Evolutionary weight tuning based on diphone pairs for unit selection speech synthesis

Francesc Alías[†] *and Xavier Llorà*[‡]

[†]Dept. of Communications and Signal Theory
Enginyeria i Arquitectura La Salle
Ramon Lull University
Psg. Bonanova 8, 08022-Barcelona, Spain
falias@salleURL.edu

[‡]Illinois Genetic Algorithms Lab (IlliGAL) &
National Center for Supercomputing Applications
University of Illinois at Urbana-Champaign
104 S. Mathews Avenue, Urbana, IL 61801, USA
xllora@illigal.ge.uiuc.edu

Abstract

Unit selection text-to-speech (TTS) conversion is an ongoing research for the speech synthesis community. This paper is focused on tuning the weights involved in the target and concatenation cost metrics. We propose a method for automatically adjusting these weights simultaneously by means of diphone and triphone pairs. This method is based on techniques provided by the evolutionary computation community, taking advantage of their robustness in noisy domains. The experiments and their analyses demonstrate its good performance in this problem, thus, overcoming some constraints assumed by previous works and leading to a new interesting framework for further investigations.

1. Introduction

Concatenative speech synthesis based on unit selection techniques has become a basic technology for Text-to-Speech (TTS) conversion in recent years [1, 2, 3]. These techniques overcome limitations of synthesis from diphone based methods with only one instance per unit. They minimize the number of artificial concatenation points, reducing the need for prosodic modification at synthesis time. This is due to the use of a large database of continuous read speech where many instances of every unit are stored. The selection process makes use of dynamic programming techniques in order to obtain the sequence of units that minimize a cost function at run-time [4]. In fact, it is important to note that the database has to be designed to cover as much linguistic variability as possible, given a particular language or a limited domain [5].

Unit selection TTS systems can produce sentences with good intelligibility and naturalness, nevertheless, this quality cannot usually be maintained along the whole sentence. Therefore, there is still a substantial amount of work necessary for the tuning of all parameters and features involved in the selection process [5]. For instance, the elements of the cost function must be optimized in order to find the set of units from the database that best matches the target sequence of the desired speech sounds. Designing the appropriate measures, as well as correctly tuning them (e.g. adjusting the weights), is essential for achieving high quality synthetic speech.

Weight tuning is one of the most difficult issues in this training process. Hunt and Black presented two approaches in [4]. The first approach was based on adjusting the weights through an exhaustive search of a prediscretized weight space (*weight space search*, WSS). The second approach proposed by the authors used a multilinear regression technique (MLR), across the entire database to compute the desired weights. Later, Meron and Hirose [6] presented a methodology that improved the efficiency of the WSS and refined the MLR method. They also described an extension of these procedures by using unit pairs in the training process and considering prosodic modification at synthesis time. In this paper we propose a novel approach based on population search algorithms for weight optimization.

Section 2 presents the elements involved in the unit selection process. Then, section 3 describes the proposed method for weight training. The conducted experiments and analyses are presented in section 4. Finally, section 5 discusses some conclusions about the work presented in this paper.

2. Unit Selection Cost Function

The cost function plays a leading role in the unit selection process. It takes into account the unit distortion of the candidate unit from the target (*target cost*, C^t), and the continuity distortion between consecutive units (*concatenation cost*, C^c) [4].

$$C^t(t_i, u_i) = \sum_j^p w_j^t C_j^t(t_i, u_i) \qquad (1)$$

$$C^c(u_i, u_{i+1}) = \sum_j^q w_j^c C_j^c(u_i, u_{i+1}) \qquad (2)$$

The target and concatenation costs are defined as a weighted sum of p and q sub-costs, equations (1) and (2) respectively. These measures are calculated as the difference of relevant prosodic and phonetic features. Once the desired features and their corresponding weights are defined, the unit selection process is developed to minimize the cost function obtained from the linear combination of C^t and C^c across the n units of the utterance (see equation (3)).

$$C(t_i^n, u_i^n) = \sum_i^n C^t(t_i, u_i) + \sum_i^{n-1} C^c(u_i, u_{i+1}) \qquad (3)$$

