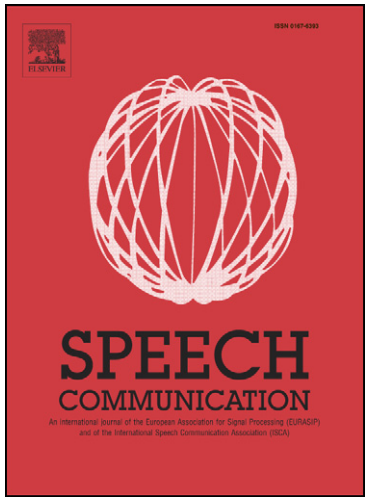
Efficient and reliable perceptual weight tuning for unit-selection Text-to-Speech synthesis based on active interactive genetic algorithms: a proof-of-concept

Francesc Alías, Lluís Formiga, Xavier Llorà



Efficient and reliable perceptual weight tuning for unit-selection Text-to-Speech synthesis based on active interactive genetic algorithms: a proof-of-concept

Francesc Alías, Lluís Formiga, Xavier Llorà

PII: S0167-6393(11)00005-7
DOI: 10.1016/j.specom.2011.01.004
Reference: SPECOM 1962

To appear in: *Speech Communication*

Received Date: 26 June 2009
Revised Date: 5 October 2010
Accepted Date: 3 January 2011

Please cite this article as: Alías, F., Formiga, L., Llorà, X., Efficient and reliable perceptual weight tuning for unit-selection Text-to-Speech synthesis based on active interactive genetic algorithms: a proof-of-concept, *Speech Communication* (2011), doi: 10.1016/j.specom.2011.01.004

Efficient and reliable perceptual weight tuning for unit-selection Text-to-Speech synthesis based on active interactive genetic algorithms: a proof-of-concept

Francesc Alías*, Lluís Formiga

GTM - Grup de Recerca en Tecnologies Mèdia. La Salle - Universitat Ramon Llull. C/Quatre Camins 2, 08022 Barcelona, Spain

Xavier Llorà

National Center for Supercomputing Applications. University of Illinois at Urbana-Champaign. 1205 W. Clark Street, Urbana IL 61801, USA

Abstract

Unit-selection speech synthesis is one of the current corpus-based text-to-speech synthesis techniques. The quality of the generated speech depends on the accuracy of the unit selection process, which in turn relies on the cost function definition. This function should map the user perceptual preferences when selecting synthesis units, which is still an open research issue. This paper proposes a complete methodology for the tuning of the cost function weights by fusing the human judgments with the cost function, through efficient and reliable interactive weight tuning. To that effect, active interactive genetic algorithms (aiGAs) are used to guide the subjective weight adjustments. The application of aiGAs to this process allows mitigating user fatigue and frustration by improving user consistency. However, it is still unfeasible to subjectively adjust the weights of the whole corpus units (diphones and triphones in this work). This makes it mandatory to perform unit clustering before conducting the tuning process. The aiGA-based weight tuning proposal is evaluated in a small speech corpus as a proof-of-concept and results in more natural synthetic speech when compared to previous objective

*Tel.: +34 932902476; fax: +34 932902470
Email address: falias@salle.url.edu (Francesc Alías)
URL: http://www.salle.url.edu/~falias (Francesc Alías)

and subjective-based approaches.

Key words: Perceptual weight tuning, unit selection text-to-speech synthesis, active interactive genetic algorithms
PACS: 43.70.Mn, 43.72.Ja, 43.71.An

---

1. Introduction

The main goal of any Text-to-Speech (TTS) synthesis system is the generation of natural synthetic speech from text. The *unit selection* TTS (US-TTS) approach is one of the current corpus-based synthesis techniques that try to reach this aim [1, 2, 3]. During the last decade, US-TTS systems have been the basis for making a significant jump towards obtaining natural synthetic speech thanks to overcoming the limitations of diphone-based concatenative synthesis, and becoming one of the dominant synthesis techniques [2]. This method generates the synthetic speech signal by means of the selection and concatenation of prerecorded speech units (e.g., phones, diphones, etc.) from a large database of continuous read speech (with many instances per unit). As a result of increasing the database size and coverage, US-TTS synthesis minimizes the number of artificial concatenation points, and reduces the need for prosodic modification at synthesis time when compared to diphone-based TTS systems [3]. Although US-TTS systems can produce sentences with high intelligibility and naturalness in general, this quality sometimes cannot be maintained along the whole utterance [4, 5], yielding synthetic errors when a required phonetic and prosodic context is underrepresented in the speech database [2, 3]. The tuning of all parameters and features involved in the selection process is a key issue for addressing this problem [6, 7]. In particular, the weight tuning is a very important stage in the design of the cost function, which drives the unit selection process. The weights should reflect the relative importance of each sub-cost (acoustic, linguistic, concatenative, etc.) for retrieving the most appropriate set of candidate units from the speech database so as to achieve the best synthetic speech quality (e.g., see [8, 9, 10, 5, 11]).

Since the performance of TTS systems is evaluated based on the perceived speech quality, it is key to embed these subjective criteria into the tuning process of TTS systems. For US-TTS synthesis, the perceptual component may be modeled by the sub-cost functions (computing target and concatenative distances) and their weights. The unit selection weight tuning problem has

been historically addressed by: (*i*) defining an objective measure representing subjective similarities accurately [8, 9, 12], or (*ii*) mapping some sparse samples of users' perception as a post-processing step [13, 14] (see section 3). Nevertheless, weight training is still an open research issue. For instance, in [3] it is stated that heuristically tuned weights provide acceptable high quality speech rendering not worth it to automate the process. In contrast, other works such as [15] try to embed the perceptual preferences into the target cost and automate the processes, since they consider that manually tuning the many weights of the cost function is unlikely to produce optimal results.

This paper describes an *actual* subjective weight tuning methodology. We use active interactive genetic algorithms (aiGAs) [16] to efficiently and reliably embed the perceptual criteria for the weight tuning problem. To that effect, aiGAs are adapted to work on a real-valued continuous search space, besides introducing several evolutionary indicators to assess user interaction. Moreover, we consider diphone and triphone pairs instead of phone pairs [12] during the tuning process, since these are the basic units of our US-TTS synthesis system, as in [3]. As a result, the proposed approach is able to adjust both target and concatenative weights at the same time in contrast to other works (e.g., [9, 15]), avoiding the naive hypothesis they are independent.

This paper is organized as follows. Section 2 briefly describes the basic concepts related to the considered unit selection cost function. Section 3 reviews the main approaches to the weight tuning problem, both objective and subjective. Section 4 describes the main features of aiGAs, which are the core component of the proposed interactive weight tuning methodology at cluster level detailed in section 5. Section 6 describes the conducted experiments on a controlled environment, analyzing users' consistency and comparing the achieved synthetic speech quality to previous weight tuning approaches. Finally, we discuss the results achieved in section 7, and the conclusions and future work in section 8.

2. Unit Selection Cost Function

The cost function plays a key role in the unit selection process of US-TTS systems, retrieving the set of units $u_1^n = (u_1, \ldots, u_n)$ from the speech corpus that potentially yields the best synthetic speech quality given a target sequence $t_1^n = (t_1, \ldots, t_n)$ [8]. To that effect, the cost function takes into account the unit distortion between the candidate unit (u_i) and the target

(t_i), the *target cost* (C^t), and the continuity distortion between consecutive units (u_i and u_{i+1}), the *concatenation cost* (C^c) [8]. The target and concatenation costs are defined as a weighted sum of p target sub-costs $C_j^t(t_i, u_i)(j = 1, \ldots, p)$ and q concatenation sub-costs $C_j^c(t_i, u_i)(j = 1, \ldots, q)$[8]:

$$C^t(t_i, u_i) = \sum_j^p \mathrm{w}_j^t \, C_j^t(t_i, u_i), \tag{1}$$

$$C^c(u_i, u_{i+1}) = \sum_j^q \mathrm{w}_j^c \, C_j^c(u_i, u_{i+1}). \tag{2}$$

where w_j^t and w_j^c stand for target and concatenation weights, respectively.

These sub-costs are generally calculated as the difference of relevant linguistic, acoustic or phonetic features. Once the desired features and their corresponding weights are defined, the unit selection process can be run. Its main goal is to retrieve the set of units from the speech corpus that minimize the given cost function $C(t_i^n, u_i^n)$. In particular, we compute the linear combination of C^t and C^c across the n units of the utterance as [8]

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

It should be noticed that other options could have been chosen, and that several works exist focusing on defining more complex integration schemes for building the cost function from sub-cost functions (see e.g., [5, 17]). Nevertheless, these approaches lay beyond the scope of this paper.

Since this work is focused on validating the proposed methodology for subjective weight tuning, we have only defined the sub-costs in the prosodic framework at this research stage, simplifying the computation of the cost function and centering the perceptual comparisons on speech prosodic variability due to the changes of the weight values. The target sub-costs of Eq. (1) are measured scoring mean differences in pitch (Pit.T), energy (Ene.T) and duration (divided into left halphone, DurL.T, and right DurR.T) between units (following Eq. (4)). The concatenation sub-costs of Eq. (2) take into account the local differences in pitch (Pit.C), energy (Ene.C) and Mel-frequency cepstral coefficients (Mel.C) at the point of concatenation (see Eq. (5)). As it can be observed from these equations, the differences are normalized through a sigmoid function[5, 11] to bound the sub-cost values between 0 and 1.

$$C_j^t(t_i, u_i) = 1 - e^{-\frac{(X^t)^2}{\sigma(X^t)^2}}$$
$$X^t = |\overline{P_j}(t_i) - \overline{P_j}(u_i)| \quad (4)$$

$$C_j^c(u_i, u_{i+1}) = 1 - e^{-\frac{(X^c)^2}{\sigma(X^c)^2}}$$
$$X^c = \sum_{1}^{N} |P_j^R(u_i) - P_j^L(u_{i+1})| \quad (5)$$

where $\overline{P_j}(\cdot)$ represents the mean value of parameter j for the analyzed unit (target t_i or candidate u_i), $\sigma(X^i)$ is the deviation of the parameter differences *across the analyzed units*, and $N = 1$ for pitch and energy sub-costs, while $N = 24$ for the Mel.C measure (12 coefficients plus their respective 12 derivatives). Finally, superscripts R and L of Eq. (5) represent right and left values of parameter $\overline{P_j}(\cdot)$ at the concatenation point, respectively.

