge-based* methodology.

repectively. May be noted that the *Knowledge-based* architecture presents lower values of SD by frame compared with *Interframe*, indicating that the partially known individual that is inserted in the population helps to find Klatts parameters value closest to the correct values. The same analysis is true for the sentence *p*1010 as shown in Figures 10 and 11. But it is clear that SD values grows according to the insertion of more parameters to be estimated, indicating the difficulty that the *GASpeech* finds when the increases the amount of the variables involved in the problem.

11. Conclusions

This chapter presented a brief description about the estimation problem of a formant synthesizer, such as the Klatt. The combination of its input parameters to the imitation of the human voice is not a simple task, because a reasonable number of parameters to be combined and each one of them has an interval of acceptable values that must be carefully adjusted to produce a determined voice.

The *GASpeech* used genetic algorithm to estimate the Klatt parameters, however the achieved results were not completely satisfactory, regarding the generated voice when more than 27 parameters are estimated. Good results were achieved only utilizing 10 of the 42 variant parameters. So, careful adjustments is necessary in the framework such as the application of the probabilities of mutation and crossover specific to each Klatt parameter, the utilization of a specific auto-adaptation of these probabilities to a case of real encoding of the variables (Deb et al., 2007) and an specific treatment to better estimate the values of the fundamental frequency due to the fact that an incorrect value of this parameter causes a significant degradation of the quality of the generated voice.

Therefore, it is important to point out that the estimations of the values from the Klatt's parameters, with the objective that they will be as close as possible of the real values, depending on the adequate metric, that really reflect the quality of the produced voice. As seen in the previous session, SD, the MSE, and the CC delay are not adequate when these metrics are calculated considering all frames of the voice files because the metrics values obtained frame by frame is added to obtain an overall average for each synthesized voice file, and in some situations does not reflect the actual quality of voice. Therefore, it is necessary to develop a more efficient mechanism for evaluating the quality of the generated voice as a whole and include it in the *GASpeech* framework.

12. References

Allen, J., Hunnicutt, M. S. & and, D. K. (1987). *From Text-To-Speech: The MITalk System*, Cambridge University Press.

Bickley, C. & Bruckert, E. (2002). Improvements in the voice quality of dectalk reg;, *Proceedings of 2002 IEEE Workshop on Speech Synthesis*, pp. 55 – 58.

Boersma, P. & Weenink, D. (Visited on June, 2011.). Praat: doing phonetics by computer. URL: *http://www.fon.hum.uva.nl/praat/*

Borges, J., Couto, I., Oliveira, F., Imbiriba, T. & Klautau, A. (2008). Gaspeech: A framework for automatically estimating input parameters of klatt's speech synthesizer, *Neural Networks, Brazilian Symposium on* 0: 81–86.

Breidegard, B. & Balkenius, C. (2003). Speech development by imitation. URL: *http://cogprints.org/3328/*

Coello, C. A. C., Lamont, G. B. & Veldhuizen, D. A. V. (2007). *Evolutionary Algorithms for Solving Multi-Objective Problems*.

de Oliveira Imbiriba, T. C. (2008). *Aprendizado supervisionado e algoritmos genéticos para obtenção dos parâmetros do sintetizador de klatt*, Master's thesis, Universidade Federal do Pará.

Deb, K. (2001). *Multi-Objective Optimization using Evolutionary Algorithms*, Wiley.

Deb, K., Agrawal, S. & Pratap, A. (2000). A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: Nsga-ii, *Proceedings of the Parallel Problem Solving from Nature VI*, pp. 849–858.

Deb, K., Sindhya, K. & Okabe, T. (2007). Self-adaptive simulated binary crossover for real-parameter optimization., *GECCO'07*, pp. 1187–1194.