Different measures have been proposed to score these sub-costs, allowing symbolic, scalar and vectorial comparisons [3]. Recent efforts have been carried out in order to improve these measures [7, 8]. As a first approximation we have defined these sub-costs in the prosodic framework, simplifying the computation of the unit selection cost function. Therefore, this paper focuses on the weight training process. The target sub-costs of equation (1) are measured scoring mean differences in pitch, energy and duration between units (follow equation 4). The concatenation sub-costs of equation (2) take into account the local differences in pitch, energy and Mel-frequency cepstral coefficients (MFCC) at the point of concatenation (Right and Left values) (see equation 5).

$$C_j^t(t_i, u_i) = \frac{\left|\overline{P_j}(t_i) - \overline{P_j}(u_i)\right| - m(C_j^t)}{M(C_j^t) - m(C_j^t)} \quad (4)$$

$$C_j^c(u_i, u_{i+1}) = \frac{\sum_1^N \left|P_j^R(u_i) - P_j^L(u_{i+1})\right| - m(C_j^c)}{M(C_j^c) - m(C_j^c)} \quad (5)$$

These measures are normalized by means of the minimum (m) and the maximum (M) values of the sub-cost of parameter P_j for the analyzed unit or set of units. N represents the number of concatenative parameters considered (see equation 5). In our approach, $N = 1$ for pitch and energy sub-costs. This value is the number of cepstral parameters for the MFCC measure.

3. Adjusting the Weights

Training the weights involved in unit selection (w^t and w^c, see section 2) is not a trivial process. As a first approximation, they can be obtained by a hand-tuning process that is perceptually supervised [3, 7]. However, it is believed that automatic training will achieve more robust results. Due to the nature of the problem presented in section 2, it can be modeled as an optimization problem where the decision variables are real-valued. Weighted space search and multilinear regression are the two current main contributions to the automatic approach.

3.1. Weight Space Search

This technique discretizes the search space using a finite set of possible weights $\mathcal{W}$. The optimal weight values are obtained by an analysis-by-synthesis exploration of the chosen variable configurations, that is $|\mathcal{W}|^{p+q}$. Initially, this method was employed for training weights all together [4], and later applied to concatenation weight tuning [1]. Moreover, Meron and Hirose [6] accelerated the process by splitting it into two steps: first precalculating the analysis (selection) and then, running the synthesis (evaluation). Unfortunately, the exhaustive search becomes non-feasible due to its prohibitive computational cost when accurate adjustments are desired.

3.2. Multilinear Regression

This method is much more robust than WSS due to its selection and comparison of all units thorougly, as opposed to the selection of only some data points of the weight space [6]. Moreover, the computational cost is reduced. The regression predicts the objective distance by weighting linearly the sub-costs measures. This training process is fully described in [4], where it is only applied to target weight generation. In [6], MLR is applied to phone pairs, thus, target and concatenation weights can be tuned simultaneously.

3.3. Genetic Algorithms

Genetic algorithms (GA) [9, 10] are population-based search algorithms. Inspired in natural evolution ideas, GA evolve a population of candidate solutions (i.e. weights) adapting them to a given environment, or *fitness* function (i.e. unit selection cost). This process takes advantage of mechanisms such as the survival of the fittest and genetic material recombination.

The scheme of the proposed GA (figure 1) starts with a population generated at random. Each individual is a vector $\mathcal{W}$ (weight configuration) containing the weights to be adjusted, resulting in $\mathcal{W} = (w_1^t, \ldots, w_p^t, w_1^c, \ldots, w_q^c)$. Then, the population is evaluated. Each weight configuration is used for computing the cost function of unit selection based on equation (3), as later explained. The next step performed by the GA is the survival of the fittest weight configuration. This process, known as *selection*, builds a new population sampling the previous one. This process is biased using the computed *fitness*. There are several approaches to the *selection* step, however, we used deterministic binary tournament selection due to its ability to deal with noisy evaluations effectively [10]. Once the new population is obtained, the individuals are recombined in two different phases. The first, *crossover*, given two randomly chosen individuals with a probability p_c, recombines the weight values producing two new offsprings. This process is done using the one point crossover operator [9]. Moreover, the offspring replace their parents in the population. The second phase is known as *mutation*. It introduces random perturbations to the weights values with a given probability p_m. At this point, we have obtained a new population that replaces the original one, starting the evolutionary cycle again. This process stops when a certain finalization criteria is met (i.e. a fixed number of iterations, *iter*).

