

A Constraint-Based Evolutionary Learning Approach to the Expectation Maximization for Optimal Estimation of the Hidden Markov Model for Speech Signal Modeling

Shamsul Huda, John Yearwood, and Roberto Togneri, *Senior Member, IEEE*

Abstract—This paper attempts to overcome the tendency of the expectation-maximization (EM) algorithm to locate a local rather than global maximum when applied to estimate the hidden Markov model (HMM) parameters in speech signal modeling. We propose a hybrid algorithm for estimation of the HMM in automatic speech recognition (ASR) using a constraint-based evolutionary algorithm (EA) and EM, the CEL-EM. The novelty of our hybrid algorithm (CEL-EM) is that it is applicable for estimation of the constraint-based models with many constraints and large numbers of parameters (which use EM) like HMM. Two constraint-based versions of the CEL-EM with different fusion strategies have been proposed using a constraint-based EA and the EM for better estimation of HMM in ASR. The first one uses a traditional constraint-handling mechanism of EA. The other version transforms a constrained optimization problem into an unconstrained problem using Lagrange multipliers. Fusion strategies for the CEL-EM use a staged-fusion approach where EM has been plugged with the EA periodically after the execution of EA for a specific period of time to maintain the global sampling capabilities of EA in the hybrid algorithm. A variable initialization approach (VIA) has been proposed using a variable segmentation to provide a better initialization for EA in the CEL-EM. Experimental results on the TIMIT speech corpus show that CEL-EM obtains higher recognition accuracies than the traditional EM algorithm as well as a top-standard EM (VIA-EM, constructed by applying the VIA to EM).

Index Terms—Constraint-based evolutionary algorithm (EA), expectation maximization (EM), fusion strategies, hidden Markov model (HMM), hybrid algorithm, Lagrange multiplier (LM), signal modeling and classification, speech recognition.

I. INTRODUCTION

THE HIDDEN Markov model (HMM) is the most successful and widely used statistical modeling technique for

speech signal modeling in automatic speech recognition (ASR) [1], [2] and also signal classification. This is because the HMM has a powerful ability to model the temporal nature of speech signals statistically as well as the ability to represent arbitrarily complex probability density functions. In a Bayesian classification scenario (for signal classification and recognition), a model for a speech signal provides the mapping from the features of the instances of a particular phoneme class (a basic theoretical unit of speech sound) to the probabilistic parameterized model (HMM). The HMM provides the posterior probability of the speech signal given the phoneme classes/signal classes in signal classification. Therefore, the success of the recognizer/classifier for speech depends heavily on how precisely the estimated HMM can represent the underlying phoneme or signal classes in speech data. In a real recognition task, feature vectors from different instances of the same phoneme vary largely due to variations in speakers, variation in the emotion of the speaker, and changes in the environment. Therefore, it is very difficult to find an appropriate estimation for the parameters of the HMM that can precisely represent all the phonemes in the training speech data.

The standard method to estimate the parameters of HMM-based acoustic models in ASR systems is the Baum-Welch (expectation maximization, EM) [1]–[3] algorithm. The Baum-Welch (EM) [1]–[3] estimation approach is attractive because it can approximate the underlying distribution from the set of observed data which has some missing or hidden components. In particular, while modeling a speech signal using HMM, features from the speech signal are observed, but the state sequence of HMM which generated the signal remains hidden. In this case, optimization of the likelihood function is usually analytically intractable [3]. However, the EM algorithm simplifies the likelihood function by considering some additional variables for hidden components of the data and initial values for those variables so that the likelihood function can be optimized. The EM estimates the parameters of HMM in an iterative manner that makes it more computationally efficient and helps to converge fast because EM guarantees an increment in the likelihood function at each iteration [1]–[4] of its estimation procedure.

Unfortunately, the estimation of HMM parameters computed by the Baum-Welch (EM) approach is not always the best [1], [2], and thereby, the use of the model estimated by EM may

Manuscript received January 25, 2008; revised April 29, 2008 and July 18, 2008. First published December 9, 2008; current version published January 15, 2009. This paper was recommended by Associate Editor Y. Soon.

S. Huda and J. Yearwood are with the Center for Informatics and Applied Optimization, University of Ballarat, Ballarat, Vic. 3350, Australia (e-mail: shuda@ballarat.edu.au; shuda9203@yahoo.com; j.yearwood@ballarat.edu.au).

R. Togneri is with the Centre for Intelligent Information Processing System, School of Electrical, Electronic and Computer Engineering, University of Western Australia, Perth, W.A. 6009, Australia (e-mail: roberto@ee.uwa.edu.au).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TSMCB.2008.2004051

lower the recognition accuracy of ASR systems. The reason is that the EM algorithm is strongly dependent on the selection of the initial values of model parameters and is guaranteed to produce a local rather than a global maximum of the likelihood function [1], [2], [4]. This gives a nonoptimized estimation of the parameters of HMM and consequently lowers the recognition accuracy. Two important research questions are therefore as follows: How do we choose initial estimates of the HMM parameters and how can we escape from a local maximum point if EM is found to be stuck there for better estimation for HMM parameters and higher recognition accuracy of the ASR system. In this paper, we focus on these drawbacks of the EM algorithm for the estimation of HMM parameters in signal classification or ASR systems.

A global search method such as an evolutionary algorithm (EA) could be used to avoid the local maximum problem of EM. The EA can explore the search space without using any knowledge about the underlying problem structure and is less likely to be trapped into the local maxima. However, it is well known that EA is inefficient for high-dimensional optimization problems. Application of a hybrid algorithm using EA and local search will be investigated.

