Differential Evolution Schemes for Speech Segmentation: A Comparative Study

Sunday Iliya
Centre for Computational Intelligence,
School of Computer Science and Informatics,
De Montfort University, The Gateway,
Leicester LE1 9BH, England,
United Kingdom
Email: sundayiliyagoteng@yahoo.com

Ferrante Neri
Centre for Computational Intelligence,
School of Computer Science and Informatics,
De Montfort University, The Gateway,
Leicester LE1 9BH, England,
United Kingdom Email: fneri@dmu.ac.uk

Dylan Menzies
Centre for Electronics and Communication Engineering,
School of Engineering, Media and Sustainable Development,
De Montfort University, The Gateway,
Leicester LE1 9BH, England,
United Kingdom Email: dylan.menzies1@gmail.com

Pip Cornelius
Human Communication Studies,
School of Allied Health Sciences,
De Montfort University, The Gateway,
Leicester LE1 9BH, England,
United Kingdom Email: pcornelius@dmu.ac.uk

Lorenzo Picinali
Interactive and Media Technologies,
School of Engineering, Media and Sustainable Development,
De Montfort University, The Gateway,
Leicester LE1 9BH, England,
United Kingdom Email: lpicinali@dmu.ac.uk

***Abstract*—This paper presents a signal processing technique for segmenting short speech utterances into unvoiced and voiced sections and identifying points where the spectrum becomes steady. The segmentation process is part of a system for deriving musculoskeletal articulation data from disordered utterances, in order to provide training feedback. The functioning of the signal processing technique has been optimized by selecting the parameters of the model. The optimization has been carried out by testing and comparing multiple Differential Evolution implementations, including a standard one, a memetic one, and a controlled randomized one. Numerical results have also been compared with a famous and efficient swarm intelligence algorithm. For the given problem, Differential Evolution schemes appear to display a very good performance as they can quickly reach a high quality solution. The binomial crossover appears, for the given problem, beneficial with respect to the exponential one. The controlled randomization appears to be the best choice in this case. The overall optimized system proved to segment well the speech utterances and efficiently detect its uninteresting parts.**

I. Introduction

Differential Evolution (DE) is a simple algorithm that displays, despite a modest effort for programming and tuning it, a very good performance of a wide variety of optimization problems, see [1] and [2]. For this reason, DE has been used in many real-world applications ranging from control engineering, see [3], to image processing, see [4]. One of the first application domains for DE has been signal processing and more specifically the design of a digital filter, see [5]. Many other studies have shown the efficiency of this algorithmic structure in handling these problems, as shown in [6], [7], and [8]. A comparative study focussing on the capability of DE schemes of handling non-standard filter design problems has been presented in [9].

Since, as shown in [2], [10], [11], and [12], DE is characterized by a limited amount of search moves, modifications of the original scheme can lead to a performance enhancement. These modifications, in some cases, are not major in terms of programming effort but can still lead to significant improvements, see [13] and [14]. A popular modification of the original DE scheme consists of hybridisation with local search, often coordinated by adaptive rules. By means of this hybridisation, modern implementations based on DE but tailored to specific filter design problems for defect detection in paper production are given in [15], [16], and [17]. The most recent studies on this topic use improved/modified versions of DE for classical filter design problems, such as the design of Finite Impulse Response (FIR) and Infinite Impulse Response (IIR) filters, see [18], or modified DE based schemes to address advanced engineering problems such as the design of two-channel quadrature mirror filters with linear phase characteristics, see [19].

By following this trend, in this paper several DE variants are applied to speech segmentation. This is the first stage in a system, now in development, that can predict musculoskeletal articulation of the vocal tract from disordered speech. Articulation prediction could be used within systems that help speech

disorder patients train themselves, in addition to the help they receive from professional therapists. The present study follows on from an earlier study where a library of 3-dimensional representations of vocal articulation were demonstrated as a useful training resource [20]. With articulation prediction it is possible to animate these representations precisely according to the detailed nature of the disorder.

For a training process to be effective the predicted articulation should respond to small changes in the disordered utterance so the patient can know whether they are progressing in a useful way. Therefore the speech analysis component should be sensitive to small changes, and also able to characterise the articulations accurately across a wide variety of voice types. By contrast speech recognition systems are required only to find the most likely text matching a sound, not to evaluate the quality of utterance. Speech recognition is aided by the context present in a string of sounds. In the proposed disordered speech training system the patient is requested to speak only short utterances. Currently we are considering sounds of the form vowel - consonant - vowel. The leading vowel sound is added to make it easier to pronounce the following part. In order to extract structured information from the sound the utterance is first segmented by isolating the unvoiced and voiced sections of the final consonant - vowel, and also identifying the point where the vowel spectrum becomes steady. To evaluate an automated segmentation it is compared with a segmentation prepared by hand that is considered optimal. DE is used to optimise over the parameter spaces of the segmentation methods. A variety of derived signals are used in the segmentation process: Performing a good segmentation is not straightforward, hence the application of DE.

The remainder of this article is organized in the following way. Section II describes the speech model and the signal processing algorithms for the extraction of the speech features that are relevant to this study. These features are used for the identification of the three sections of the disordered speech i.e silence, unvoiced and steady state. Finally, Section II states the optimization problems investigated in this paper. Section III briefly illustrates the optimization algorithms considered in this study. Section IV displays the numerical result obtained by the various algorithmic frameworks considered in this paper. Finally, section V summarizes the findings of this study and gives the conclusions.