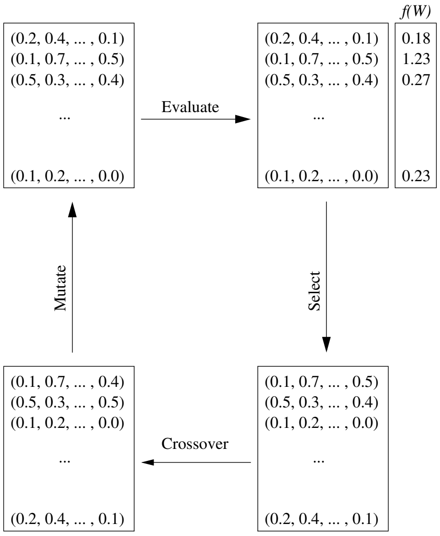
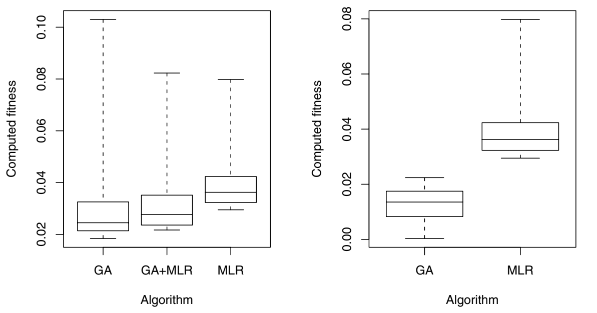


Figure 1: Scheme of simple genetic algorithm.

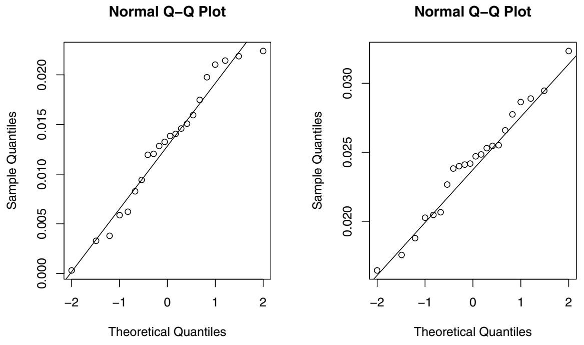
The fitness computation is based on a database that has been clustered into basic units. Computation follows several steps. Firstly, a random target unit is selected. This sampling process allows us to reduce the computational cost required for computing the fitness (cost function). Sampling also adds noise to the evaluations. However, GA can perform efficiently in noise-optimization situations [9, 10]. The second step computes the cepstral distance between all parameterized candidates and the randomly selected target, after a time-alignment following a DTW path. Finally, the k-best acoustic units (this paper assumes k=10) are used to obtain the final value for the cost function (*fitness*). This value is computed as an average of the weighted cost function involving the retrieved k-best individuals and using the weights of the individual $\mathcal{W}$ being evaluated (see equation 3). Thus, the fitness $f(\mathcal{W})$ can be summarized as:

$$f(\mathcal{W}) = \frac{1}{k} \sum_{i \in k\text{-best}} C(t_i^n, u_i^n) \quad (6)$$



(a) Unit /b@/ (b) All units

Figure 2: Final fitness (cost value) computed across the runs.



(a) GA (b) MLR

Figure 3: Quantile-Quantile plots of the costs achieved across all units by the two compared methods.

4. Experiments and Analysis

The acoustic corpus used in the experiments is composed of a simple collection of 1,520 Catalan sentences read by a professional native male speaker. It is not a very large database (approximately 10,000 units), and no greedy algorithm has been carried out in the designing process. However, it can be useful for some initial experiments for our ongoing research in unit selection. Diphones and triphones are the basic units, opposed to half-phones (or half-diphones) [2, 7]. We assume that this approach will provide, at least, the same speech quality as a traditional diphone TTS system with only one instance per unit.

As depicted in [6], we chose unit pairs as training elements, however, we used diphones and triphones instead of phones. Thus, concatenation and target weights are tuned concurrently. In order to show the usefulness of the GA proposed in this paper for weight adjustment, we conducted several experiments on basic unit clusters containing more than 25 instances. The tests were performed using the following parameters: $population\_size = 200$, $iter = 100$, $p_c = 0.3$, and $p_m = 0.003$ [9, 10].