Recently, several investigators have applied a hybrid algorithm using neighborhood search such as Tabu Search (TS) [5] and [6] simulated annealing (SA) [7] in combination with EM to overcome the problem of EM for continuous density HMM (CDHMM). The TS requires huge amount of memory to maintain the list of already visited solutions due to the high number of variables in CDHMM. The move attribute-based method could be used to reduce the memory requirement, but this makes TS too restrictive. Empirical studies show that in a high-dimensional search space, a limited number of iterations with restrictive TS make it dependent on initial point. An inappropriate choice of initial point results in a failure to find an optimal solution. Both SA and TS work on single-candidate models. In this context, hybrid algorithm (that combines population-based algorithm (EA) and local search EM) may be more effective. In the hybrids of EA [8]–[13], the probability of choosing an inappropriate initial point is minimized due to the use of a large number of initial points of EA distributed over the whole search space. Therefore, hybrids of EA [8]–[13] can explore the search space more extensively than hybrids of single-candidate-model-based approaches (with SA and TS).

In the literature, the authors in [8]–[10], and [14] have used EA in combination with EM for optimal estimation of a Gaussian mixture model (GMM) in a nonlinear classification problem and unsupervised clustering. These hybrid algorithms in [8]–[10], and [14] ignore the constraints of the GMM and assume equal mixture weights which may fail in many practical situations where the mixture weights of individual mixtures of GMM are not the same. Therefore, these algorithms [8]–[10], [14] cannot be applied on the constraint-based models. When a hybrid of EA and EM is applied on a constraint-based model like HMM, the problem context is changed. HMM combines several GMMs into a single model that constitutes a large number of parameters and mixture constraints aggregated from the GMMs. The HMM also has transition and observation probability constraints for each state. These constraints must be

satisfied while estimating the HMM parameters. When the EM is applied separately, the constraints are automatically satisfied [1]. However, EA is stochastic and can violate the constraints of HMM when applied with the hybrid algorithm. Therefore, the algorithms proposed in [8]–[10], and [14] cannot be applied on the HMM.

Moreover, in the literature [8]–[10], hybrids of EA and EM use a pipelining fusion strategy [15]–[17] with the Lamarckian viewpoint [16]–[18] where EA provides the initial points for EM at every generation of the EA. A pipelining strategy is conceptually similar to segmental-K-means segmentation [19] in the context of varying the EM initial point. However, the optimization problem in EM for HMM is very high-dimensional, and the surface of the optimization function in ASR is very complex, which includes many local maxima [1], [2]. Empirical studies [18], [20] show that a pipelining mechanism is not suitable for high-dimensional functions, since it reduces the global sampling capabilities of EA and cannot maintain the diversity in the population [20] (diversity in the EA population is essential for high-dimensional function optimization [21]).

The power of global sampling capabilities in EA is due to its schema processing capabilities (which represents the hyperplane partition) [22], [23]. In fact, far more hyperplanes are sampled simultaneously than the actual number of chromosomes contained in the population of EA using the implicit parallelism technique of EA. This provides the global search capabilities to the EA [22], [23]. However, the use of pipelining strategy [8], [9], [14] changes the genetic information of the chromosome at each generation of EA that results a loss of statistical information about the hyperplane partition information implicitly contained in the population as well as the inherited schema [15], [18], [24]. This, in turn, decreases the global sampling capabilities of EA [18], [22]–[24]. The disadvantages of pipelining hybrid EA have also been discussed in many papers including those in [15]–[18], [20], and [24].

In this paper, we therefore propose a constraint-based evolutionary learning approach to EM (CEL-EM) that hybridizes a constraint-based EA and the EM for optimal estimation of HMM for ASR systems. The novelty of our hybrid algorithm (CEL-EM) is that it is applicable for estimation of the constraint-based models with many constraints and large numbers of parameters like HMM.

Our contribution also includes the following investigation hitherto unreported in the literature.

- 1) Different constraint-based versions for CEL-EM have been developed and formulated for HMM in the ASR systems to avoid the local maxima problem of EM in the HMM estimation process.
- 2) Combinations of constraint-based versions of and fusion strategies for CEL-EM have also been developed and experimentally verified to find a suitable constraint-based method and fusion strategy for the estimation of HMM in ASR systems.
- 3) In CEL-EM, a variable initialization approach (VIA) has been proposed using a variable segmentation to provide a better initialization for EA and also for EM. Experimental

results of CEL-EM have been compared with a standard EM as well as VIA-EM which has been constructed by applying the VIA to EM.

Two different constraint-based versions of “CEL-EM” have been developed. The first constraint-based version of “CEL-EM” follows a penalization method [25] similar to the traditional constraint-handling technique used in the EA. The second version of “CEL-EM” uses a Lagrange multiplier (LM)-based technique [26], [27] to handle the constraints. The traditional constraint-handling method [25] of EA makes different levels in a particular constraint depending on the values of the constraint. This requires a large number of optimal static penalty coefficients for many HMM constraints which are hard to find and increases the number of parameters to be optimized [25]. Therefore, in the second constraint-handling approach, we have used LMs. LM can be used to transform a constraint-based optimization problem into an unconstraint-based problem [26], [27]. LM also adds some additional parameters to be optimized, but the total number of multipliers in LM is fixed for HMM. The only concern here is to find optimal values for LM. We have proposed an evolutionary approach to find the values for LM. The LM approach has also been found to provide better recognition than the traditional method.