II. Speech Features

The three sections of the disordered speech are identified using four features extracted from the speech. These features are: energy, zero crossing count rate (ZCCR), linear predictive coding (LPC) coefficients and autocorrelation coefficients at unit sample delay [21]. Since the steady states of interest are those within certain voiced sections of the disordered speech, some of the features that characterizes normal voiced speech were exploited for identification of the steady states.

A. DC removal and pre-emphasis

Prior to the analysis for the extraction of any of these features from the speech, the speech signal is preprocessed. the DC offset was removed by subtracting the mean of the speech signal from each of the speech samples as shown in (2). The DC offset removal is vital for correct determination of ZCCR of each frame. The DC offset removal is done over the entire speech signal not frame by frame. After the DC removal, the high frequency components of the speech were emphasized using a high pass finite impulse response (FIR) filter shown in (3). The high frequency components were emphasized (bursted) in order to compensate for the de-emphasis (attenuation) of the high frequency components that took place in the glottis during voiced speech production [22]. The pre-emphasis flatten the speech signal spectrally thus reducing the large dynamic range of the speech spectrum by adding a zero to the spectrum. The flattening allow LPC to model all formants equally well.

$$Mean = \frac{1}{N}\sum_{n=0}^{N-1} x(n) \tag{1}$$

$$s(n) = x(n) - Mean, \quad n = 0, 1, 2....N-1 \tag{2}$$

where: x is speech with DC offset, s speech without DC offset and N is the number of samples of the speech signal.

$$H(z) = 1 - 0.95z^{-1} \tag{3}$$

B. Short time energy

The short time energy is the energy computed for each frame using (4). This quantity is defined as the sum of the square of the speech samples in a frame [23] while the short time power is the average of the short time energy (5). The energy of unvoiced speech is greater than energy of silence but less than the energy of steady states. The frame size (window) in this research is 10ms for a sampling frequency of 16 MHz.

$$E_i = \sum_{n=0}^{L-1} s_i(n)^2 \tag{4}$$

$$P_i = \frac{1}{L}\sum_{n=0}^{L-1} s_i(n)^2 \tag{5}$$

Where P_i and E_i are the short time power and energy of frame i respectively, s_i is speech samples of frame i and L is the number of samples per frame.

C. Zero crossing count rate

The zero crossing count rate (ZCCR) is the number of times the speech signal changes sign. Is the measure of the frequency at which the energy is concentrated in the signal spectrum. The ZCCR is computed frame by frame hence it is referred to as short time ZCCR. Voiced speech is produced due to excitation or vibration of the vocal tract by periodic flow of air at the glottis and often have low ZCCR compare to unvoiced speech [24]. Silence has lower ZCCR than unvoiced but quite comparable to voiced speech. The ZCCR is calculated using (6) after removing the DC offset [25].

$$ZCCR_i = \sum_{n=0}^{L-2} \frac{|sign(s_i(n)) - sign(s_i(n+1))|}{2} \tag{6}$$

Where $ZCCR_i$ = ZCCR of frame i.

D. Normalized autocorrelation coefficient at unit sample delay

This is the measure of the correlation between two consecutive or adjacent speech samples. Since voiced speech is characterized with high concentration of low frequency energy, adjacent samples of voiced speech signal are highly correlated, thus this parameter is close to 1 for voiced speech. The concentration of high frequency energy in unvoiced speech, makes adjacent samples uncorrelated, hence the normalized unit delay autocorrelation is close to zero or negative [26]. The normalized autocorrelation at unit sample delay is given by (7).

$$NC_i = \frac{\sum_{n=1}^{L} s_i(n)s_i(n-1)}{\sqrt{\sum_{n=1}^{L} s_i(n)^2 \sum_{n=0}^{L-1} s_i(n)^2}} \tag{7}$$

where NC_i is the normalized autocorrelation coefficient of frame i at unit sample delay.

E. Steady states detection

Steady states refer to the sections of the disordered speech just after the unvoiced section in which the average rate of change of the formants is fairly constant within a given threshold for at least 0.08s (8 frames). In order to locate the steady states, each speech frame is windowed using the Hamming window to reduce the spectral leakage due to discontinuities and to enhance the convergence of the Durbin's algorithm for computation of the LPC coefficients [27]. LPC was used to obtain 13 LPC coefficients for each windowed frame. LPC model the vocal tract as an all poles filter of which the formants frequencies can be inferred from the LPC coefficients. The first, second and third formants frequencies were obtained from the resonance frequency of the vocal tract filter model. Since spectral transition plays an important role in this research, we obtain the first derivative of the formants frequencies and compute the sum of the absolute values of the derivatives. The derivatives obtained directly from the formants frequencies are quite noisy, to filter the noise, the derivatives were obtained via polynomial approximation. A second order polynomial is fit into the trajectory of each of the three formants, the derivative is obtained using (8). To set the decision metric, the absolute sum of the formant derivative is normalized, and the weighted mean and standard deviation were used to set the threshold as depicted by (9)

$$\delta(n,m) = \frac{\sum_{p=-q}^{q} f(n,m+p)p}{\sum_{p=-q}^{q} p^2} \tag{8}$$

$$\top = \alpha\mu \ + \ \beta\sigma^r \tag{9}$$

Where: n = 1, 2 and 3 (number of formants used), m = 1,2,3,4....Number of frames, δ is the derivative, q is set to 3 in this research which gives a good approximation and not too long delay. The fitting width=$2q+1$ introduce q frames of delay. $\top$ is the decision threshold while μ and σ are the mean and standard deviation of the normalized formant derivative. The weights α, β, and r have been obtained by using multiple variants of DE algorithms as discussed in section III.