Ding, W., Campbell, N., Higuchi, N. & Kasuya, H. (1997). Fast and robust joint estimation of vocal tract and voice source parameters, *Acoustics, Speech, and Signal Processing, 1997. ICASSP-97., 1997 IEEE International Conference on*, Vol. 2, pp. 1291 –1294 vol.2.

do Couto, I. C. & Borges, J. V. M. (2008). Otimização multi-objetivo aplicada à síntese de voz. Trabalho de Conclusão de Curso apresentado para obtenção do grau de Engenheiro em Engenharia da Computação, do Instituto de Tecnologia, da Faculdade de Engenharia da Computação da Universidade Federal do Pará.

Hallahan, W. I. (1995). Dectalk software: Text-to-speech technology and implementation.

Heid, S. & Hawkins, S. (1998). Procsy: A hybrid approach to high-quality formant synthesis using hlsyn, *Third International Workshop on Speech Synthesis, Jenolan Caves, Australia*, pp. 219–224.

Ho, C., Lee, K. & Leung, K. (1999). A genetic algorithm based on mutation and crossover with adaptive probabilities, *Proceedings of the 1999 Congress on Evolutionary Computation*, Vol. 1, p. 775 Vol. 1.

http://http://www.sens.com/ (Visited on March, 2010.).

Imbens, G. W., Newey, W. K. & Ridder, G. (2005). Mean-square-error calculations for average treatment effects, *IEPR Working Paper No. 05.34* .

Klatt, D. & Klatt, L. (1990). Analysis, synthesis, and perception of voice quality variations among female and male speakers, *Journal of the Acoustical Society of America* 87: 820–57.

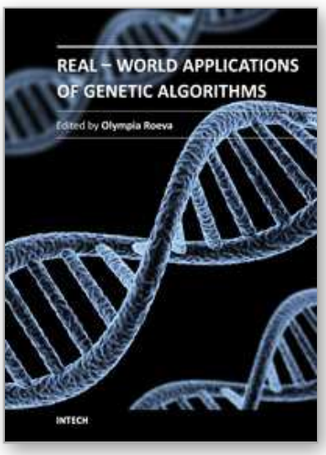
Lalwani, A. & Childers, D. (1991). A flexible formant synthesizer, *Acoustics, Speech, and Signal Processing, 1991. ICASSP-91., 1991 International Conference on*, pp. 777 –780 vol.2.

Laprie, Y. (2010). Winsnoori 1.34 - a speech research tool. URL: *http://www.loria.fr/ laprie/*

Lemmetty, S. (1999). *Review of Speech Synthesis Technology*, PhD thesis, Department Electrical and Communication Engineering - Helsinki University of Technology.

Srinivas, N. & Deb, K. (1994). Multiobjective optimization using nondominated sorting in genetic algorithms, *Evolutionary Computation* 2(3): 221–248.
URL: *citeseer.ist.psu.edu/srinivas94multiobjective.html*

Styger, T. & Keller, E. (1994). *Fundamentals of Speech Synthesis and Speech Recognition: Basic Concepts, State of the Art, and Future Challenges*, John Wiley & Sons Ltd.



Real-World Applications of Genetic Algorithms
Edited by Dr. Olympia Roeva

ISBN 978-953-51-0146-8
Hard cover, 376 pages
Publisher InTech
Published online 07, March, 2012
Published in print edition March, 2012

The book addresses some of the most recent issues, with the theoretical and methodological aspects, of evolutionary multi-objective optimization problems and the various design challenges using different hybrid intelligent approaches. Multi-objective optimization has been available for about two decades, and its application in real-world problems is continuously increasing. Furthermore, many applications function more effectively using a hybrid systems approach. The book presents hybrid techniques based on Artificial Neural Network, Fuzzy Sets, Automata Theory, other metaheuristic or classical algorithms, etc. The book examines various examples of algorithms in different real-world application domains as graph growing problem, speech synthesis, traveling salesman problem, scheduling problems, antenna design, genes design, modeling of chemical and biochemical processes etc.

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

IntechOpen