In addition, we have proposed two fusion strategies for “CEL-EM” where Lamarckian evolution [18] is executed periodically after the execution of Darwinian evolution [22], [23], [28] for a specific period of time, making the two stages of the hybrid algorithm a staged fusion [15], [18], [20]. In the staged fusion of CEL-EM, EM is executed periodically after the execution of EA for a specific period of time (a Darwinian evolution [22], [23], [28]), thus utilizing the local knowledge of EM as well as minimizing the loss of hyperplane partition information in the EA and maintaining the global sampling capabilities of EA. The first fusion strategy of CEL-EM uses a simple staged-fusion approach, and the second strategy applies a biased-crossover technique [29], [30] with the staged fusion. The staged-fusion strategy of CEL-EM requires careful choice of evolutionary operators for EA that can produce feasible solutions in the offspring population of EA which, in turn, requires a feasible initial population that is created by problem-specific heuristics [25] (not randomly generated). Therefore, in the CEL-EM, we have proposed a variable segmentation technique to create the initial population for the evolutionary process of CEL-EM. However, the recognition experiments have shown that the second fusion strategy with the LM-based constrained version outperforms all other combination of fusion strategies and constraint-based versions of CEL-EM.

The remainder of this paper is organized as follows. In the next section, a brief description of the HMM, its parameters, and the constraints are discussed. Section III briefly describes the EM algorithm and its problems to estimate the HMM parameters. The CEL-EM with fusion strategies for the estimation of HMM parameters is described in Section IV. Section V discusses the experimental procedure and results. The significance of the results is analyzed in Section VI. Conclusions of this paper are given in the last section.

II. HMM FOR SPEECH SIGNAL MODELING, ITS PARAMETERS, AND CONSTRAINTS

The HMM is a doubly stochastic process. One stochastic process comprises the distribution of observations at each state which is a multimodal Gaussian mixture for a CDHMM. The other stochastic process involves the transitions between the HMM states, which are the transition probabilities $A = a_{ij}$. HMM parameters can be represented as $\{\lambda = c_{jn}, \mu_{jn}, \Sigma_{jn}, a_{ij}\}$, where $i, j = 1, 2, \dots, K$ (K = total number of states), and $n = 1, 2, \dots, M$ (M = total number of mixtures), with c_{jn} = mixture weight, μ_{jn} = mean vector, and Σ_{jn} = covariance for j th state and n th mixture. The probability for the t th observation at the j th state is considered as $b_j(O_t) = \sum_{n=1}^M c_{jn} b_j^n(O_t)$, where $b_j^n(O_t) = (1/\sqrt{(2\pi)^D |\Sigma_{jn}|}) \exp(-1/2(O_t - \mu_{jn})' \Sigma_{jn}^{-1} (O_t - \mu_{jn}))$ and D = dimension of O_t . The constraints of HMM are $\sum_{n=1}^M c_{jn} = 1$, $\sum_{j=1}^K a_{ij} = 1$, and $\sum_{t=1}^T b_j(O_t) = 1$, where T = total number of observations in an instance. In a Bayesian classification scenario, the model for speech signals provides the posterior probability of unknown signal data or phoneme where each signal class is represented by a separate HMM. Therefore, appropriate estimation of HMM parameter is a primary concern in signal classification/phoneme recognition. The maximum likelihood (ML) estimation is the standard method to estimate the parameters of a probabilistic model. The ML estimate of HMM parameters is $\hat{\lambda}_{ML} = \arg \max_{\lambda} \{\log P(O|\lambda)\}$, where $\log P(O|\lambda)$ is the log-likelihood of the observed data. $P(O|\lambda)$ can be written as

$$P(O|\lambda) = P(O, q|\lambda) / P(q|O, \lambda) \quad (1)$$

where q = HMM state sequence. Taking the logarithm of (1)

$$\log [P(O|\lambda)] = \log [P(O, q|\lambda)] - \log [P(q|O, \lambda)]. \quad (2)$$

If we knew the state sequence q , the ML estimate of the HMM parameters $\hat{\lambda}_{ML}$ could be computed by taking the derivative of (2) with respect to λ and then equating the derivative to zero. However, we do not know the state sequence q which generated the observed data O . Therefore, there is no closed form of solution from the derivative, and a direct derivative method of ML estimation will not work to estimate the HMM parameters. This estimation problem for HMM can be solved by the EM [1], [3] algorithm.

III. EM ALGORITHM TO ESTIMATE HMM PARAMETERS AND PROBLEM OF EM

The EM algorithm [1], [3] is an iterative method to solve the estimation problem of HMM parameters. EM simplifies the log-likelihood $L = \log P(O|\lambda)$ of observed data O in terms of the expected value of the log-likelihood $Q(\lambda, \lambda^k) = \sum_{q \in Q} P(q|O, \lambda^k) \log [P(O, q|\lambda)]$ of complete data (O, q) by assuming a set of variables for hidden states q (that generates the observed data O) and initial values of model parameters λ^k . Then, it maximizes $Q(\lambda, \lambda^k)$ for estimation of the parameters. However, maximization of $Q(\lambda, \lambda^k)$ maximizes the likelihood

L and gives ML estimation of HMM. The log-likelihood L is simplified by taking the expectation of both sides of (2) with respect to the distribution of hidden states $P(q|O, \lambda^k)$ given the parameter values λ^k . Thus, by (2)

$$\begin{aligned} & \sum_{\forall q \in Q} P(q|O, \lambda^k) \log [P(O|\lambda)] \\ &= \sum_{\forall q \in Q} P(q|O, \lambda^k) \log [P(O, q|\lambda)] \\ & \quad - \sum_{\forall q \in Q} P(q|O, \lambda^k) \log [P(q|O, \lambda)]. \end{aligned} \quad (3)$$