F. Voice activity detection (VAD)

Silence refer to regions without voice activity. To isolate silence regions from voice activity sections (unvoiced, voiced), we use two speech features i.e the short time power (5) and short time average ZCCR (10). Since we don't have prior knowledge of signal to noise ratio of the various speech signals, we consider the first ten frames (0.1s) as noise. The mean and variance of equation (11) were computed for the first ten frames which are used to set the decision threshold as depicted in 12. If the value obtain from equation (11) for any frame falls below the threshold, the frame will be considered as silence or noise.

$$AZCCR_i = \frac{1}{L}\sum_{n=0}^{L-2} \frac{|sign(s_i(n)) - sign(s_i(n+1))|}{2} \tag{10}$$

$$VAD_i = P_i(1 \ - \ AZZCR_i)\varphi \tag{11}$$

$$S_T = a\eta \ + \ b\rho^c \tag{12}$$

where: P_i and $AZCCR_i$ are the short time power and average ZCCR of frame i respectively, L is number of samples per frame while φ is a constant set at 1000 just to avoid having very small value for VAD_i. S_T is the silence threshold, η and ρ are mean and variance of equation (11) for the first ten frames. The weights a, b and c are obtained using multiple variants of DE algorithms as discussed in section III.

G. Unvoiced detection

Three features are used for unvoiced detection, these are: autocorrelation at unit sample delay, energy and ZCCR. There are three independent thresholds, one for each of the features. A speech segment is considered as unvoiced if it satisfy the three constrains set by the thresholds as described in Fig. 1. The thresholds are set by weighting the long time information of these three features as shown in (13), (14) and (15). The average energy, ZCCR and unit delay autocorrelation (short time information) of the speech segment to be classified are compared with the thresholds for decision making.

$$A_T = \mu_a \ + \ \gamma\varrho^{\xi} \tag{13}$$

$$Z_T = \tau\mu_z \tag{14}$$

$$E_T = \kappa\mu_e \tag{15}$$

Where μ_a and ϱ are the mean and standard deviation of unit sample delay autocorrelation of the speech signal obtain from (7). μ_z and μ_e are the mean of ZCCR and energy respectively. A_T, Z_T and E_T are the thresholds for unit delay correlation, ZCCR and energy respectively. The weights τ, κ, γ and ξ are obtained using multiple variants of DE algorithms as discussed in section III.

NC_i is the normalized autocorrelation coefficient at unit sample delay of frame i obtain using (7). For the other thresholds (Z_T and E_T) to be considered, the NC of the first three consecutive frames (30ms) of the disordered speech must be $\leq A_T$ to mark the beginning of the unvoiced segment. This choice gives a more promising results after several trials and observations seen from the plots of NC of the disordered speeches. After this, the NC of two consecutive frames NC_i and NC_{i+1} will be compared with A_T and if at least one

```
1: i = 1
2: while i <(Number of frames)−1 do
3:    First_Frame = i
4:    if NC_i ≤ A_T and NC_{i+1} ≤ A_T and NC_{i+2} ≤ A_T
      then
5:       while NC_i ≤ A_T or NC_{i+1} ≤ A_T do
6:          i = i + 1
7:       end while
8:       Last_Frame = i
9:       ZMean = Mean of ZCCR of frames from
         First_Frame to Last_Frame
10:      EMean = Mean of energies of frames from
         First_Frame to Last_Frame
11:      if ZMean ≥ Z_T and EMean ≥ E_T then
12:         Frames from First_Frame to Last_Frame are
            unvoiced segment
13:      else
14:         Frames from First_Frame to Last_Frame are
            Not unvoiced segment
15:      end if
16:   else
17:      i = i + 1
18:   end if
19: end while
```

Fig. 1. Unvoiced segments detection decision algorithms

of them is $\leq A_T$, frame i will be included in the segment list otherwise the segment list is terminated. Finally, The thresholds Z_T and E_T are applied on the segment to qualify it as unvoiced or not.

The optimization problem is the minimization of the Average Percentage Error (APE) of misclassification of the three sections (silence, unvoiced and steady state) of the disordered speech by proper setting of a well generalized and robust thresholds. The DE algorithms are used to tune the parameters α, β, and r for steady state detection, a, b and c for silence and τ, κ, γ and ξ for unvoiced. The tuning aims at minimizing (16) for each of the section which is our objective function. The problem is modelled as three independent objective functions optimisation problem. The APE consists of the sum of misdetection and false alarm given by (16). Each variable in the dimension has a different decision space. For steady state detection, the decision space of α, β, and r are [0.1 1], [0.01 0.3] and [0.01 0.6] respectively, for silence detection a, b and c are constrained within [0.04 1], [0.07 8] and [0.02 1] respectively while for unvoiced detection, τ, κ, γ and ξ are constrained within [0.25 3], [0.01 0.1], [0.2 1] and [1.2 7]. respectively.

$$APE_i = \frac{M_i + Fa_i}{N_f} \tag{16}$$

where APE_i is for $i = 1$ silence, for $i = 2$ unvoiced and for $i = 3$ steady state. M_i is the total number of frames belonging to i which were not detected (miss detection) i.e classified as not belonging to i, Fa_i is the number of frames which do not belong to i but were classified as belonging to i (false alarm). N_f is the total number of frames. The total number of frames used in this study is 3690 obtained from five spoken disordered speech signal using a frame size of 10ms.