3. Related work

As seen in Eq. (3), the cost function is composed of two costs (target and concatenation) computed as a weighted set of sub-costs (see Equations (1) and (2)). These weights w_j^t and w_j^c define the relevance of each sub-cost in the selection of the candidate units. The goal of any scheme for the training of these weights is obtaining the values of $\mathcal{W} = (\mathrm{w}_1^t, \ldots, \mathrm{w}_p^t, \mathrm{w}_1^c, \ldots, \mathrm{w}_q^c)$ which lead to the *highest* synthetic speech quality, observing the following restrictions [18, 19]

$$\sum_{i=1}^{p+q} \mathrm{w}_i = 1, \ \mathrm{w}_i \geq 0. \quad (6)$$

The efficient training of these weights is a key issue for the retrieval of the candidate units regarding the set of considered objective features (e.g., phonetic, acoustic, linguistic, etc.) [6]. However, the weight tuning of the unit selection cost function is one of the toughest problems when designing US-TTS systems [8, 20, 5, 11, 15]. This is due to the fact that the *criterion* for retrieving the *best* set of units from the speech corpus should be able to somehow embed the *subjective* preferences of listeners [20, 14, 5].

The following sections describe the main approaches to the weight training problem in the literature, from both objective and subjective perspectives.

3.1. Objective weight tuning

Weights map the sub-costs to the considered acoustic distance and, hence, model the relevance of sub-cost differences objectively. Weight space search and multilinear regression [8] are the basis of the objective approach, lately extended in [12]. The following paragraphs briefly describe the weight tunning approach.

3.1.1. Weight space search (WSS)

The weight space search technique discretizes the real-valued weight search space $\mathcal{W}$ and then operates on the resulting finite space. Optimal weights are obtained by analysis-by-synthesis exploration of all the possible variable configurations [8]. The process starts with the selection of a target utterance taken from the speech corpus. Next, the unit selection process is run, obtaining a synthesized utterance for each one of the possible weights in the discretized space[14]. Finally, the best sequence of candidate units (and thus, the best set of weights) is determined as the one attaining the minimum distance (typically computed as a cepstral distance [8]) to the target.

This process is repeated for all the selected target utterances and the most *consistent* set of weights is chosen as the final solution [8].

Initially, WSS was used to optimize weights all together [8]. Later it was only used to obtain concatenation weights [9]. WSS tends to be computationally expensive. Proposed improvements attempt to accelerate the search process by splitting the process in two steps [12]: (*i*) precalculate the analysis (selection) and then, (*ii*) run the synthesis (evaluation). Other authors have also applied WSS to solve related problems [14, 11]. Unfortunately, this approach quickly becomes infeasible as we increase the number of weights to adjust and want to maintain a reasonable accurate adjustment [8].

3.1.2. Multilinear Regression (MLR)

Another proposed approach to adjust the weights relies on solving a multilinear regression between an objective measure (typically based on a cepstral distance) and the sub-costs for a given target unit [8]. This process was only applied to target weight generation at the phoneme level in [8]. The target cost is computed according to Eq. (1), only considering candidate units acoustically closest to the target ($k = 20$). This process is repeated for *all* the realizations of the phoneme at hand, being considered as the target unit successively.

MLR can adjust weights at a unit level (e.g., for each phoneme of the speech corpus), or for sets of phonemes (e.g., all nasals, fricatives, etc.), or all phonemes together [8]. MLR was applied to pairs of concatenated phones simultaneously tunning target and concatenation weights [12].

It is important to note that MLR takes into account *all* the units' realizations instead of making use of a prediscretized set of values when adjusting the target weights, yielding theoretically more robust solutions than WSS [12]. Also, the computational cost of the MLR technique increases linearly with the number of considered sub-costs in front of the exponential increase of the WSS [8]. However, it forces a linear relationship between the sub-cost measures and the acoustic distance losing relevant higher order dependences.

3.1.3. Non-linear approaches

After the WSS and MLR seminal proposals, there have been several works introducing different approaches to the weight training process since neither the WSS nor the MLR approaches are optimal. Among these works, [21] defines three generic categories of weights: w_l (*loose*), w_t (*tight*) and w_o (*overlapped*), according to the considered unit types. In [18, 19] a non-linear discriminative weight training technique is presented, which is based on representing the unit selection process as a classification problem. Weights are updated by means of the gradient descent technique, considering the classification error as the objective measure to be optimized (including a heuristic adjustment of several parameters). However, the proposal is only applied for the tuning of the target weights, leaving the extension to the training of concatenation weights for future works.

In [22] a scheme for the application of genetic algorithms to the cost function weight tuning is introduced. Genetic algorithms (GA) [23, 24] are population-based search algorithms. Inspired in natural evolution ideas, GAs evolve a population of candidate weights solutions adapting them to a given environment, in this case, the unit selection cost function. This process takes advantage of mechanisms such as the survival of the fittest and genetic material recombination to deal with the multiple local optima of the cost function, which is a highly non-linear function [15]. GAs can overcame the restrictions of MLR and WSS with a feasible computational effort and use of a cepstral distance as a means of evaluating the quality of the weight configurations within the fitness function [22].

3.2. Subjective Weight Tuning

The subjective (human) criterion may be included in the unit selection cost function through the weights of the sub-costs. There are different approaches for addressing this goal, which we will summarize below.

3.2.1. Manual tuning

The cost function weights can be obtained by some hand-tuning process that is perceptually supervised (see, for instance [25, 26, 27, 11, 3]). This process is based on a preselection of a finite set of weight values, synthesizing several utterances. The optimal set of weights is determined after ranking the preferences of evaluators (generally, speech technology experts) when presented with the resulting synthetic utterances.

This approach involves several problems. It is mandatory to consider a *small* set of weights to make feasible the tuning process (e.g., weights can be chosen among $\{0.25, 0.5, 0.75, 1\}$ values [14]). Thus, the weight search space is dramatically discretized and highly dependent on the synthesis quality. Also, the large number of evaluations necessary to adjust the weights may produce poor or noisy results [28].

As a consequence, tuning the weights of the cost function manually (despite the discretization heuristic used) is unlikely to produce optimal results [15]. US-TTS synthesis systems using perceptually-adjusted weights consistently produce smoother transitions and better voice quality than systems using weights that have been set manually [20].

3.2.2. Post-mapping subjective criterion

There are several works which propose different methodologies for setting the weights of the cost function based on perceptual preference tests as a post-mapping stage. In these works, the Mean Opinion Score (MOS) is used, since it is one of the most widely accepted measures for evaluating the naturalness of synthetic speech subjectively [13, 14]. In [20], an algorithm based on the downhill simplex method is introduced, which searches the set of weights that rank the utterances according to the ranking achieved from the perceptual tests. In [13], a cost function built from a categorical set of factors is introduced, later improved in [14], whose correlation to a set of MOS values is calculated besides optimized, obtaining apparent good average correlation. In [26, 5, 17], the definition of the cost function for conveying perceptual MOS is optimized as accurately as possible, leaving the analysis of

the obtained weights for future works. In section 6, the MOS post-mapping approach is compared to our proposal.

3.2.3. Interactive Genetic Algorithms

Interactive Genetic Algorithms (iGAs) can be defined as an optimization model capable of combining the adjustment of quantitative parameters and the subjective evaluation of the results by replacing the traditional computer-based fitness and selection scheme (objective measure) of classical GAs [23, 24] by a human-driven selection process (subjective data). This kind of algorithm has been applied in several disciplines to fuse human and computer efforts when subjective evaluation is a key element [29], e.g., for perceptual tuning of hearing aids in [30], or as an *actual* perception-guided weight adjustment technique for US-TTS synthesis systems in [31].

The results obtained in [31] showed that the objective weights (based on MLR and GA) were poorly correlated with subjective weights obtained from human perception. However, the experiments evidenced two main problems [31]: the tediousness of the process (user fatigue) and the complexity of maintaining a stable comparison criterion throughout the whole process (user consistency), which may yield unreliable results.

4. Weight tuning by using Active Interactive Genetic Algorithms

As described in previous paragraphs, our previous work proved the usefulness of using GAs (as an objective method) [22] and iGAs (as a subjective method) [31] for addressing the non-linear weight tuning problem [15, 29]. However, further research was needed to improve the quality of the achieved synthesis and to combat user fatigue, though. Later, *active* iGAs (aiGAs) introduced several advances for combating user fatigue [16], greatly reducing the number of evaluations required to achieve high-quality solutions.

aiGAs are a particular kind of genetic algorithms, which are population-based search algorithms [23, 24], with the particularity of learning from user interaction and exploiting the learned knowledge to guide the process of collecting user evaluations [16]. Inspired in natural evolution ideas, aiGAs evolve a population of candidate solutions (in this work, weights) adapting them to the given environment (i.e., the users preferences), by learning from user interaction to anticipate which hypotheses the user may be interested in through a synthetic fitness function [16].

The following paragraphs describe the main issues related to obtaining the synthetic fitness function from users' preferences, and how aiGAs are adapted to the weight tuning problem.