Since $\sum_{\forall q \in Q} P(q|O, \lambda^k) = 1$ and $P(O|\lambda)$ is independent of λ^k

$$\begin{aligned} \log [P(O|\lambda)] &= \sum_{\forall q \in Q} P(q|O, \lambda^k) \log [P(O, q|\lambda)] \\ & \quad - \sum_{\forall q \in Q} P(q|O, \lambda^k) \log [P(q|O, \lambda)]. \end{aligned} \quad (4)$$

Thus, using the log-likelihood L and $Q(\lambda, \lambda^k)$

$$L = \log [P(O|\lambda)] = Q(\lambda, \lambda^k) - H(\lambda, \lambda^k) \quad (5)$$

where $H(\lambda, \lambda^k) = \sum_{\forall q \in Q} P(q|O, \lambda^k) \log [P(q|O, \lambda)]$. For two successive iterations of the EM algorithm, if the values for HMM parameters are λ^k and λ^{k+1} , then, by (5), we get

$$\log [P(O|\lambda^{k+1})] = Q(\lambda^{k+1}, \lambda^k) - H(\lambda^{k+1}, \lambda^k) \quad (6)$$

$$\log [P(O|\lambda^k)] = Q(\lambda^k, \lambda^k) - H(\lambda^k, \lambda^k). \quad (7)$$

By subtracting (6) and (7)

$$\begin{aligned} & \log [P(O|\lambda^{k+1})] - \log [P(O|\lambda^k)] \\ &= \underbrace{Q(\lambda^{k+1}, \lambda^k) - Q(\lambda^k, \lambda^k)}_{\text{Part-A}} - \underbrace{H(\lambda^{k+1}, \lambda^k) + H(\lambda^k, \lambda^k)}_{\text{Part-B}}. \end{aligned} \quad (8)$$

By using Jensen's inequality and $\sum_{\forall q \in Q} P(q|O, \lambda^{k+1}) = 1$ in Part-B of (8), we get

$$\begin{aligned} & \sum_{\forall q \in Q} \log [P(q|O, \lambda^k)] \log [P(q|O, \lambda^k)] / P(q|O, \lambda^{k+1}) \\ &= \sum_{\forall q \in Q} P(q|O, \lambda^k) [-\log [P(q|O, \lambda^{k+1})] / P(q|O, \lambda^k)] \\ &\geq -\log \sum_{\forall q \in Q} P(q|O, \lambda^k) P(q|O, \lambda^{k+1}) / P(q|O, \lambda^k) \\ &= -\log(1) = 0. \end{aligned} \quad (9)$$

By definition, the EM algorithm maximizes $Q(\lambda, \lambda^k)$, and the maximized value of the model parameters is λ^{k+1} such that $\lambda^{k+1} = \arg \max_{\lambda} Q(\lambda, \lambda^k)$. Therefore

$$Q(\lambda^{k+1}, \lambda^k) - Q(\lambda^k, \lambda^k) \geq 0. \quad (10)$$

By using (9) and (10) in (8), we get

$$\log [P(O|\lambda^{k+1})] - \log [P(O|\lambda^k)] \geq 0. \quad (11)$$

Equation (11) shows that EM maximizes the expected log-likelihood $Q(\lambda, \lambda^k)$ of complete data, but it indirectly maximizes the log-likelihood $L = \log [P(O|\lambda)]$ of the observed data. The EM estimates the HMM parameters by maximization of $Q(\lambda, \lambda^k)$ in two steps. In the E-step (Expectation), EM computes $Q(\lambda, \lambda^k)$ with respect to $P(q|O, \lambda^k)$ given the initial value λ^k . In the M-step (Maximization), EM maximizes $Q(\lambda, \lambda^k)$ of E-step and obtains a new set of values for model parameters λ^{k+1} . The procedure is repeated until L does not change. The estimated values for HMM parameters (\bar{c}_{jn} , $\bar{\mu}_{jn}$, $\bar{\Sigma}_{jn}$, and \bar{a}_{ij}) are as follows:

$$\bar{\Sigma}_{jn} = \sum_{t=1}^T \gamma_t(j, n) (O_t - \mu_{jn})(O_t - \mu_{jn})' / \sum_{t=1}^T \gamma_t(j, n) \quad (12)$$

$$\bar{a}_{ij} = \sum_{t=1}^{T-1} \xi_t(i, j) / \sum_{t=1}^{T-1} \gamma_t(i) \quad (13)$$

$$\bar{\mu}_{jn} = \sum_{t=1}^T \gamma_t(j, n) \cdot O_t / \sum_{t=1}^T \gamma_t(j, n) \quad (14)$$

$$\bar{c}_{jn} = \sum_{t=1}^T \gamma_t(j, n) / \sum_{t=1}^T \sum_{n=1}^M \gamma_t(j, n) \quad (15)$$

$$\xi_t(i, j) = \frac{\alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{\sum_{i=1}^K \sum_{j=1}^K \alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)} \quad (16)$$

$$\gamma_t(i) = \sum_{j=1}^K \xi_t(i, j) \quad (17)$$

$$\gamma_t(j, n) = \frac{\alpha_t(j) \beta_t(j)}{\sum_{j=1}^K \alpha_t(j) \beta_t(j)} \frac{c_{jn} N[O_t, \mu_{jn}, \Sigma_{jn}]}{\sum_{\hat{n}=1}^M c_{j\hat{n}} N[O_t, \mu_{j\hat{n}}, \Sigma_{j\hat{n}}]} \quad (18)$$