The problem studied in this paper consists of minimizing APE_i, $\forall i$.

```
1: Generate an initial population of Np individuals
2: Evaluate fitness of each solution in population
   Np
3: while termination condition is not met do
4:    for each x_i in Np do
5:       Generate provisional offspring x'_off by
         mutation
6:       Generate offspring x_off by crossover
7:       Evaluate fitness of x_off
8:       Make a note whether x_i or x_off has a
         better performance
9:    end for
10:   for each x_i in Np do
11:      Perform all the replacements by choosing
         the best between parent offspring
12:   end for
13: end while
```

Fig. 2. Differential Evolution

III. Optimization Algorithms

This section briefly describes the implementation details of the optimization algorithms under consideration used to address the three optimization problems above. In order to clarify the notation, let us indicate with x the generic vector of the function $APE_i(x)$. The general DE scheme, see [28], characterized by a population that is progressively perturbed and updated, is summarized in the Fig. 2.

In this study, the following DE variants have been considered.

A. DE/rand/1/bin and DE/rand/1/exp

The two basic DE variants, see [28] namely DE/rand/1/bin and DE/rand/1/exp have been used. Both these variants generate the provisional offspring by means of the DE/rand/1 mutation, that is the following formula:

$$x'_{off} = x_t + F \cdot (x_r - x_s) \tag{17}$$

where x_t, x_r and x_s are three randomly selected individuals of the population such that $x_t \neq x_r \neq x_s$. In the bin variant, each variable in x_i is exchanged with provisional offspring with probability Cr, which is called *crossover rate*. More precisely, binomial crossover is done by following formula:

$$x_{off}[j] = \begin{cases} x_{i,off}[j] & \text{if} \quad (\mathcal{U}(0,1) \leq Cr \quad \text{or} \quad j = j_{rand}) \\ x_i[j] & \text{otherwise} \end{cases} \tag{18}$$

where j is index of gene being selected and $\mathcal{U}(0,1)$ is a uniformly distributed random number. The bin stands for binomial as that would be the distribution shape from which the points are sampled.

The exp variant makes use of the so-called exponential crossover. This operator copies a section of the genes of the parent into the offspring until a sequence of random numbers is smaller than a threshold Cr. Implementation details are reported in Fig. 3.

```
x_off = x_i
generate j = round(n · rand(0,1))
x_off[j] = x_{i,off}[j]
k = 1
while rand(0,1) ≤ Cr AND k < n do
    x_off[j] = x_{i,off}[j]
    j = j + 1
    if j == n then
        j = 1
    end if
    k = k + 1
end while
```

Fig. 3. Exponential crossover

```
x_old = x_0
for i = 1 : n do
    δ[i] = α (x^U[i] − x^L[i])  (α ∈ ]0,1])
    i ← i + 1
end for
while stop condition not met do
    for i = 1 : n do
        x_trial[i] = x_s[i] − δ[i]
        if f(x_trial) ≤ f(x_s) then
            x_s = x_trial
        else
            x_trial[i] = x_s[i] + δ/2
            if f(x_trial) ≤ f(x_s) then
                x_s ← x_trial
            end if
        end if
        i ← i + 1
    end for
    if f(x_s) == f(x_old) then
        δ ← δ/2
    end if
end while
Output x_s
```

Fig. 4. Short Distance Exploration

B. Super-fit DE

One of the most efficient ways to hybridise a DE scheme with a local search is the application of the super-fit logic, see [16]. This idea consists of applying the local search to a solution and injecting the improved solution in the initial DE population. This logic appeared promising in various contexts, displaying a good performance on test functions as well as control engineering and image processing problems, see [16]. Furthermore, the generation of a super-fit individual appeared beneficial also for compact DE structures, see [29]. When an individual with high performance is injected into the population, the DE scheme is led by it and quickly improves upon the super-fit, thus generating a new best individual.

In this paper, the initial local search is performed by the so-called Short Distance Exploration or simply S algorithm, see [30], [31], and [32]. The S algorithm is a simple greedy local search that performs moves along the axes and halves its radius when it is unable to detect a better solution. For the sake of clarity the working principles of S are briefly illustrated in Fig. 4.

C. jDE

The jDE scheme is a popular way to enhance upon DE performance with a very modest programming effort, see [33]. The jDE algorithm employs an ingenious to enhance the pool of DE search moves, simply including a certain degree of randomization into the original framework, see [2]. In jDE, the values of mutation and crossover are encoded within each individual. For example the generic individual x_i will be composed of

$$x_i = (x_i[1], x_i[2], \ldots, x_i[n], F_i, Cr_i)\,. \tag{19}$$

Hence, the offspring generation is generated, for each individual, with the parameters F_i and Cr_i belonging to its parent. In addition these parameters are periodically refreshed on the basis of the following randomized criterion:

$$F_i = \begin{cases} F_l + F_u rand_1, & \text{if } rand_2 < \tau_1 \\ F_i, & \text{otherwise} \end{cases} \tag{20}$$

$$CR_i = \begin{cases} rand_3, & \text{if } rand_4 < \tau_2 \\ CR_i, & \text{otherwise} \end{cases} \tag{21}$$

where $rand_j$, $j \in \{1,2,3,4\}$, are uniform pseudo-random values between 0 and 1; τ_1 and τ_2 are constant values which represent the probabilities that parameters are updated, F_l and F_u are constant values which represent the minimum value that F could take and the maximum variable contribution to F, respectively.