4.1. Synthetic fitness function and Partial-Ordering Graphs

The synthetic fitness function used in aiGAs assumes that the interaction of the user with the evolutionary process can be archived for mining and learning purposes (i.e., modeling users' subjective preferences). As considered in [16, 30], the minimal scenario for collecting meaningful user evaluation is provided by a binary tournament scheme ($s = 2$) [32], where users are prompted to choose after listening the two resulting synthetic solutions (similar to a pair-wise preference test). Thus, given two solutions $\{s_1, s_2\} \in \mathcal{V}$ the user is able to provide three possible outcomes: (i) $s_1 > s_2$, (ii) $s_1 < s_2$, and (iii) $s_1 = s_2$ —or *equal/don't know/don't care*. This pair-wise relative comparison is denoted as partial ordering. As a result, user evaluations introduce a partial order among the solutions (weight configurations) presented so far.

The partial order is made explicit by assembling a graph $\mathcal{G} =< \mathcal{V}, \mathcal{E} >$— as shown in [16]. A vertex in $\mathcal{V}$ represents the solutions presented to the user, whereas the edges in $\mathcal{E}$ represent the partial-ordering evaluations provided by the user. The synthesized sentences (one per tested weight configuration) are presented to the user wisely by the so-called tournament ordering to guarantee that (i) all of them are evaluated, and (ii), the partial order introduced by the user evaluations produces a connected graph $\mathcal{G}$ (see Fig. 1). The partial ordering graph provided by the user may be undirected (equal evaluations are allowed), however, such a graph $\mathcal{G}$ can be easily turned into a normalized directed graph $\mathcal{G}'$ (see Fig. 2) by replacing the equal (undirected edges) by the proper *greater than* or *less than* relations (directed edges)— see [16] for further details. In [30], they also attempted to ensemble global rankings based on pair-wise comparisons in order to provide the tournaments to the users. However, they never explored the model building over the obtained graph to reduce user fatigue by means of *educated guesses* of the user preferences, which is one of the key features of aiGAs, as later explained.

Given a normalized partial-ordering graph $\mathcal{G}'$, aiGAs create a synthetic fitness based on the partial ordering provided by user evaluations and the Pareto dominance concept [33] of multiobjective optimization [34, 35]. A global ordering measure may be computed using a heuristic based on two dominance measures, δ and ϕ, inspired by multiobjective optimization [34,

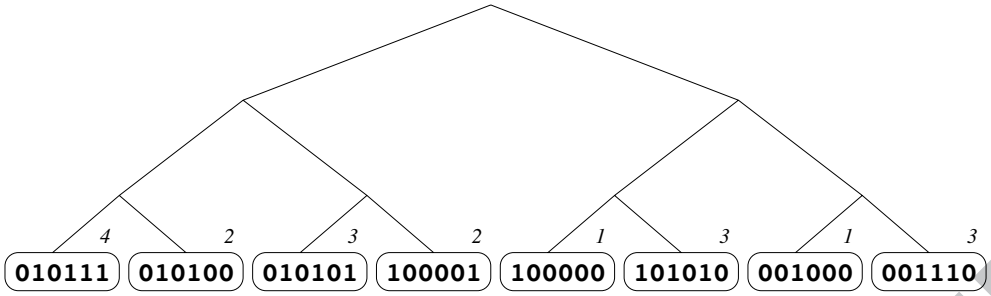
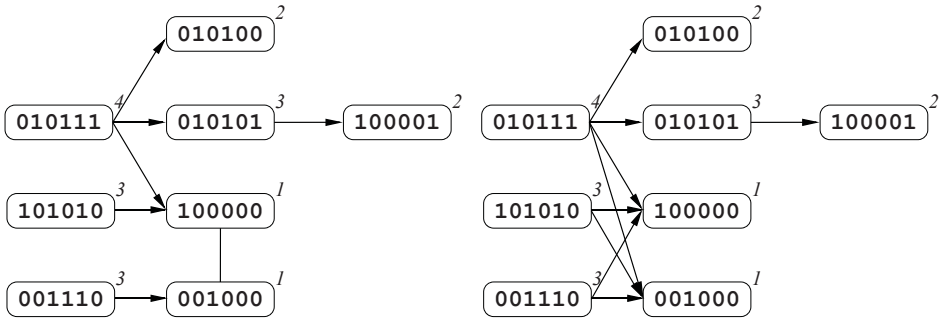


Figure 1: Eight randomly chosen individuals from a population are grouped in seven different tournaments {(010111, 010100), (010101, 100001), (100000, 101010), (001000, 001110), (010111, 010101), (100000, 001000), (010111, 100000)}. The number beside each node simulates the objective function in the user's mind [16].

35]. Let $\delta(v)$ be the number of different nodes present on the paths departing from vertex v, and $\phi(v)$ the number of different nodes present on the paths arriving at v. Table 1 presents $\delta(v)$ or $\phi(v)$ given the graph presented in Fig. 2(b). The estimated fitness of a given solution v may be computed as $\hat{f}(v) = \delta(v) - \phi(v)$. Intuitively, the more solutions a vertex v dominates (is *greater than*), the greater the fitness. Otherwise, the more solutions dominate (are *greater than*) a solution v, the smaller the fitness. The final global estimated ranking $\hat{r}(v)$ is obtained sorting the vertex $v \in \mathcal{V}$ by $\hat{f}(v)$, as shown in Table 1.

A synthetic fitness is obtained applying a regression to the aforementioned global estimated ranking. Following the specifications detailed in [16], the synthetic fitness regression must accomplish two properties: (*i*) fitness extrapolation and (*ii*) order maintenance. ε-SVM [36] satisfies these conditions as it is demonstrated in [16]. Thus, the columns (v,$\hat{r}(v)$) of global estimated ranking are used as training data to obtain a supervised ε-SVM model. Also in [16], it is stated that the number of training examples for the ε-SVM must be $\ell + 2$, where ℓ is the problem size (in our case, $\ell = 7$, which is the number of weights). Thus, the population is composed of 16 individuals (2^4), since 8 individuals (2^3) do not ensure consistency on the $\varepsilon - SVM$ training step ($8 < \ell + 2$, $8 < 9$). Finally, by optimizing such a synthetic fitness (the output of ε-SVM for unevaluated input data) we can obtain new individuals (*educated guesses*) about the user preferences. In this work, the optimization step is conducted by a continuous population-based incremental learning (PBIL) [37], instead of using compact GA, as done in [16], due to the real-valued



(a) *Initial graph* $\mathcal{G}$ (b) *Normalized graph* $\mathcal{G}'$.

Figure 2: Partial ordering graph built from the comparisons provided by a user based on the tournaments of Fig. 1. Direction on the arrows indicates *greater than* relations. When no direction is provided, equality is assumed [16].

representation of the weight tuning problem [28].

4.2. aiGAs for weight tuning

Figure 3 presents the basic execution flow of the proposed aiGA-base methodology for the weight tuning of the cost function. It starts with a population generated at random. Each individual is a vector W (weight configuration) containing the weights to be adjusted, that is $\mathcal{W} = (\mathrm{w}_1^t, \ldots, \mathrm{w}_p^t, \mathrm{w}_1^c, \ldots, \mathrm{w}_q^c)$. Then, the population is evaluated by the user. Given an initial set of tournaments presented to the user, the user listens to the product of the synthesis of the proposed weight configurations. Then, we collect the user preferences of the proposed pairs, and a partial-ordering graph is incrementally built by adding this round of user preferences. This graph is used to compute the synthetic fitness function presented on the previous section. Once it is available, the continuous-variable based PBIL optimizes the fitness function. The important output of this process is a probability distribution over the weight configurations. This probability distribution models the current user preferences toward perceptually good solutions. The new round of solutions to be presented to the user is a fifty-fifty combination of previously shown top-ranking solutions (weight configurations), and new solutions sampled out of the learned probability distribution—representing promising solutions or *educated guesses*. The process continues until the finalization criterion is

Table 1: Estimation of the global ranking based on the dominance measure using the partial order presented in Fig. 2(b).

v	$f(v)$	$r(v)$	$\delta(v)$	$\phi(v)$	$\hat{f}(v)$	$\hat{r}(v)$
010111	4	1	5	0	5	1
010100	2	3	0	1	-1	5
010101	3	2	1	1	0	4
100001	2	3	0	2	-2	6
100000	1	4	0	3	-3	7.5
101010	3	2	2	0	2	2.5
001000	1	4	0	3	-3	7.5
001110	3	2	2	0	2	2.5

met—usually a predefined time duration to avoid tiring the user (see section 6).

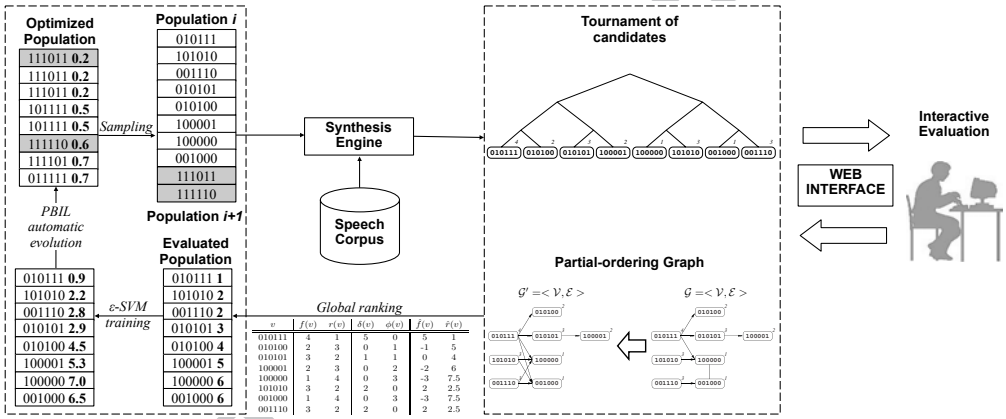


Figure 3: Execution flow of the aiGA-based weight tuning methodology for unit selection TTS synthesis.