$$\beta_t(i) = \sum_{j=1}^K a_{ij} b_j(O_{t+1}) \beta_{t+1}(j) \quad (19)$$

where $t = T-1, T-2, \dots, 1$ and $\beta_T(i) = 1$

$$\alpha_{t+1}(j) = \sum_{i=1}^K \alpha_t(i) a_{ij} b_j(O_{t+1}) \quad (20)$$

where $1 \leq t \leq T-1$ and $\alpha_1(j) = \pi_j b_j(O_1)$. A detailed derivation for the estimated values of HMM parameters can be found in [1] and [2]. However, the estimation of HMM parameters using EM is not optimal. This is because EM is strongly dependent on the initial values of model parameters following (5), (8), (10), and (11), and it increases the values for the likelihood function L at successive iterations following (11). Due to these two properties, EM may terminate at a local maximum of the likelihood function L assuming it as a global maximum, which may give a nonoptimized estimation for

HMM resulting in a lower recognition accuracy. The problem can be avoided using a hybrid approach combining a local search (EM) and a stochastic global search (EA).

IV. CEL-EM WITH FUSION STRATEGIES

In general, standard EA [22], [23] can find globally competitive solutions for optimization problems by exploring the search space simultaneously using the numerous solutions in the population (details of the EA can be found in the literature [22], [23], [28]). However, the EA often represents an unsatisfactory compromise and suffers from the lack of accuracy when a high-quality solution is required for high-dimensional complex optimization problems [21]. In contrast, local search algorithms (such as EM) find the local maximum quickly by focusing solely on precision and time. The hybrids of EA and local search algorithms combine the benefits from the complementary properties of EA and EM [14], [15], [18], [20]. In the hybrid algorithm, EA converges on globally competitive solutions irrespective of local optima, and then, local search can potentially improve on the solutions discovered by the EA by ascending the hill to the optima of their corresponding attraction basins. However, when designing a hybrid algorithm, several important issues need to be considered such as fusion strategies of EA and EM, constraint-handling methods for EA, and creation of the initial population. In the CEL-EM, we focus on all of these issues.

The CEL-EM hybridizes a constraint-based EA and the EM to avoid the local maximum problem of EM while estimating the HMM for ASR systems. The CEL-EM executes a Lamarckian evolution [18], [24] periodically after the execution of a Darwinian evolution [22], [28] for a specific period of time using a staged-fusion strategy [15], [18], [20] (details of the Lamarckian evolution and Darwinian evolution can be found in [18], [22]–[24], and [28]). In the staged fusion, EA is executed up to a certain extent following a constraint-handling mechanism that forms one turn of the Darwinian evolution [22], [28]. The EM is executed on every offspring of the final EA population after each turn of Darwinian evolution [22], [23], [28]. The resulting solutions from EM are passed back to EA to reinitialize its initial population forming a Lamarckian evolution [18], [24]. Periodic execution of Lamarckian evolution [18], [24] after the execution of Darwinian evolution [22], [23], [28] for a specific period of time is performed over several iterations forming the two stages of the hybrid algorithm. The trait (problem-specific local knowledge) acquired by each individual during Lamarckian evolution (EM learning) is transmitted to the next EA generation using the reproduction operators of EA. However, periodic execution of EM in the staged fusion potentially improves on the globally competitive solutions discovered by the EA as well as maintains the global sampling capabilities of EA by minimizing the loss of hyperplane partition information [15], [22], [23].

During the execution of the EA stage, CEL-EM requires a feasible initial population, specific genetic operators, and constraint-handling methods for the satisfaction of HMM constraints. Conventional EA [25] handles the constraints in two

different ways. The first one is based on preserving feasibility of solutions (PFS) [25]. In PFS, when a particular parameter of a chromosome is mutated, EA determines the domain of the parameters by a function of the linear constraints and the remaining parameters of the chromosome. Then, the new value is taken from this domain with a probability distribution (uniform or nonuniform). Speech signals are so diverse, and constraints of HMM are so complex that to find a domain determination function for the parameters is very hard in ASR. However, a specialized genetic operator such as “Arithmetic Crossover” [25] could be used to transform a feasible chromosome into a feasible or a near-feasible chromosome [25] if we are provided an initial population for EA which has been created by the problem-specific heuristic method (not generated randomly). The second method is based on a penalty function (PF) [25]. The PF approach considers different levels based on the values of each constraint with a static penalty coefficient for each level in a particular chromosome. Due to many levels in each constraint, PF depends on the values of many static penalty coefficients. In HMM, there are many constraints. It is quite hard to get an optimal set of static penalty coefficients. Therefore, in the CEL-EM, two different constraint-based versions have been introduced. Both of these constraint-based versions of CEL-EM combine the properties from constraint-handling methods (PFS and PF) used in traditional EA. To take the merits of the PFS, CEL-EM uses an arithmetic crossover operator. The first constraint-based version of CEL-EM is a combination of the PFS and PF (PFS-PF). However, to avoid the demerits of the PF method, CEL-EM proposes another constraint-based version (PFS-LM) based on PFS and LMs [26], [27]. In PFS-LM, LMs are used to transform the constraint-based optimization problem into an unconstrained optimization problem [26], [27]. LMs also add some additional parameters, but the number of multipliers in LMs is fixed and the same as the total number of HMM constraints. The only concern here is to find the optimal values for LMs. CEL-EM proposes an evolutionary approach to find the values for LMs. In the following sections, these two versions of CEL-EM (PFS-PF and PFS-LM) and their fusion strategies are described in detail.