D. Comprehensive Learning Particle Swarm Optimization

In order to have for reference comparison an algorithm that is not based on a DE framework, the Comprehensive Learning Particle Swarm Optimization (CLPSO), see [34], has also been taken into account. This swarm intelligence algorithm perturbs its particle by updating the velocity by means of the following formula:

$$v_k^{t+1} = \phi_1 v_k^t + \phi_2 \mathbf{U}\left(x_{lb-f}^t - x_i^t\right) \tag{22}$$

where $\mathbf{U}$ is an $n \times n$ random matrix and $\mathbf{x_{lb-f}^t}$ is a heterogeneous vector composed, along each design variable, of local best design variables. In order to determine (select) each $x_{lb-f}^t[i]$ value, a probabilistic procedure is implemented. For each design variable, a random number is generated and compared with a threshold value P_c. If it is higher then, for the corresponding design variable, the local best particle is followed. If it is lower then two local best particles are randomly selected from the swarm and their fitness value compared. The winning particle donates the corresponding design variable to x_{lb-f}^t. Regarding the setting of P_c, in [34] an empirical rule has been proposed. For each particle i, a P_{c-i} value is calculated by the following expression:

$$P_{c-i} = 0.05 + 0.45\frac{e^{\frac{10(i-1)}{SS-1}} - 1}{(e^{10} - 1)} \tag{23}$$

where SS is the swarm size (number of particles). Finally, when the velocity has been calculated, the position of the particle id updated by the standard PSO formula:

$$x_k^{t+1} = x_k^t + v_k^{t+1}. \tag{24}$$

IV. RESULTS

In order to minimize the three APE_i functions, the algorithms listed above have been run for 30 independent runs. Each run has been continued with a budget of 4000 fitness evaluations for each objective function APE_i. The population size for all the algorithms included in this study has been set to 20 individuals. The parameters used by each algorithm have been set after a manual tuning. The following parameters have been used.

- DE/rand/1/bin has been run with $F = 0.3$ and $Cr = 0.1$
- DE/rand/1/exp has been run with $F = 0.3$ and $Cr = 0.1$
- the super-fit implementations of the two DE variants above, namely S/DE/bin and S/DE/exp, have been run with the same parameter setting of the original DE versions but allowing 20% of the total budget to the initial local search and initial radius rate $\alpha = 0.4$.
- jDE/rand/1/bin and jDE/rand/1/exp have been run with $F_l = 0.1$ and $F_u = 0.9$, $\tau_1 = 0.1$ and $\tau_2 = 0.1$.
- CLPSO has been run with $C_1 = 0.3$, $C_2 = 0.4$, $V_{max} = 0.3 * (Max(Range) - Min(Range))$, $V_{min} = -V_{max}$, $W_{max} = Max(Range)$, $W_{min} = Min(Range)$, $Range$ is the decision space of the variables as described in section II-G.

Table I shows the numerical results in terms of error for silence, unvoiced and steady state detection. Obviously, a null error would mean that the detection of these features was perfectly performed. The final results of each algorithm have been averaged over the available 30 runs. The related standard deviation values are also shown. Since the jDE/rand/1/bin displayed the best performance, it has been taken as a reference for the Wilcoxon test [35]. A "+" indicates that jDE/rand/1/bin significantly outperforms the other algorithm while "=" indicates that the there is no outperformance. There are no cases in this experimental setup where jDE/rand/1/bin was outperformed. It is interesting to note that the jDE/rand/1/bin and some of the DE variants achieve many different solutions that have the same best fitness function performance.

TABLE I. TEST RESULTS WITH JDE BIN AS REFERENCE

Algorithms	Silence	Unvoiced	Steady State
	APE STD	APE STD	APE STD
DE/rand/1/bin	0.0794 (0.0002+)	0.0077 (0.0000=)	0.1263 (0.0000=)
DE/rand/1/exp	0.0802 (0.0004+)	0.0077 (0.0000=)	0.1266 (0.0002+)
S/DE/bin	0.0794 (0.0002+)	0.0077 (0.0000=)	0.1263 (0.0001=)
S/DE/exp	0.0802 (0.0004+)	0.0077 (0.0000=)	0.1265 (0.0002+)
jDE/rand/1/bin	**0.0793 (0.0000)**	**0.0077 (0.0000)**	**0.1263 (0.0000)**
jDE/rand/1/exp	0.0802 (0.0003+)	0.0077 (0.0000=)	0.1266 (0.0002+)
CLPSO	0.0817 (0.0012+)	0.0084 (0.0011+)	0.1264 (0.0001+)