5. Obtaining subjective weight patterns at cluster level

As the literature has shown, the weight training process can be conducted at three different levels: at unit level (e.g., for each phoneme of the speech corpus), at cluster level (i.e. for several groups of units) or at corpus level (i.e. for all the corpus' units as a whole) [8, 9]. In our previous works, the weights

were adjusted objectively at unit level using genetic algorithms [22], and at corpus level subjectively through a very reduced set of sentences by means of interactive GAs [31, 28]. However, dividing the unit space into clusters offers an intermediate level of precision between global (all units together) and unit-dependent (one weight set per unit) adjustment techniques [8, 12]. And even more important, it becomes the most feasible way for conducting the weight tuning in the perceptual framework this work is focused on. On the one hand, both the number of tests that the user should conduct and the difficulty involved in comparing variations at unit level (i.e. by presenting only the synthesis of a stand-alone unit or embedding the unit in a carrier word or sentence) discards the weight tuning at unit level. On the other hand, tuning the cost function weights altogether may yield oversmoothed (averaged) weights since all units vary at the same time, making it difficult to know the actual contribution of each sub-cost to the final synthetic quality. Therefore, the interactive weight tuning approach described in this work makes working at cluster level mandatory. As a consequence, this approach allows obtaining different weights for different kinds of units. As an added value, it also minimizes the drawbacks of sparsely populated units by means of distributing them among the clusters.

5.1. Clustering weights by classification and regression trees

In this work, the automatic weights obtained from GA-based weight tuning methodology (see section 3.1.3) are clustered according to their units' phonetic features by using a classification and regression tree (CART) (the *wagon* function of the Festival platform [38]).

On the one hand, grouping units depending on their weights resulting from an automatic process assures consistent clusters in terms of weight patterns (although based on an objective distance), whereas working at the parameter level may not lead to this goal, besides making also indispensable the inclusion of some weighted objective distance to determine unit similarity (see for instance the distance defined in [9]). On the other hand, CART implicitly deals with sparseness of units [9], obtaining the phonetic set that best minimizes the entropy of each cluster, thus, avoiding the clustering of weight patterns obtained from phonetically different units.

The CART's question set includes the following information for each half phone of the unit: the *type* (vowel, consonant, semivowel or silence), the *sonority* (voiced or unvoiced), the *manner of articulation* (plosive, fricative,

etc.) and the *place of articulation* (bilabial, dental, etc.). Since the current cost function considers sum-normalization of weights (i.e. their sum must be 1, see Eq. (6)), the cosine distance has been selected for comparing weight vectors similarity, as neither their direction nor their modulus is the significant characteristic for clustering purposes. Finally, after building the clustering tree, the optimal number of clusters is obtained by computing several clustering impurity measures (see section 6.1).

5.2. Weight patterns

After grouping the speech units thanks to CART clustering, a weight pattern per cluster is obtained by means of the aiGA-based methodology described in section 4. To that effect, firstly, we selected some phonetically balanced sentences containing the largest number of units of each cluster from the speech corpus by means of a simple entropy maximization algorithm. Next, a specific corpus is built for each one of the selected sentences by considering: (*i*) only one candidate unit (i.e. carrier unit) for those units not belonging to the cluster at hand, and (*ii*) all versions of the units whose weights are to be tuned with the exception of the target ones (i.e. coming from the recorded sentence). Then, the weight tuning process (with the original recorded prosody as the target) is put into users' hands by means of the "*Sin-Evo*" platform [31, 28], where the varying units of the sentences are underlined and the reference sentence (the original recording) is also available. The users are asked to evolve the weights interactively for a particular sentence according to the execution flow depicted in Fig. 3. To that effect, 16 different weight settings are compared in a binary tournament (15 comparisons between 2 synthetic speech samples in each iteration) during 3 iterations —this configuration was found to be a good trade-off to obtain reliable weight values [28]. As a result, each user conducts a total of 45 comparisons (i.e., subjective evaluations) for sentence. Moreover, user consistency is controlled by means of the κ measure (see appendix A).

After finishing the tuning process, the unreliable evaluations of each test (i.e., the ones attaining a consistency $\kappa < 1$, see Eq. (7)) are removed. Next, for the reliable results, it has to be determined whether the user has provided significant information until the last iteration or not. This might occur either because a good solution has been found before finishing the tournaments or because the user has become fatigued, being unable to realize the minor variations the last iteration of the evolutionary process typically attain. The stop criterion of the iterative process is determined by considering the evolution of

the number of draws with respect to the total number of comparisons of the directed graph (i.e., the certainty ratio)[39]). The last iteration the certainty ratio increases (before decreasing until the end of the iterative process) is considered as the point where the user has converged (see Fig. 6). Thanks to this measure, it can be observed that users tend to draw final comparisons of the run as they become fatigued.

Once the stop criterion is determined, the best weights coming from the complete order $\hat{r}(v)$ of each user-test sentence pair are used to obtain the weights patterns for each cluster c. These weights are the ones whose fitness is over 0.9 in the $[0, 1]$ range. To that effect, the median is calculated for each weight throughout the whole set $\mathcal{W}^c$ (we use median and not average because of working with *discrete values*) resulting in the median vector $\bar{\mathrm{w}}^c$. Finally, after normalizing $\bar{\mathrm{w}}^c$ according to Eq. (6), the closest real weight vector of $\mathcal{W}^c$ to the $\bar{\mathrm{w}}^c$ (in terms of cosine distance) is labeled as the weight pattern of the cluster c.

6. Experiments

The main goal of the following experiments is to evaluate the performance of the proposed weight tuning methodology in a controlled environment as a proof-of-concept. To that effect, the experiments have been conducted on a labeled Catalan speech corpus composed of 1520 sentences (containing 9863 units, diphones and triphones, in total). The referred corpus had not been intentionally designed for its use in a US-TTS system framework. However, it allows conducting the desired proof-of-concept of the proposed methodology since there are 100 units with enough variability (with 23 to 282 realizations), making 6064 retrievable candidate units during unit selection.

Three different baseline methods have been considered to obtain the weights for each of the 100 most populated units of the speech corpus: two objective methods, based on MLR and GA, and the MOS post-mapping approach, as subjective method. Moreover, the aiGA-based perceptual weight adjustments have been conducted at cluster level by using the "*Sin-Evo*" platform [31, 28]. The proposed subjective weight tuning methodology is evaluated in terms of user consistency, certainty and correlation among users.

After obtaining the weight patterns for each cluster, the synthetic speech quality obtained through the four different weight tuning methods is evaluated by means of preference tests.

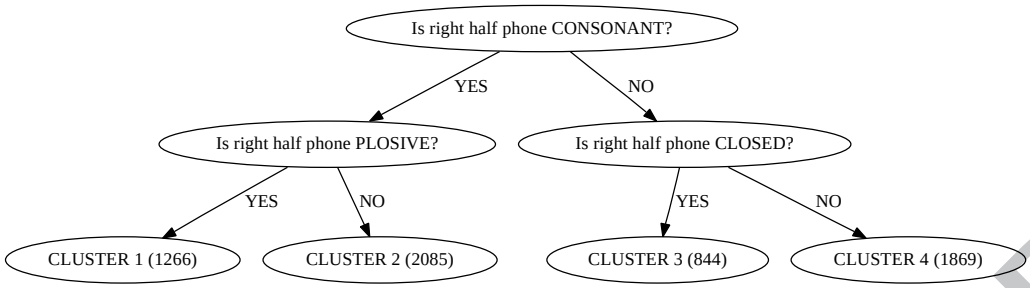


Figure 4: Resulting CART of 4 clusters with the number of instances per cluster indicated between brackets.

6.1. Number of clusters

The main goal of the following analysis is to define the number of clusters considered for the interactive weight tuning experiments. Firstly, a CART is trained on the GA-based weights of the 100 most populated units, resulting in the tree phonetic question set used to partition the whole corpus. Next, the optimal number of clusters is determined. To that effect, we consider several well-known cluster impurity measures: Silhouette, Dunns, Davies-Bouldin, etc. For a review on validation measures the reader is referred to [40].

Table 2 shows that the different measures yield their optimal values between 2 and 4 clusters. After analyzing the distribution of units (see the number of units per cluster of Fig. 4), we selected 4 as the number of clusters to be considered for the perceptual weight tuning. Although cluster 3 and cluster 4 represent quite similar objective weight patterns (see the boxplots of Fig. 5), the quite large number of unit realizations contained in the merged cluster results in a quite unbalanced data partition. Moreover, 4 clusters still yield a feasible number of subjective tests.

6.2. aiGA-based weight tuning

After defining the clusters of units for the aiGA-based subjective weight tuning process, 16 representative sentences (4 per cluster) are selected from the speech corpus by means of a simple greedy algorithm applied to each cluster—containing 30.2 ± 11.6 units, with 5.7 ± 2.4 varying units per sentence. 21 users conducted the interactive weight tuning adjustment on different clusters. Each sentence was adjusted by 4.8 users in average. The weight tuning lasted 13 ± 3 minutes for each sentence approximately (each version could be listened to as many times as desired).