The /b@/ unit cluster (SAMPA notation) has been randomly selected as a benchmark for comparing MLR, GA and GA+MLR configurations. The latter represents a GA with a percentage of initial population ($10\% - 50\%$) obtained from the MLR solution. Figure 2(a) presents the statistics of the cost function across all the instances of the benchmark cluster, given the best weight configuration provided by the different techniques. The weight solution obtained by means of the GA presents a better performance compared to the MLR result in terms of mean cost, despite a higher deviation. The GA+MLR only reduces this deviation without improving the mean cost value, thus, it was discarded throughout the rest of the test. Moreover, we concluded that our simply designed database presents non-comparable distributions of the chosen sub-costs, biasing the solutions obtained by the GA.

We have noticed that the various runs of the GA have obtained different weight values. This is due to the sampling procedure introduced by means of random (noisy) selection, whereby the *fitness* landscape becomes highly multimodal. Nevertheless, the GA can perform better than the classical optimization algorithms due to its noise tolerant nature. After weight computation across all tested units occurred, we concluded that the GA outperfomed the MLR in terms of mean and deviation of the resulting cost function (fitness in figure 2(b)).

The cost function (C, see equations 3 and 6) for both algorithms across the tested units presents a quasi-normal distribution (see figure 3). Thus, a t-test can be used for analyzing the statistical significance of these results. This test shows $C_{GA} < C_{MLR}$ with a confidence level of $p = 3.756 \cdot 10^{-8}$. This result reinforces the conclusion that the GA outperforms MLR for weight tuning in unit selection synthesis.

Figures 4 and 5 depict two pair plots for the weights achieved by both algorithms. The $\omega_{i=1}^{3}$ are the target weights and the $\omega_{i=4}^{6}$ are the concatenation weights. The diagonal of these figures contains the histogram of each weight across all tested units. The remaining sub-figures (ij cells) represent the relationship between weight pairs (w_i, w_j). A superimposed smooth line shows the character of this correlation: linear, quadratic, exponential, etc. The relationship of MLR weights are more linear than the GA ones, however their fitness are worse (see figure 2). Moreover, the biased sub-cost behavior and the unit-dependent tested clusters promote w_3 (the target duration cost) to be the most relevant measure for unit selection, showing the importance of having a well-designed database.

The GA presents a higher computational cost when compared to MLR. However, it grows linearly with the number of instances, in opposition to the WSS approach, which increases exponentially. Locating the optimal solution (the global minimum) is not impossible, nevertheless, it becomes computationally non-feasible. For the WSS approach, an intensive discretization becomes essential (involving several weeks, or even months, of computations) and for the GA method, an elitist process should be included after several runs of the algorithm.

5. Conclusions

A new method based on GA for simultaneously training target and concatenation weights in unit selection TTS conversion, was presented. GA locates high-performing weight configurations, taking advantage of sampling and noise addition techniques. This method overcomes some constraints from previous approaches, proving its usefulness across the developed experiments.

Due to the use of diphone and triphone pairs, the search-

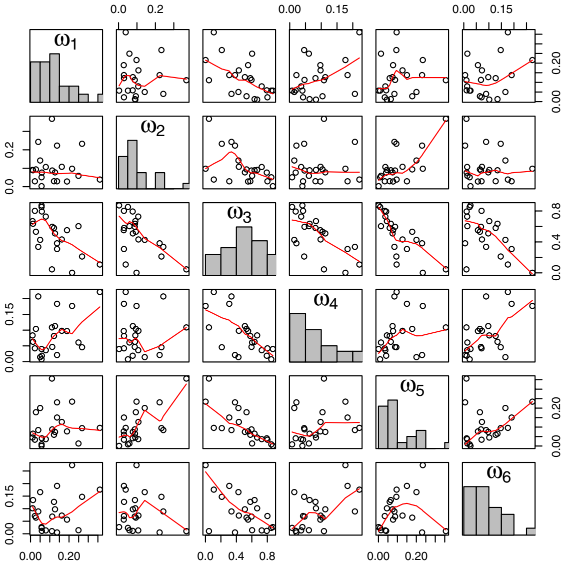


Figure 4: Weight analysis across different units using MLR.