As shown in Table I, jDErand/1/bin outperforms all the other algorithms in silence detection. For unvoiced detection, the performance of jDErand/1/bin is the same as other DE variants implemented, with CLPSO having the least performance. For steady stead detection, jDErand/1/bin perform equally well just as the other DE variants with binomial crossover but outperform CLPSO. For this problem, binomial crossover outperforms its exponential crossover variant for the same algorithm with reference to silence and steady state detection; but the two crossovers (binomial and exponential), perform equally well for unvoiced detection. On the basis of our tests and interpretation of the problem features, the unvoiced detection was a fairly easy problem with a strong basin of attraction. The other two optimization problems appeared more challenging. From Table I, the APE for silence detection using DE/rand/1/bin and jDE/rand/1/bin are 0.0794 and 0.0793 respectively. There is a difference between these APEs depending on the number of frames, for few frames the difference is insignificant but for large number of frames (in millions) the difference is significant. Also, these values (0.0794 and 0.0793) are average over 30 runs and the standard deviation for jDE/rand/1/bin is 0.0000 which shows that it converge to the same error (APE) for the 30 independent runs while DE/rand/1/bin have a standard deviation of 0.0002. It must be remarked that the total budget allocated to each algorithm was fairly modest due to the high computational cost of each function call (approx 0.4 seconds on a modern computer). The DE based schemes appeared more efficient for this problem than the CLPSO to quickly achieve a solution with a high performance. According to our observation, in order to achieve better results CLPSO would have required a longer budget.

The best solution detected by jDE/rand/1/bin contains the following parameters. For silence $a = 2.1130$, $b = 0.1231$ and $c = 0.2362$. For unvoiced $\tau = 2.0097$, $\kappa = 0.0224$, $\gamma = 0.2604$, and $\xi = 5.0677$. For steady state $\alpha = 0.1618$, $\beta = 0.0395$, and $r = 0.2490$.

In order to illustrate the physical meaning of this study, the effect of the best solution on a short speech utterance is illustrated in Fig. 5 with reference to a signal used in the database for error identification during the optimization (training set). More specifically, in the first subplot of Fig. 5 the silence detections are highlighted. It can easily be observed that the proposed system efficiently distinguishes from silence to non-silence sections of the signal. The second subplot shows the detection of unvoiced sections (dashed line) and the steady states (continuous line). Clearly the unvoiced sections of the signal are identified. It can be observed that, as expected, the areas between dashed beams are aperiodic. Also the steady state detection reliably identifies the end of the informative part of the speech. The third subplot shows the normalized trajectory of the average of the absolute rate of change (first derivative) of the first three formants obtained from (8) (thin line). The threshold that set the decision metric for steady states obtained using the optimization best solution is given by the thick horizontal line. The normalized trajectory and the threshold are used to identify the steady states, marked with dash lines. The positions of the steady states identified on the normalized formant derivative are also marked on the speech waveform as depicted on the second subplot with thick vertical lines. The optimized best solution does well on the training speech as well as the validation speech.

Fig. 6 shows how the tuned system can perform an efficient segmentation on a speech utterance that was not in the training set (validation set). It can be seen that although the speech belongs to a different person and is derived from a female voice, unlike the training set that was composed of male voice speeches, the segmentation is reliable and precise. The same

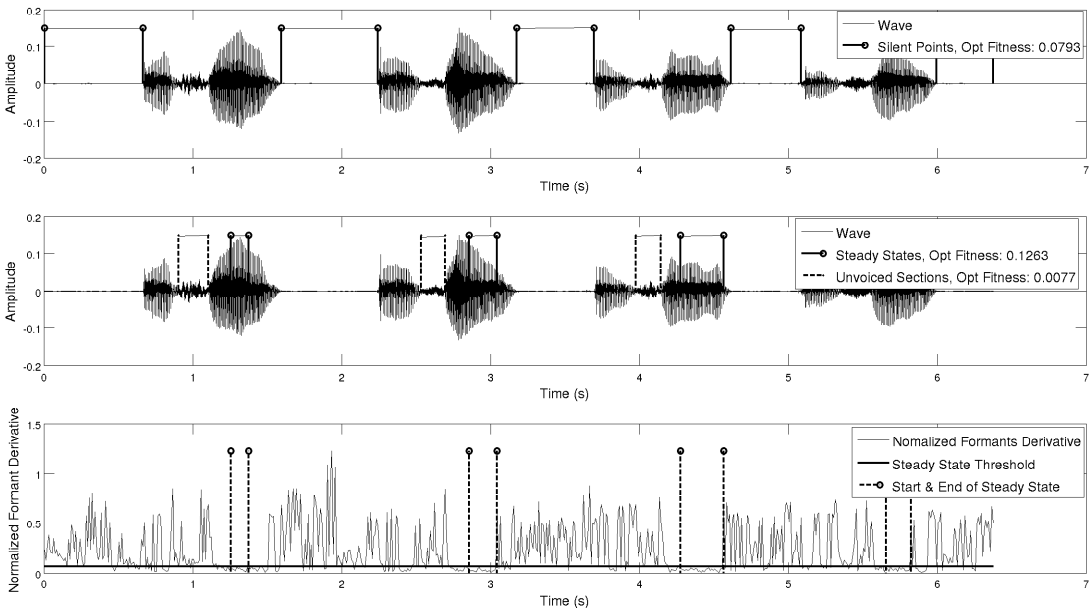

Fig. 5. Speech utterance belonging to the training set.