A. PFS-PF-Based CEL-EM

In the PFS-PF method, EA is executed up to certain extent, and then, EM is applied on each of the chromosome of the final EA population. The resulting EM population is passed back to EA to reinitialize the initial population of EA. This process is executed over several iterations. The EA process in PFS-PF follows several steps (creation of an initial population, selection, crossover, mutation, constraint handling, and evaluation). Finally, EM reestimation is executed. The flowchart for PFS-PF-based CEL-EM is given in Fig. 2. A detailed description of PFS-PF is given in the next sections.

1) *Creation of Initial Population in CEL-EM*: The initial population for EA in CEL-EM can be created using a random process. However, a random initial population is not feasible for HMM in ASR [31]. We have proposed a variable segmentation with incremental clustering approach to create the initial population for CEL-EM.

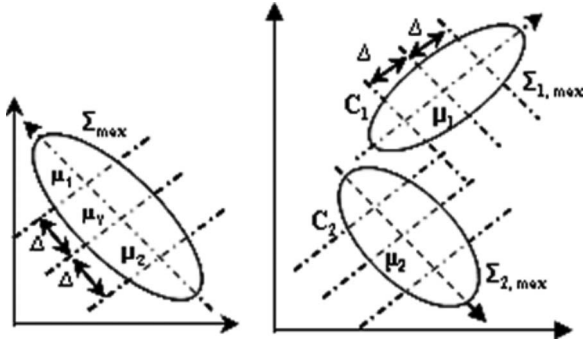


Fig. 1. Creation of initial population using variable segmentation with incremental clustering.

2) *Variable Segmentation With Incremental Clustering*: In segmentation, the observation vectors from the feature extraction process are divided into HMM states. Standard EM divides the observation vectors between HMM states uniformly [1]. In variable segmentation, a time index is considered for each vector of each state. Let the time index of the first vector of the i th state be denoted by b_i . Therefore, the i th state begins with a feature vector at time b_i and ends with a feature vector $b_i - 1$. The aim is to find some criterion for the $K - 1$ boundaries of the K states (b_2, b_3, \dots, b_{K-1}) where $b_1 = 1$ and $b_K =$ total number of observation vectors. The boundaries are computed by minimizing the total distortion: $\sum_{i=1}^K \sum_{t=b_i}^{b_{i+1}-1} \|O_t - \mu_i\|^2$, $\mu_i =$ mean of state i . Minimization of the total distortion is done by constrained clustering with a level-building dynamic programming technique [32]. The time index b_i is assigned an upper and a lower limit. The lower limit of b_i is unity, and the upper limit of b_i is the maximum number of frames per HMM state. For each value of the maximum number of frames, a separate segmentation is computed. We varied the upper limit of b_i and applied constrained clustering [32] that gives several different segmentations. Then a Gaussian model of the observation vectors is obtained by estimating the expected vector “ μ ” and the covariance matrix “ Σ ” for each HMM state. The expected vector represents the data centroid μ_Y in Fig. 1. The higher distortion direction is determined by the eigenvector “ Σ_{\max} ” associated with the maximum eigenvalue σ_{\max}^2 . Two new centroids μ_1 and μ_2 are optimally calculated as the couple of points at a distance $\Delta = \sqrt{2/\pi} \sigma_{\max}$ from the centroid μ_Y in the direction of maximal distortion. The two new centroids μ_1 and μ_2 outline two clusters C_1, C_2 (Fig. 1) obtained by the nearest neighbor approach with Euclidean distance. A Gaussian model is found for each of the two clusters by estimating the mean and covariance matrix. The weight associated with each mixture component is given by the fraction of vectors belonging to each cluster. The cluster C_1 with the highest distortion is divided again (Fig. 1). Cluster C_2 is taken as a Gaussian mixture of a particular HMM state. Thus, given a set of n clusters for a particular step, a set of $(n + 1)$ clusters is found. This gives $(n + 1)$ Gaussian mixtures for a particular HMM state which is repeated for all states of HMM. This gives one initial model for a particular phoneme. The process is repeated for other values of maximum frame number, and other initial models are obtained. The EM reestimation is applied

on each initial model, which gives initial population $P_1(t)$ for “CEL-EM.”

3) *Encoding of the Population in CEL-EM*: Each chromosome in the initial population $P_1(t)$ of CEL-EM is encoded as the real variable which consists of the parameters of each HMM state ($c_{jn}, \mu_{jn}, \Sigma_{jn}$) and transition probabilities $A = a_{ij}$. Each chromosome of the population in CEL-EM is encoded as

$$I_y = \underbrace{(c_{11}, \mu_{11}, \Sigma_{11}), \dots, (c_{1M}, \mu_{1M}, \Sigma_{1M})}_{\text{Gaussian Model for HMM state } j=1}, \dots, \underbrace{(c_{K1}, \mu_{K1}, \Sigma_{K1}), \dots, (c_{KM}, \mu_{KM}, \Sigma_{KM})}_{\text{Gaussian Model for HMM state } j=K}, A. \quad (21)$$

4) *Computation of Average Log-Likelihood and Evaluation of Initial Population in CEL-EM*: After encoding, the chromosomes of the initial population $P_1(t)$ are evaluated using the average value of likelihood $P_{\text{avg}}(\lambda) = (1/I) \sum_{l=1}^I \log(P_l(O|\lambda))$ over the total number of instances I of a phoneme. Since, in the encoded population, any chromosome is equivalent to model parameter λ , the likelihood L of each chromosome is computed using the forward-backward algorithm [1] with

$$P_l(O|\lambda) = \sum_{i=1}^K \alpha_T(i) \quad (\text{Instance } l \text{ has } T \text{ features}) \quad (22)$$

where $\alpha_T(i)$ is computed using (20), which is a recursive procedure that follows the forward-backward algorithm [1]. Then, the average value P_{avg} for one chromosome in $P_1(t)$ is taken over all instances. Chromosomes in $P_1(t)$ are ordered based on their P_{avg} in descending. Since EM satisfies the constraints of HMM [1]–[3], the initial population is evaluated using P_{avg} only.