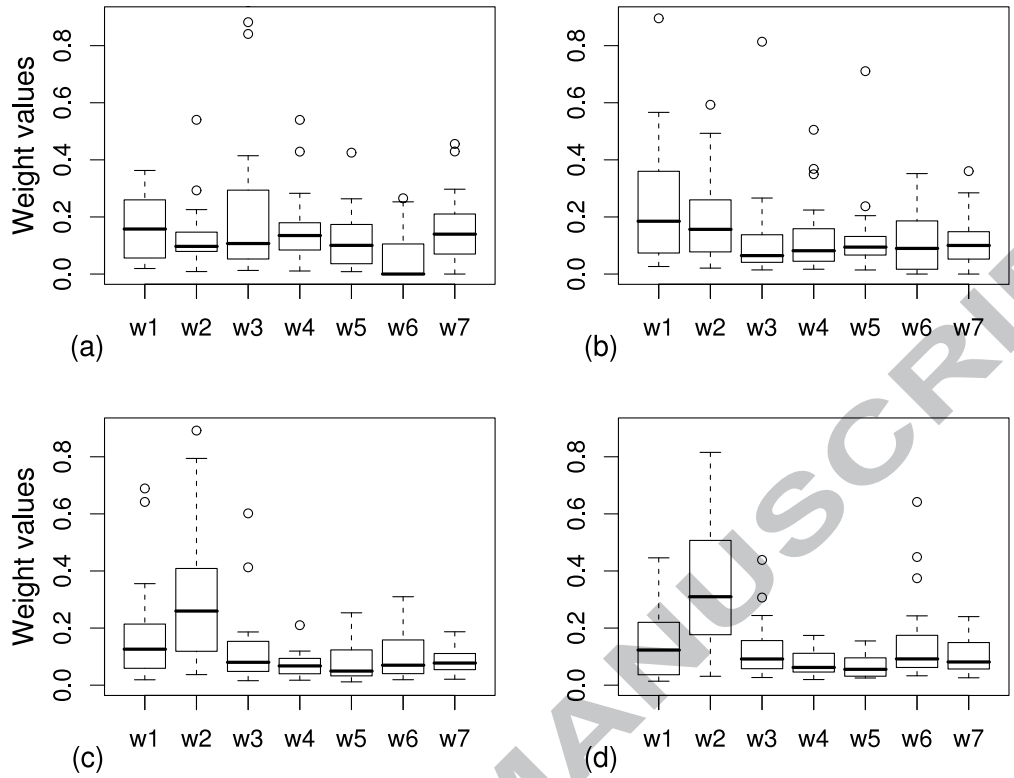


Figure 5: Boxplots of the weight values obtained after clustering the GA-based weight patterns for (a) cluster 1, (b) cluster 2, (c) cluster 3 and (d) cluster 4. Note that w1 stands for DurL.T, w2 for DurR.T, w3 for Ene.C, w4 for Ene.T, w5 for Mel.C, w6 for Pit.C and w7 for Pit.T.

Table 2: Clustering impurity measures used for determining the optimal number of clusters. In bold the optimal value per measure.

Measure/#clusters	**2**	**3**	**4**	**5**
Silhouette	**0.1072**	0.0753	-0.0447	-0.0748
Davies-Bouldin	**1.896**	3.4578	7.5295	7.5091
Calinski-Harabasz	13.114	**24.458**	17.361	14.584
Dunn	**0.8272**	0.2835	0.1072	0.113
C-index	0.4201	**0.349**	0.3545	0.3771
Krzanowski-Lai	0.4547	**3.0914**	2.9821	0.3346
Hartigan	13.114	31.695	**2.4406**	4.4043
weighted inter/intra	0.2281	**0.5348**	0.4734	0.4702
Homogeneity	0.628	**0.6692**	0.6649	0.6474

6.2.1. Collecting reliable information from users' evaluations

The consistency index computed by the κ measure allows considering only those sentence-evaluator pairs, which finish the subjective iterative process in a consistent way (i.e., $\kappa = 1$, see Eq. (7)). As indicated in section 5.2 only consistent results are considered before selecting the best ranked weight patterns that yield the clusters' weight patterns. Figure 6 shows an example of the consistency measure of 5 users for a particular sentence along the tournaments. It can be observed that three users are always consistent, one of them (user1) is able to return to the consistency path thanks to aiGAs, whereas user4 is unable to return to $\kappa = 1$ due to significant consistency drops —a similar behavior was observed in other sentences as already reported in [28]. As a result, 6 of the 77 sentence-evaluator pairs (i.e., the 7.8%) were discarded since the weight tuning process finished inconsistently.

Then, for each test finishing consistently, the last informative iteration (i.e., the stop criterion) is determined by computing the certainty ratio, as described in section 5.2. Figure 6 shows an example of this computation for a particular sentence. It can be observed how each of the 5 users converges between iterations 40 and 51 (the last informative iteration is determined for each user). Starting from that iteration, the certainty ratio keeps decreasing to the end of the test, i.e., the iterative process does not contribute to new information. After analyzing the conducted experiments, the last informative iteration was 44.5 ± 4.8 in average (i.e., around the 80% of the 60 iterations were informative).

6.2.2. Weights convergence and users' correlation

The degree of weight values convergence obtained by the users and their global correlation are evaluated so as to validate the proposed methodology. Firstly, Fig. 7 shows the averaged seven weight values along the iterative process for a particular sentence selected to set an example. As it can be observed, the weight values start from a noisy evolutionary pattern (from the first iteration to iteration 20, approximately), but they converge to stable values at around iteration 50. It is notable that the weights of the different sentences present similar behaviors of weight values evolutions.

Secondly, although each user generates a particular graph built from his/her preferences, the correlation of the resulting best ranked weights for each cluster is also evaluated since it may give information about the reliability of the results. Figure 8 shows the correlation among the users for each one of the 16 sentences whose weights are interactively adjusted grouped per

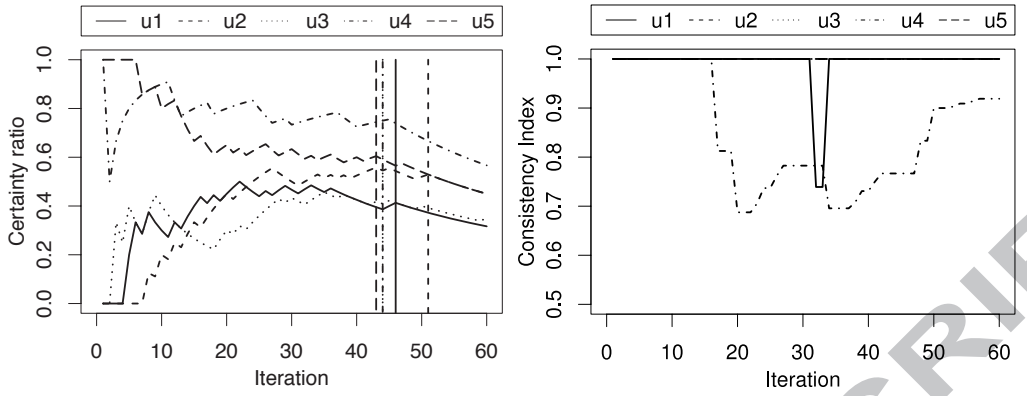


Figure 6: Evolution of five users' consistency measures along the iterative process for a particular sentence of cluster 1. The vertical lines of the leftmost plot indicate the last significant iteration in terms of certainty for each of the 5 users.

cluster. It is worth noting that, although starting from very dispersed values, the correlation values are 0.76 ± 0.02 at the end of the iterative process, which is a very good result considering the subjective problem at hand [14, 17].

6.2.3. Final results

Finally, Fig. 9 shows the boxplots of the resulting patterns of weights obtained for each cluster, following the aiGA-based interactive methodology and the two objective-based weight tuning methods considered in this work (MLR and GA). These patterns were computed as indicated in section 5.2 by considering the weights adjusted by means of MLR and GA of the most populated units at unit level, after being grouped according to the clusters obtained in the experiment reported in section 6.1. It can be observed that the application of MLR and aiGA yield different patterns, but with similar standard deviations ($std_{\mathrm{MLR}} = \pm 0.07$ and $std_{\mathrm{aiGA}} = \pm 0.08$), whereas GA presents a definitely different behavior due to its prominent elitist search ($std_{\mathrm{GA}} = \pm 0.12$).

Moreover, in order to compare the aiGA-based weight values to the ones obtained through the MOS post-mapping technique, the subjective preferences of users in front of synthetic sentences have to be collected. In this work, this information is obtained from the preference test involving aiGA, MLR and GA-based weight patterns described in section 6.3. As a result, three MOS values for each sentence are obtained, following the same scheme

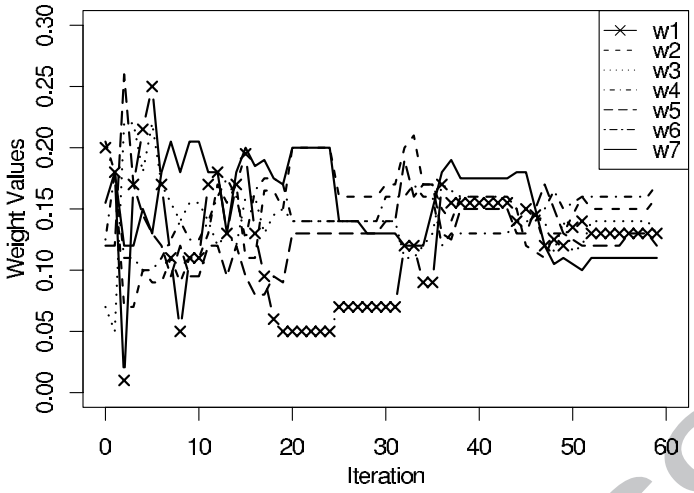


Figure 7: Users' median normalized weight values evolution for a particular sentence of cluster 1. Note that w1 stands for DurL.T, w2 for DurR.T, w3 for Ene.C, w4 for Ene.T, w5 for Mel.C, w6 for Pit.C and w7 for Pit.T.

as the one described in [13, 14], but in a different way since the corpus at hand is not large enough to conduct a size sweep. The CMOS values are converted into MOS values in two stages. Firstly, the CMOS values are normalized into an absolute range for each sentence and weight configuration by averaging the users' punctuations. And secondly, these values are mapped into the 5-point MOS scale (1 to 5), through a simple max-min normalization process. Since the reliable aiGA-based weights are obtained around iteration 45 (see section 6.2.1), we collected 45 evaluations from each user, making the size of the MOS post-mapping evaluation set equivalent to the one used in the aiGA-based approach.