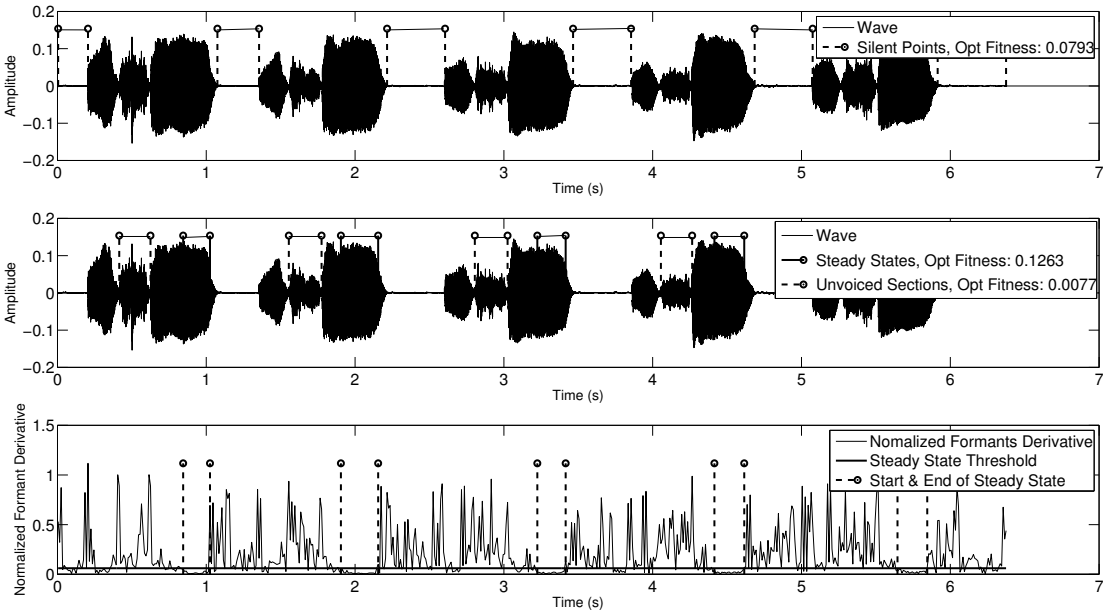

Fig. 6. Speech utterance belonging to the validation set.

best solution would return a high fitness score also in this case.

V. Conclusion

This paper proposes a segmentation system, tuned by a Differential Evolution framework, for speech utterances. This segmentation system is part of a platform under development whose role is to support people affected by speech impairments. The proposed segmentation system detects silence, unvoiced, and steady state sections of the speech, thus isolating the most informative part of the signal. This operation corresponds to the solution of three optimization problems. In order to tackle this problem six implementations of Differential Evolutions have been tested and compared. In addition, as a

reference a popular swarm intelligence algorithm has also be included in this study. Numerical results show that Differential Evolution can efficiently solve this problem, especially when it employs binomial crossover and a controlled randomization of its parameters. The resulting system appears efficient and robust also when speech utterances not considered in the training phase are examined.

ACKNOWLEDGMENT

This work is supported by the Higher Education Innovation Fund at De Montfort University. This research is supported by the Academy of Finland, Akatemiatutkija 130600, Algorithmic design issues in Memetic Computing.

REFERENCES

[1] S. Das and P. Suganthan, "Differential Evolution: A Survey of the State-of-the-Art," *Evolutionary Computation, IEEE Transactions on*, vol. 15, no. 1, pp. 4–31, feb. 2011.

[2] F. Neri and V. Tirronen, "Recent Advances in Differential Evolution: A Review and Experimental Analysis," *Artificial Intelligence Review*, vol. 33, no. 1–2, pp. 61–106, 2010.

[3] N. Salvatore, A. Caponio, F. Neri, S. Stasi, and G. L. Cascella, "Optimization of Delayed-State Kalman Filter-based Algorithm via Differential Evolution for Sensorless Control of Induction Motors," *IEEE Transactions on Industrial Electronics*, vol. 57, no. 1, pp. 385–394, 2010.

[4] I. D. Falco, D. Maisto, U. Scafuri, E. Tarantino, and A. D. Cioppa, "Distributed Differential Evolution for the Registration of Remotely Sensed Images," in *Proceedings of the IEEE Euromicro International Conference on Parallel, Distributed and Network-Based Processing*, 2007, pp. 358–362.

[5] R. Storn, "Differential evolution design of an IIR-filter," in *Proceedings of IEEE International Conference on Evolutionary Computation*, 1996, pp. 268–273.

[6] N. Karaboga and B. Cetinkaya, "Performance Comparison of Genetic and Differential Evolution Algorithms for Digital FIR Filter Design," in *Advances in Information Systems LNCS*, ser. Lecture Notes in Computer Science, vol. 3261. Springer, 2004, pp. 482–488.

[7] N. Karaboga, "Digital IIR Filter Design Using Differential Evolution Algorithm," *EURASIP Journal on Applied Signal Processing*, vol. 8, pp. 1269–1276, 2005.

[8] N. Karaboga and B. Cetinkaya, "Design of Digital FIR Filters Using Differential Evolution Algorithm," *Circuits, Systems, and Signal Processing*, vol. 25, no. 5, pp. 649–660, October 2006.

[9] R. Storn, "Designing nonstandard filters with differential evolution," *IEEE Signal Processing Magazine*, vol. 22, no. 1, pp. 103–106, 2005.

[10] M. Weber, V. Tirronen, and F. Neri, "Scale Factor Inheritance Mechanism in Distributed Differential Evolution," *Soft Computing - A Fusion of Foundations, Methodologies and Applications*, vol. 14, no. 11, pp. 1187–1207, 2010.

[11] M. Weber, F. Neri, and V. Tirronen, "A Study on Scale Factor in Distributed Differential Evolution," *Information Sciences*, vol. 181, no. 12, pp. 2488–2511, 2011.