5) *Selection Procedure in CEL-EM*: A stochastic sampling with replacement (SSR) [33] selection strategy is applied in PFS-PF. Top order chromosome is assigned with a ranking position equal to population size. Fitness of the chromosomes is computed by

$$\text{Fitness} = 2 - \text{SP} + 2 * (\text{SP} - 1) * (\text{pos} - 1) / (\text{PS} - 1) \quad (23)$$

where SP = selection pressure $\in \{1.0, 2.0\}$, pos = ranked position of the chromosome in the ordered population, and PS = total number of chromosomes in the population. The chromosomes are mapped to contiguous segments of a line, such that each chromosome’s segment is equal in size to its fitness. This is done by computing the selection probability (where selection probability = fitness/PS) which is mapped along a line. The highest fitness chromosome is mapped first, and then, the rest are following a descending order. A random number is selected between zero and one. If the random number falls in any boundary value along the line, the corresponding chromosome is selected for the parent pool. The process is repeated until the desired number of chromosomes is obtained to build the pool of parent $P_2(t)$.

6) *Crossover and Mutation Process in CEL-EM*: During crossover, a pair of parent chromosomes (P_1 and P_2) is randomly selected from the parent pool $P_2(t)$. Then, a crossover

operator is applied on P_1 and P_2 , which does some exchange and reordering of information in parent chromosomes and produces two offspring (Offspring₁ and Offspring₂). The CEL-EM needs to ensure the offspring chromosomes in the population do not violate the constraints. The use of the specific genetic operator arithmetic crossover [25] is a promising way to do this. Therefore, we have used arithmetic crossover [25]. Arithmetic crossover requires a feasible initial population. The variable segmentation approach for creation of the initial population meets this requirement. The variables of offspring (Offspring₁ and Offspring₂) are determined by

$$\text{Offspring}_1 \text{Var}_r = P_1 \text{Var}_r + \alpha(P_2 \text{Var}_r - P_1 \text{Var}_r) \quad (24)$$

$$\text{Offspring}_2 \text{Var}_r = P_1 \text{Var}_r + (1 - \alpha)(P_2 \text{Var}_r - P_1 \text{Var}_r) \quad (25)$$

where Offspring₁Var_r = r th variable of Offspring₁, Offspring₂Var_r = r th variable of Offspring₂, $P_1 \text{Var}_r$ = r th variable of P_1 , and $P_2 \text{Var}_r$ = r th variable of P_2 , with $r = 1, 2, \dots, V_T$. V_T = total number of variables, and $\alpha \in \{0, 1\}$. After crossover, the mutation operator is applied on the variables of offspring (Offspring₁ and Offspring₂), which gives chromosomes for population $P_3(t)$. The process is repeated for a desired number of chromosomes in $P_3(t)$. We have used the same mutation operator as in the breeder genetic algorithm [34]. Offspring variables are mutated by small perturbations with low probability using

$$\text{Offspring}_g \text{Var}_r^{\text{after mutation}} = \text{Offspring}_g \text{Var}_r + S_r * R_r * A_r \quad (26)$$

where $g = \{1, 2\}$, $S_r \in \{-1, +1\}$ uniform at random, $R_r = R * \text{range}_r$, R = mutation range $\in \{0.1, 10^{-6}\}$, and $A_r = 2^{-u * \text{mp}}$, with $u \in \{0, 1\}$, mp = mutation precision = 4, 5, 6, ..., 20, and range_r = range of value for the variable " r ." The range is determined from the mean value of the initial population and adding a multiple of standard deviation of the population with the mean.

7) *Evaluation of Offspring Population and Constraint-Handling Methods in PFS-PF-Based CEL-EM:* In PFS-PF, the evaluation of the chromosomes in the offspring population $P_3(t)$ is computed by taking the advantages of both the PFS and PF method used in conventional EA [25]. The conventional EA considers several levels for each constraint depending on the values of constraint. Each level is assigned with a static penalty coefficient. HMM has many constraints. Making levels for constraints requires huge penalty coefficient. It is also difficult to set the optimal values for many static penalty coefficients for HMM constraints. The PFS-PF method makes a grouping of the chromosomes of $P_3(t)$ and penalizes the chromosomes according to (27) by following the procedure described next.

- 1) If chromosomes (I_y) are not violating any constraints, then these are ranked with the top position.
- 2) If chromosomes violate the observation probability constraints but the mixture constraints and transition probability constraints are preserved, they are ranked with the second position.