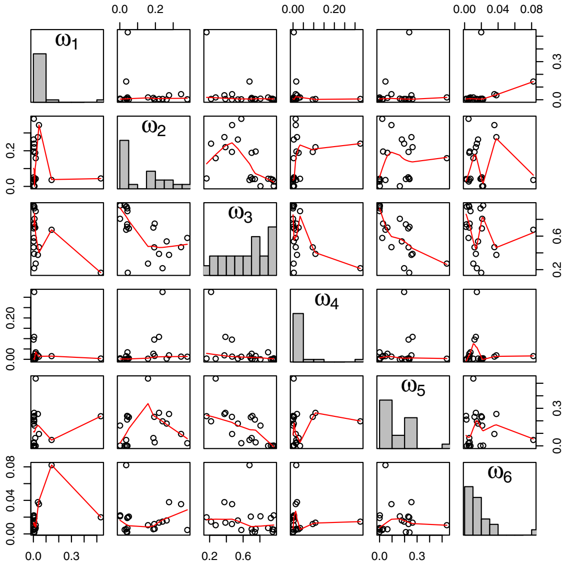


Figure 5: Weight analysis across different units using GA.

ing space is increased considerably, in comparison to the phone pairs' space. Therefore, adjusting the weights for a diphone and triphone selection TTS system is a higher time-consuming process. However, these units allow optimal concatenations at synthesis time. Moreover, the proposed method can be used in the training of weights for: (1) unit-dependent collections, or (2) clusters of similar units, or (3) units altogether. From the analysis presented in the previous section, we conclude that it is essential to take into account a well-designed database for unit selection by means of a greedy algorithm.

Our current ongoing work is focused on (1) designing a new Catalan speech database, and (2) improving several important areas including prosodic modifications when comparing candidates to target units [6], and considering more complex measures for the cost function [7, 8]. Moreover, we are also interested in analyzing context clustering [1] to avoid target cost computation at synthesis time.

Furthermore, formal listening tests will be planned in the near future to evaluate the performance of the GA weights.

6. Acknowledgments

We would truly thank Ignasi Iriondo for many useful discussion during the preparation of this manuscript. We would also like to thank the Generalitat de Catalunya and the D.U.R.S.I. for their support under grant number 2000FI-00679. This work was also supported by the Technology Research, Education and Commercialization Center (TRECC), a program of the University of Illinois at Urbana-Champaign, administered by the National Center for Supercomputing Applications (NCSA) and funded by the Office of Naval Research under grant N00014-01-1-0175. We would also like to thank the Air Force Office of Scientific Research, Air Force Materiel Command, USAF, under grant F49620-00-0163. Research funding for this work was also provided by a grant from the National Science Foundation under grant DMI-9908252.

7. References

[1] A. Black and P. Taylor, "Automatically clustering similar units for unit selection in speech synthesis," in *EuroSpeech*, pp. 601–604, Rodes, Greece, 1997.

[2] M. Beutnagel, A. Conkie, J. Schroeter, Y. Stylianou, and A. Syrdal, "The AT&T Next-Gen TTS system," in *Joint Meeting of ASA, EAA, and DAGA2*, pp. 18–24, Berlin, Germany, 1999.

[3] G. Coorman, J. Fackrell, P. Rutten, and B. Van Coile, "Segment selection in the L&H RealSpeak laboratory TTS system," in *ICSLP*, vol. 2, pp. 395–398, Beijing, China, 2000.

[4] A. Hunt and A. Black, "Unit selection in a concatenative speech synthesis system using a large speech database," in *ICASSP*, vol. 1, pp. 373–376, Atlanta, USA, 1996.

[5] A. Black, "Perfect Synthesis for all of the people all of the time," in *IEEE TTS Workshop 2002 (Keynote)*, Santa Monica, USA, 2002.

[6] Y. Meron and K. Hirose, "Efficient weight training for selection based synthesis," in *EuroSpeech*, vol. 5, pp. 2319–2322, Budapest, Hungary, 1999.

[7] C. Blouin, O. Rosec, P. Bagshaw, and C. d'Alessandro, "Concatenation cost calculation and optimisation for unit selection in TTS," in *IEEE TTS Workshop*, Sta. Monica, USA, 2002.

[8] H. Peng, Y. Zhao, and M. Chu, "Perpetually optimizing the cost function for unit selection in a TTS system with one single run of MOS evaluation," in *ICSLP*, Denver, USA, 2002.

[9] D. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Publishing Company Inc., 1989.

[10] D. Goldberg, *The Design of Innovation: Lessons from and for Competent Genetic Algorithms*. Kluwer Academic Publishers, 2002.