Next, following the process described in [13, 14], the log file of the synthesis process for each sentence and weight configuration is retrieved. Each sub-cost of the cost function is averaged among the selected set of units for each sentence and weight configuration pair. Next, the averaged sub-costs are mapped into the users' MOS punctuations by means of a multilinear regression (MLR) [13, 14]. In this work, the MLR is implemented by Non Negative Least Squares (NNLS) algorithm, which assures positive weight values, obtaining a correlation of -0.49 (see Fig. 10)—a similar value to the ones reported in [17] when the lower range of sub-costs is considered. The resulting weight values are the following: w1 = 0.12, w2 = 0, w3 = 0.20, w4 = 0,

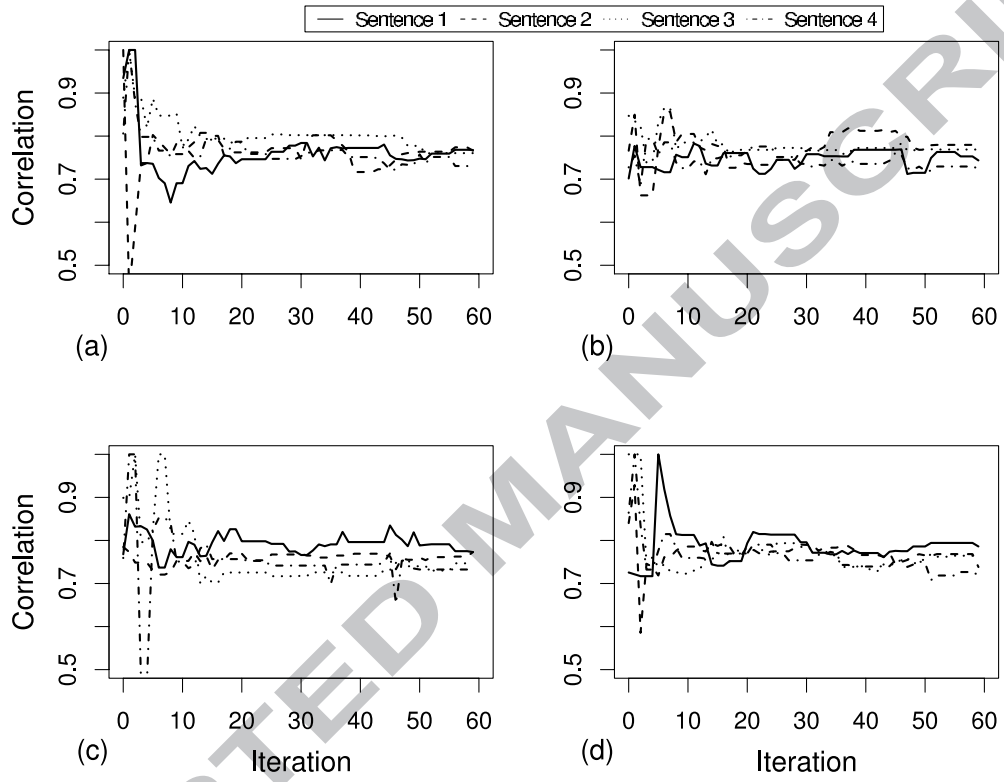


Figure 8: Correlation of the median weight values of the different users for the 4 sentences selected from (a) cluster 1, (b) cluster 2, (c) cluster 3 and (d) cluster 4.

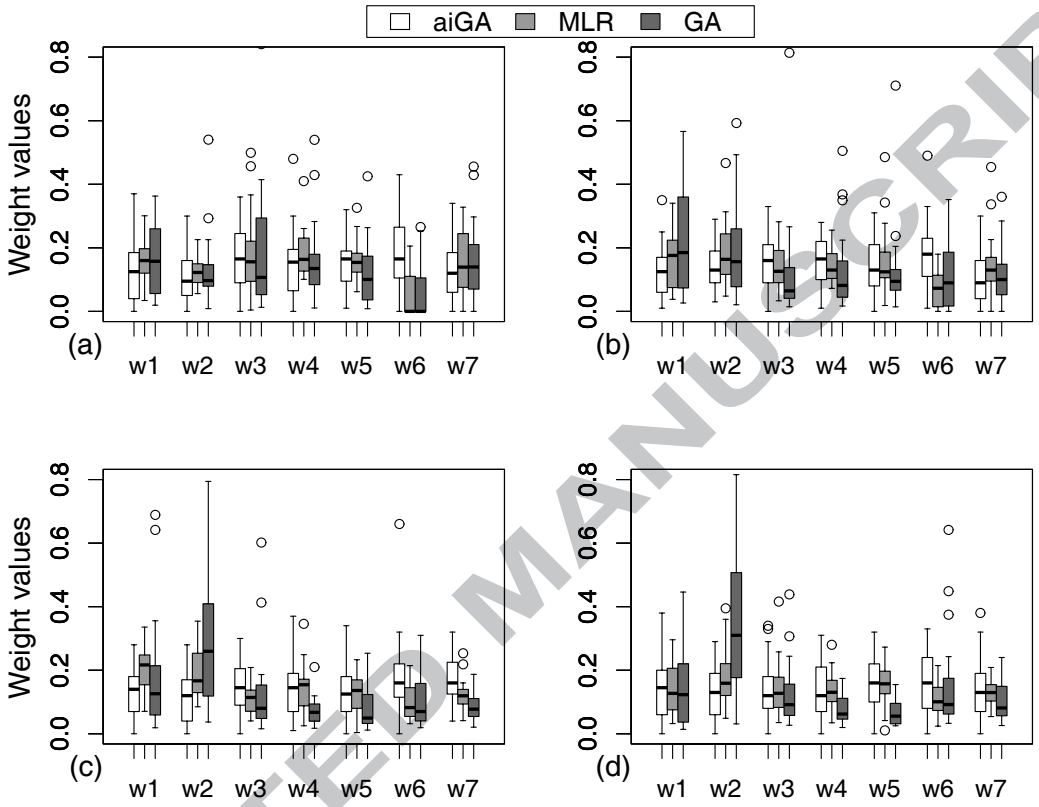


Figure 9: Boxplots of the weight values obtained after applying aiGA, MLR and GA weight tuning approaches on (a) cluster 1, (b) cluster 2, (c) cluster 3 and (d) cluster 4. Note that w1 stands for DurL.T, w2 for DurR.T, w3 for Ene.C, w4 for Ene.T, w5 for Mel.C, w6 for Pit.C and w7 for Pit.T.

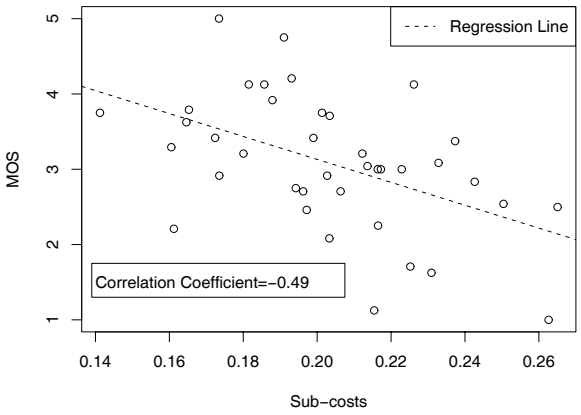


Figure 10: Multilinear regression between the averaged sub-costs and the MOS values obtained from the collected preferences of users in front or the different tested sentences (see section 6.3).

w5 = 0.03, w6 = 0.65, and w7 = 0. Note that w1 stands for DurL.T, w2 for DurR.T, w3 for Ene.C, w4 for Ene.T, w5 for Mel.C, w6 for Pit.C and w7 for Pit.T.

6.3. Preference tests

The purpose of the following experiments is to evaluate the synthetic speech quality obtained by the considered weight adjustments techniques, validating the impact of the introduced efficient subjective tuning approach, as opposed to previous objective (MLR and GA) and subjective (MOS post-mapping) proposals. To that effect, 20 synthetic sentences (different from the ones used for tuning purposes) are considered in two preference tests, which are conducted by 19 evaluators (14 coming from the group who did the weight tuning plus 5 new users included as a control group). In the first one, three synthetic candidates per sentence are synthesized considering the weight patterns obtained by MLR, GA and aiGA-based methodologies in the unit selection search, respectively. In the second one, the aiGA-based syntheses are compared to the ones generated by the weights obtained through the MOS post-mapping approach. In both tests, the evaluators are asked to compare two synthetic candidates per sentence in similar terms of the comparison mean opinion scores (CMOS), but ranging in 5 steps from -2 (significant worse quality) over 0 (about the same quality) to +2 (significant better quality) in order to simplify the evaluators' task to obtain consistent

judgments (see the results on Fig. 11(a)).

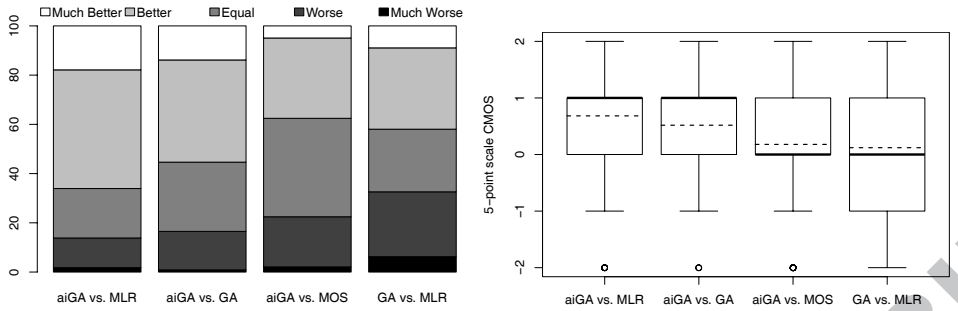
Boxplots of Fig. 11(b) depict the results of the pairwise comparisons (method A vs. method B) for the four weight tuning approaches, showing to what extent the former weight tuning method is better (indicated as positive values) or worse (indicated as negative values) than the latter. The users' preference for the synthetic results obtained using the weight patterns from the introduced aiGA-based methodology can be clearly observed when compared to the objective approaches (MLR and GA). Moreover, there is also a preference to the aiGA-based syntheses when compared to the MOS post-mapping approach (37.54 % of preference for aiGA vs. 22.46% of preference for MOS post-mapping), although there is a slight preference to find both methods with equivalent quality (40% of equals in Fig. 11(a)).