[12] F. Neri, G. Iacca, and E. Mininno, "Disturbed Exploitation compact Differential Evolution for Limited Memory Optimization Problems," *Information Sciences*, vol. 181, no. 12, pp. 2469–2487, 2011.

[13] S. Rahnamayan, H. R. Tizhoosh, and M. M. Salama, "Opposition-Based Differential Evolution," *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 1, pp. 64–79, 2008.

[14] A. Zamuda, J. Brest, B. Bošković, and V. Žumer, "Large Scale Global Optimization Using Differential Evolution With Self-adaptation and Cooperative Co-evolution," in *Proceedings of the IEEE World Congress on Computational Intelligence*, 2008, pp. 3719–3726.

[15] V. Tirronen, F. Neri, T. Kärkkäinen, K. Majava, and T. Rossi, "An enhanced memetic differential evolution in filter design for defect detection in paper production," *Evolutionary Computation*, vol. 16, pp. 529–555, 2008.

[16] A. Caponio, F. Neri, and V. Tirronen, "Super-fit control adaptation in memetic differential evolution frameworks," *Soft Computing - A Fusion of Foundations, Methodologies and Applications*, vol. 13, pp. 811–831, 2009.

[17] F. Neri and V. Tirronen, "Memetic differential evolution frameworks in filter design for defect detection in paper production," in *Evolutionary Image Analysis and Signal Processing*, ser. Studies in Computational Intelligence, S. Cagnoni, Ed. Springer Berlin Heidelberg, 2009, vol. 213, pp. 113–131.

[18] G. Liu, Y. Li, and G. He, "Design of digital fir filters using differential evolution algorithm based on reserved genes," in *Evolutionary Computation (CEC), 2010 IEEE Congress on*, 2010, pp. 1–7.

[19] P. Ghosh, S. Das, and H. Zafar, "Adaptive-differential-evolution-based design of two-channel quadrature mirror filter banks for sub-band coding and data transmission," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 42, no. 6, pp. 1613–1623, 2012.

[20] H. N. Cornelius, P. and R. Kaleem, "Sensory articulation speech system: Sassy - a 3d animation based therapeutic application for motor speech disorders," *Journal of Assistive Technologies*, vol. 5, no. 3, pp. 123–130, 2011.

[21] K. Dash, D. Padhi, B. Panda, and S. Mohanty, "Speaker identification using mel frequency cepstralcoefficient and bpnn," *International Journal of Advanced Research in Computer Science and Software Engineering Research Paper*, vol. 2, April 2012.

[22] D. Li and O. Douglas, *Speech Processing: A Dynamic and Optimization-Oriented Approach (Signal Processing and Communications)*. New York, NY, U.S.A: Marcel Dekker Inc., 2003.

[23] S. Chu, S. Narayanan, and C. J. Kuo, "Environmental sound recognition with timefrequency audio features," in *IEEE TRANSACTIONS ON AUDIO, SPEECH, AND LANGUAGE PROCESSING*, vol. 17, August 2009.

[24] M. Radmard, M. Hadavi, and M. M. Nayebi, "A new method of voiced/unvoiced classification based on clustering," *Journal of Signal and Information Processing*, pp. 336–347, 2011.

[25] K. R. AidaZade, C. Ardil, and S. Rustamov, "Investigation of combined use of mfcc and lpc features in speech recognition systems," *International Journal of Computer, Information Science and Engineering*, vol. 2, 2008.

[26] F. QI, C. BAO, and Y. LIU, "A novel two-step svm classifier for voiced/unvoiced/silence classification of speech," 2004.

[27] H. Perez-Meana, *Advances in Audio and Speech Signal Processing: Technologies and Applications*, 701 E. Chocolate Avenue, Hershey PA 17033, U.S.A, 2007.

[28] K. V. Price, R. Storn, and J. Lampinen, *Differential Evolution: A Practical Approach to Global Optimization*. Springer, 2005.

[29] G. Iacca, R. Mallipeddi, E. Mininno, F. Neri, and P. N. Suganthan, "Super-fit and Population Size Reduction Mechanisms in Compact Differential Evolution," in *Proceedings of IEEE Symposium on Memetic Computing*, 2011, pp. 21–28.

[30] F. Caraffini, F. Neri, G. Iacca, and A. Mol, "Parallel memetic structures," *Information Sciences*, vol. 227, no. 0, pp. 60 – 82, 2013.

[31] F. Caraffini, F. Neri, B. Passow, and G. Iacca, "Re-sampled inheritance search: High performance despite the simplicity," *Soft Computing*, vol. 17, no. 12, pp. 2235–2256, 2014.

[32] F. Caraffini, F. Neri, and L. Picinali, "An analysis on separability for memetic computing automatic design," *Information Sciences*, vol. 265, pp. 1–22, 2014.

[33] J. Brest, S. Greiner, B. Bošković, M. Mernik, and V. Žumer, "Self-Adapting Control Parameters in Differential Evolution: A Comparative Study on Numerical Benchmark Problems," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 6, pp. 646–657, 2006.

[34] J. J. Liang, A. K. Qin, P. N. Suganthan, and S. Baskar, "Comprehensive learning particle swarm optimizer for global optimization of multimodal functions," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 3, pp. 281–295, 2006.

[35] F. Wilcoxon, "Individual comparisons by ranking methods," *Biometrics Bulletin*, vol. 1, no. 6, pp. 80–83, 1945.