Moreover, in order to evaluate the statistical significance of these results, a paired t-test comparing the users' preferences for each pair of tuning methods is computed. As a result, the test shows that aiGA > MLR (median= 1 and mean= 0.68) with a confidence level of $p < 2 \cdot 10^{-16}$, aiGA > GA with a confidence level of $p = 8.7 \cdot 10^{-13}$ (median= 1 and mean= 0.51). Moreover, aiGA > MOS post-mapping is also statistically significant with a confidence level of $p = 0.00083$ (median= 0 and mean= 0.18). Finally, the difference between MLR and GA is not statistically significant (i.e., $p > 0.05$, with median= 0 and mean= 0.16). Thus, these results reinforce the conclusion that the aiGA outperforms the objective and subjective methods for weight tuning in unit selection synthesis in this proof-of-concept analysis. Furthermore, the 5-users control group presents a similar behavior: aiGA > MLR with a confidence level of $p = 1.9 \cdot 10^{-4}$, aiGA > GA with a confidence level of $p = 2 \cdot 10^{-3}$, aiGA > MOS post-mapping with $p = 0.036$, besides no significant difference between MLR and GA is found. Therefore, it can be concluded that the weight patterns yield sentences with a higher synthetic speech quality appreciated by users who did not participate in the tuning process too.

7. Discussion

In this section, some of the decisions taken during the realization of this work and several issues related to the proposed weight tuning methodology are discussed.

As indicated in the introduction of section 6, in order to decouple the effect of signal post-processing after unit selection, only a small amount of



(a) *Preference bars.* (b) *Preference boxplots.*

Figure 11: Five-point scale CMOS responses of users' preferences when comparing synthetic sentences generated considering the weight values obtained by MLR, GA and MOS post-mapping approaches vs. the aiGA-based methodology, using the latter of the pairs as a reference. The horizontal dotted line within the boxplots represents the distributions' mean values.

(pitch-synchronous) soft concatenation is applied for concatenating the selected units. Some works advocate the inclusion of some signal processing in the adjustment loop [12], while others propose making independent these processes [8]. We think adjusting the weights without post-processing, on the one hand, allows the portability of the results independently of the considered post-processing technique, and, on the other hand, avoids obtaining too similar synthetic candidates during the subjective weight tuning due to the masking caused by the signal processing. Some informal tests including TD-PSOLA in the aiGA-based weight tuning methodology yield very similar results within the tournaments, which in turn, slowed the convergence of the algorithm and frustrated the users since no significant improvement was noticeable during the iterative process (the graphs contained a large number of draws).

Although the weights obtained through the proposed interactive weight tuning methodology are independent of the signal processing module, they have been derived from the speech units of a particular speech corpus. Hence, these weights depend on the corpus at hand, and may not be useful for a different speech corpus. However, in future works we want to study the degree of portability of the users' preferences to other speech corpora since part of their subjective criterion is modeled by means of the ε-SVM included

in the aiGA-based tuning proposal.

The MOS post-mapping proposal evaluated in section 6 can be understood as a refinement of the initial MLR method [8], by substituting the objective acoustic distance by a set of preference values obtained from a MOS listening test. Although it is an interesting approach to correlate the objective distance to human perception, there are some issues the aiGA-based methodology improves. Firstly, MOS post-mapping only allows polinomic regression between sub-costs and sampled perception, loosing any non-linear relation or correlation between sub-costs, which can be dealt with the aiGA-based approach. Secondly, although the number of sentences used in the MOS test may be increased easily, the literature indicates that for getting a reliable optimization, the utterances used for the MOS experiment should be designed carefully so that units have wide coverage for sub-costs [14, 26], in order to avoid obtaining illogical [26] or over-fitted weight values [17]. Finally, it is well known that long perceptual tests yield users fatigue, and thus, noisy results [28]. Therefore, in order to reliably introduce the subjective criteria of users in the weight training problem it is necessary to include some machine-driven technique controlling the process, as the one included in the proposal.

It is important to note that the current methodology allows controlling users consistency avoiding the explicit inclusion of control points (i.e., A-B *vs.* B-A comparisons) along the tournaments thanks to the κ measure. Including control points may disturb the weight tuning process due to user frustration, besides leading to user fatigue. Hence, it's an implicit method for speeding up the weight tuning process, helping to increase user consistency.

8. Conclusions

This work introduced an efficient and reliable weight tuning methodology for unit selection text-to-speech synthesis systems based on active interactive genetic algorithms. The main goal of this work was to show the viability and the reliability of the current proposal, being evaluated objectively (in terms of users' consistency and correlation) and subjectively (in terms of perceptual tests). The clustering process allows diversification of the subjective weight tuning by having a sufficient number of clusters of similar units, avoiding scarcely or massively populated clusters and making the subjective tuning a feasible process. The weights obtained perceptually through the proposed

methodology were preferred in front of the ones obtained by previous objective and subjective-based techniques with regard to preference tests.

It is notable that due to the use of diphone and triphone pairs, the search space is considerably increased in relation to the phone pairs. However, these units allow optimal concatenation at synthesis time and the training cost is not critical, since this is an off-line process (i.e., it is not conducted at synthesis time).

Since the main goal of the current work is to evaluate the viability of the proposed methodology, there are several issues which are postponed for future work. For instance, we have not analyzed in depth the resulting weight values according to the phonetic units they are obtained from or we have kept quite small the number of perceptually evaluated sentences (both for the proposal and the MOS post-mapping approaches) since the speech corpus at hand may not be considered as a typical database for conducting unit selection text-to-speech synthesis. We leave these analyses for future works where larger databases will be considered. Moreover, we want to evaluate the degree of portability of the pattern weights to other speech corpora. And finally, we want to continue working on fusing the final users' preferences to determine the most appreciated weight patterns.

Acknowledgment

This work has been partially supported by the European Commission, Project SALERO (FP6 IST-4-027122-IP). We would like to thank The Andrew W. Mellon Foundation and the National Center for Supercomputing Applications for their support during the preparation of this manuscript.

A. Measuring User Consistency

Given a normalized partial-ordering graph $\mathcal{G}'$, if a vertex v appears more than once in a path computing $\delta(v)$ or $\phi(v)$, then a cycle exists. If such behavior arises, it represents an inconsistency in the user evaluations. Thus, due to the *greater than* relations, the consistency of the user evaluations can be identified [16]. This property is the basis of the consistency metric proposed in [28]. A user will be consistent at time t if no cycles can be found in the normalized partial-ordering graph $\mathcal{G}'$. In order to compute such a measure we need two components: cycle detection capabilities for a given

graph $\mathcal{G}'$ at time t ($\mathcal{G}'^t$), and a heuristic to quantify how much inconsistency the detected cycle is causing, which can be defined as [28]:

$$\kappa\left(\mathcal{G}'^t, \omega\right) = 1 - \left(\frac{1}{|\mathcal{V}'^t|} \cdot \sum_{v \in \chi(\mathcal{G}'^t)} \omega_v\right)^{\alpha} \tag{7}$$

where $|\mathcal{V}'^t|$ is the number of vertices in $\mathcal{G}'$ at time t, ω_v the weight of vertex v (not to be confused with the cost function weights), $\chi\left(\mathcal{G}'^t\right)$ the vertices in the cycles detected in $\mathcal{G}'^t$, and α a global scaling factor bigger or equal than 1. Unless noted otherwise, $\omega_v = 1, \forall v \in \mathcal{V}'^t$ and $\alpha = 1$.

In terms of cycle detection, algorithms 1 and 2 perform the detection of cycles in the graph based on the evaluations made by the user. The idea of these algorithms is to maintain the same criteria on identifying the cycles and, thus, avoiding the redundancy on the detected cyclic parts. For each vertex v in $\mathcal{G}' =< \mathcal{V}', \mathcal{E}' >$ the algorithms explore the relation with the other vertex $\mathcal{V}^{\mathcal{N}}$ not yet processed by means of an accumulative set of visited vertices in the path. Once the set of cyclic paths is formed, the non cyclic parts of the paths are filtered to avoid the subcycle ambiguity. Then, the algorithm sorts each vertex of a cycle by age. The algorithm removes the oldest edge that breaks the cycle. Finally, all the vertices that appear in at least one cycle form the set $\chi\left(\mathcal{G}'\right)$, which is used to define the user consistency measure as introduced in the next section.

Algorithm 1 Algorithmic description of the cycle detection algorithm in $\mathcal{G}'$.

cycleDetection$(\mathcal{G}', i)$

1: Create the empty sets $\mathcal{C}$ of cycles, $\mathcal{V}_{\mathcal{T}}$ of visited vertices
2: Extract the first vertex $v^i \in \mathcal{V} \mid v^i \notin \mathcal{V}_{\mathcal{T}}$
3: Create the set $\mathcal{V}^{\mathcal{N}} = \{\forall\ v \in \mathcal{V}^{\mathcal{N}} : (v \neq v^i) \cap (v \in \mathcal{G}')\}$
4: Create the set $\mathcal{C}_{\mathcal{I}} = \{\forall v \in \mathcal{V}^{\mathcal{N}}\ \forall e(v^i, v) \in \mathcal{E} : cycleExporer(\{v^i\}, v, \mathcal{G}') \subseteq \mathcal{C}_{\mathcal{I}}\}$
5: Filter the non-cyclic parts of paths $\forall c \in \mathcal{C}_{\mathcal{I}}$
6: Sort cycles considering the oldest vertex as the first/last vertex $\forall c \in \mathcal{C}_{\mathcal{I}}$
7: $\mathcal{V}_{\mathcal{T}} \leftarrow \mathcal{V}_{\mathcal{T}} \cup \{v^i\}$
8: $\mathcal{C} \leftarrow \mathcal{C} \cup \mathcal{C}_{\mathcal{I}}$
9: Go to 2 while $\forall v^i \in \mathcal{G}'$: $v^i \notin \mathcal{V}_{\mathcal{T}}$. Else *cycleDetection* $\leftarrow \mathcal{C}$

Algorithm 2 Algorithm to explore all paths departing from v in $\mathcal{V}_\mathcal{I}$

$cycleExplorer(\mathcal{V}_\mathcal{I}, v, \mathcal{G}' =< \mathcal{V}', \mathcal{E}' >)$

1: $\mathcal{V}_\mathcal{I} \Leftarrow \mathcal{V}_\mathcal{I} \cup v$
2: Create the set $\mathcal{R} = \{\forall v^i \in \mathcal{V}_\mathcal{I}\ \forall e(v, v^i) \in \mathcal{E}' : e(v, v^i) \subseteq \mathcal{R}\}$
3: $(\mathcal{R} \neq \emptyset) \Rightarrow return(\mathcal{R})$
4: Create the set $\mathcal{C}_\mathcal{I} = \{\forall v^i \in (\mathcal{V} - \{v\}) \forall e(v, v^i) \in \mathcal{E}' : cycleExporer(\mathcal{V}_\mathcal{I}, v^i, \mathcal{G}') \subseteq \mathcal{C}_\mathcal{I}\}$
5: $return(\mathcal{C}_\mathcal{I})$

References

[1] A. Black and K. Tokuda, "Blizzard Challenge – 2005: Evaluating corpus-based speech synthesis on common datasets", in *Proceedings of the 9th International Conference on Speech Communication and Technology (InterSpeech)*, 77–80 (Lisbon, Portugal) (2005).

[2] A. Black, H. Zen, and K. Tokuda, "Statistical Parametric Speech Synthesis", in *Proceedings of the 32nd IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume IV, 1229–1232 (Honolulu, Hawai'i, USA) (2007).

[3] R. Clark, K. Richmond, and S. King, "Multisyn: Open-domain unit selection for the Festival speech synthesis system", Speech Commun. **49**, 317–330 (2007).

[4] R. Breuer and J. Abresch, "Phoxsy: Multi-Phone Segments for Unit Selection Speech Synthesis", in *Proceedings of the 8th International Conference on Spoken Language Processing (ICSLP)*, 1217–1220 (Jeju Island, South Korea) (2004).

[5] T. Toda, H. Kawai, and M. Tsuzaki, "Optimizing Sub-Cost Functions for Segment Selection Based on Perceptual Evaluations in Concatenative Speech Synthesis", in *Proceedings of the 30th IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 657–660 (Montreal, Canada) (2004).

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

[7] J. R.-W. Yi, "Corpus-based unit selection for natural-sounding speech synthesis", Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts, USA (2003).

[8] A. Hunt and A. Black, "Unit selection in a concatenative speech synthesis system using a large speech database", in *Proceedings of the 21st IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 1, 373–376 (Atlanta, USA) (1996).

[9] A. Black and P. Taylor, "Automatically clustering similar units for unit selection in speech synthesis", in *Proceedings of the 7th European Conference on Speech Communication and Technology (EuroSpeech)*, 601–604 (Rhodes, Greece) (1997).

[10] M. Chu, C. Li, P. Hu, and E. Cahng, "Domain adaptation for TTS systems", in *Proceedings of the 28th IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 453–456 (Orlando, USA) (2002).

[11] F. Campillo and E. Rodríguez Banga, "A method for combining intonation modelling and speech unit selection in corpus-based speech synthesis systems", Speech Commun. **48**, 941–956 (2006).

[12] Y. Meron and K. Hirose, "Efficient weight training for selection based synthesis", in *Proceedings of the 6th European Conference on Speech Communication and Technology (EuroSpeech)*, volume 5, 2319–2322 (Budapest, Hungary) (1999).

[13] M. Chu and H. Peng, "An Objective Measure for Estimating MOS of Synthesized Speech", in *Proceedings of the 7th European Conference on Speech Communication and Technology (EuroSpeech)*, 2087–2090 (Aalborg, Denmark) (2001).

[14] 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 *Proceedings of the 7th International Conference on Spoken Language Processing (ICSLP)*, 1341–1344 (Denver, USA) (2002).

[15] V. Strom and S. King, "Investigating Festival's target cost function using perceptual experiments", in *Proceedings of InterSpeech*, 1873–1876 (Brisbane, Australia) (2008).

[16] X. Llorà, K. Sastry, D. E. Goldberg, A. Gupta, and L. Lakshmi, "Combating User Fatigue in iGAs: Partial Ordering, Support Vector Machines, and Synthetic Fitness", Proceedings of Genetic and Evolutionary Computation Conference (GECCO) 1363–1371 (2005), (Also IlliGAL Report No. 2005009).

[17] T. Toda, H. Kawai, M. Tsuzaki, and K. Shikano, "An evaluation of cost functions sensitively capturing local degradation of naturalness for segment selection in concatenative speech synthesis", Speech Commun. **48**, 45–56 (2006).

[18] S. Park, C. Kim, and N. Kim, "Discriminative weight training for unit-selection based speech synthesis", in *Proceedings of of the 8th European Conference on Speech Communication and Technology (EuroSpeech)*, volume 1, 281–284 (Geneva, Switzerland) (2003).

[19] N. Kim and S. Park, "Discriminative Training for Concatenative Speech Synthesis", IEEE Signal Process. Lett. **11**, 40–43 (2004).

[20] M. Lee, D. P. Lopresti, and J. P. Olive, "A Text-to-Speech Platform for Variable Length Optimal Unit Searching Using Perceptual Cost Functions", in *Proceedings of the 4th ISCA Workshop on Speech Synthesis*, 75–80 (Perthshire, Scotland) (2001).

[21] C.-H. Wu and J.-H. Chen, "Automatic Generation of Synthesis Units and Prosodic Information for Chinese Concatenative Synthesis", Speech Commun. **35**, 219–237 (2001).

[22] F. Alías and X. Llorà, "Evolutionary weight tuning based on diphone pairs for unit selection speech synthesis", in *Proceedings of the 8th European Conference on Speech Communication and Technology (EuroSpeech)*, 1333–1336 (Geneva, Switzerland) (2003).

[23] D. Goldberg, *Genetic Algorithms in Search Optimization and Machine Learning* (Addison-Wesley) (1989).

[24] D. E. Goldberg, *The design of innovation: Lessons from and for competent genetic algorithms* (Kluwer Academic Publisher) (2002).

[25] H. Meng, K. Keung, C.K. Siu, T. Fung, and P. Ching, "CU VOCAL: Corpus-Based Syllable Concatenation for Chinese Speech Synthesis Across Domains and Dialects", in *Proceedings of the 7th International Conference on Spoken Language Processing (ICSLP)*, 2373–2376 (Denver, USA) (2002).

[26] T. Toda, "High-quality and flexible speech synthesis with segment selection and voice conversion", Ph.D. thesis, Graduate School of Information Science, Nara Institute of Science and Technology (2003).

[27] R. Fernandez, R. Bakis, E. Eide, W. Hamza, J. Pitrelli, and M. Picheny, "The 2006 TC-STAR Evaluation of the IBM Text-to-Speech Synthesis System", in *TC-STAR Workshop on Speech-to-Speech Translation*, 175–180 (Barcelona, Spain) (2006).

[28] F. Alías, X. Llorà, L. Formiga, K. Sastry, and G. D.E., "Efficient interactive weight tuning for TTS synthesis: reducing user fatigue by improving user consistency", in *Proceedings of the 31st International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume I, 865–868 (Toulouse, France) (2006).

[29] H. Takagi, "Interactive Evolutionary Computation: fusion of the capabilities of the EC Optimization and Human Evaluation", Proceedings of the IEEE **89**, 1275–1296 (2001).

[30] E. Durant, G. Wakefield, D. Van Tasell, and M. Rickert, "Efficient Perceptual Tuning of Hearing Aids With Genetic Algorithms", IEEE Trans. on Speech and Audio Process. **12**, 144–155 (2004).

[31] F. Alías, X. Llorà, I. Iriondo, X. Sevillano, L. Formiga, and J. C. Socoró, "Perception-Guided and Phonetic Clustering Weight Tuning Based on Diphone Pairs for Unit Selection TTS", in *Proceedings of the 8th International Conference on Spoken Language Processing (ICSLP)*, 1221–1224 (Jeju Island, South Korea) (2004).

[32] D. E. Goldberg, B. Korb, and K. Deb, "Messy genetic algorithms: Motivation, analysis, and first results", Complex Systems **3**, 493–530 (1989).

[33] V. Pareto, *Cours d'Economie Politique, volume I and II* (F. Rouge, Lausanne) (1896).

[34] C. A. Coello-Coello, "An updated survey of GA-Based Multiobjective Optimization Techniques", Technical Report Lania-RD-09-08, Laboratorio Nacional de Informática Avanzada (LANIA), Xalapa, Veracruz, México (1998).

[35] K. Deb, S. Agrawal, A. Pratab, and T. Meyarivan, "A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-II", KanGAL report 200001, Indian Institute of Technology (2000).

[36] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines* (Cambridge Press) (2000).

[37] M. Sebag and A. Ducoulombier, "Extending population-based incremental learning to continuous search spaces", Lec. Notes Comput. Sc. **1498**, 418–427 (1998).

[38] A. Black and P. Taylor, "The Festival Speech Synthesis System: System documentation", Technical Report HCRC/TR-83, Human Communciation Research Centre, University of Edinburgh, Scotland, UK (1997).

[39] L. Formiga, F. Alías and X. Llorà, "Evolutionary process indicators for active IGAs applied to Weight Tuning in Unit Selection TTS synthesis", in *Proceedings of 2010 IEEE Conference on Evolutionary Computation (CEC)*, 2322-2329 (Barcelona, Spain) (2010).

[40] S. Gunter and H. Burke, "Validation indices for graph clustering", in *The 3rd IAPR-TC15 Workshop on Graph-based Representations in Pattern Recognition*, 229–238 (2001).