- 3) If observation probability constraints are preserved but other constraints are violated, then the chromosomes get the third position.
- 4) If all constraints are violated, then these chromosomes get the lowest position.

$$F(I_y) = \begin{cases} f(I_y) & \text{No violation} \\ f(I_y) + \left[\frac{n_1}{N_1} + \frac{n_2}{N_2} + \frac{n_3}{N_3} \right] f(I_y) & \end{cases} \quad (27)$$

where $f(I_y) = P_{\text{avg}}$, $F(I_y)$ = penalized value of P_{avg} , and n_1, n_2, n_3 = total number of constraints violation in each category of constraints. N_1, N_2, N_3 are the total number of constraints in each category. After grouping, the chromosomes in each group are evaluated using $F(I_y)$ in (27). Each group of chromosomes in population $P_3(t)$ is sorted based on $F(I_y)$ in descending. Then, evaluated $P_3(t)$ is passed to the next EA generation where a parent selection pool $P_2(t)$ is built using selection procedure SSR [33]. EA is executed up to a predetermined maximum number of EA generations. This finishes one turn of Darwinian evolution.

8) *Execution of EM Reestimation in PFS-PF:* When EA execution stops, the population $P_3(t)$ is passed to the EM-reestimation process. For each chromosome in $P_3(t)$, the EM-reestimation process is executed. Taking the chromosome's parameter value as the initial HMM parameter λ^k , $P_{\text{avg}}(\lambda^k)$ is computed by the forward-backward algorithm [1]. Then, new values for parameters λ^{k+1} are determined using the reestimation formulas [see (12)–(15)]. $P_{\text{avg}}(\lambda^{k+1})$ is computed for new values λ^{k+1} . The procedure is iterated until $P_{\text{avg}}(\lambda^{k+1}) - P_{\text{avg}}(\lambda^k) \leq \text{threshold value}$ which is the termination criteria (TC) for EM. The estimated parameter values by EM reestimation for each chromosome of $P_3(t)$ build the population $P_4(t)$. $P_4(t)$ is sent back to reinitialize the initial EA population $P_1(t)$ for the next turn of Darwinian evolution.

9) *TC for PFS-PF-Based CEL-EM:* One complete execution of EA and EM forms a complete iteration for PFS-PF-based CEL-EM which is executed over several iterations. Then, the best individual from the final EM execution is taken as the final HMM model.

B. PFS-LM-Based CEL-EM

The complete flowchart for PFS-LM is shown in Fig. 3. Creation of the initial population $P_1(t)$ in PFS-LM and the evaluation of this population $P_1(t)$ follow the same procedure as PFS-PF. The parent pool $P_2(t)$ is created using the SSR [33] procedure used in PFS-PF. Crossover and mutation operators are applied on $P_2(t)$ to create the offspring population $P_3(t)$. The crossover and mutation procedure is the same as PFS-PF. However, the evaluation of chromosome and constraint-handling method in PFS-LM uses the LM-based approach.

1) *LM for Constraint-Based Optimization Problem:* The LM-based approach is widely used in many constraint-based optimization problems [26], [27]. Let the HMM constraints be considered as $g(x) = 1$, where $g(x) = g^m(x)$, $m = 1, 2, 3, \dots, V$ (V = total number of HMM constraints) and x are the HMM parameters. If there exists a maximum value m_0 of the objective function P_{avg} , then there exists a real

valued vector “ γ ” which is known as an LM such that $m_0 = \max_{x \in \Omega} (P_{\text{avg}} - \langle \gamma, g(x) \rangle)$, where $\langle \cdot \rangle$ is the inner product. If both P_{avg} and $g(x)$ are differentiable, then (28) holds for each of the m th constraints

$$\nabla P_{\text{avg}}(x_0) = \gamma^m \nabla g^m(x_0). \quad (28)$$

Here, x_0 is the value of HMM parameters at maximum. Therefore, if we can find the values for LM “ γ ,” then it is possible to transform the original constraint-based optimization problem into an unconstraint-based problem by maximizing the function $E(P_{\text{avg}}, \gamma, g(x))$ in the following equation using the likelihood value P_{avg} :

$$E(P_{\text{avg}}, \gamma, g(x)) = P_{\text{avg}} - \langle \gamma, (1 - g(x)) \rangle. \quad (29)$$

The PF approach [25] requires different levels based on the values of each HMM constraint with a static penalty coefficient for each level. Due to many HMM constraints with required levels for each, PF depends on the values of many static penalty coefficients. The PFS-PF avoids the levels of many constraints and considers only violation of constraints. Thus, all constraint violations are considered as equal disregarding their numerical values. The LM-based approach only needs a fixed number of coefficients equal to the total number of HMM constraints and considers the numerical values of each constraint of each chromosome. Thus, each chromosome is properly penalized in the PFS-LM-based approach.

2) *Determination of the Values for LM and Constraint-Handling in PFS-LM-Based CEL-EM:* The LM-based constraint-handling method has been used to evaluate the EA offspring population $P_3(t)$ in the EA process of PFS-LM in Fig. 3. The values for LM (γ_1) for $P_3(t)$ are determined using a separate EA process for LM (LM-EA). The P_{avg} of each chromosome in $P_3(t)$ is computed. $P_3(t)$ is sorted based on P_{avg} . For the first EA generation in LM-EA, a randomly generated initial LM population $M_1(t)$ is created. By using the worst chromosome “ I_w ” with the lowest P_{avg} ($^w P_{\text{avg}}$) from $P_3(t)$ and the constraint values of “ I_w ,” chromosomes in the $M_1(t)$ are evaluated by the following equation where each LM chromosome gets $^w P_{\text{avg}}$ and m th constraint value (g_w^m) of “ I_w .”

$$E(^w P_{\text{avg}}, \gamma_1, g_w^m(x)) = ^w P_{\text{avg}} - \{ \gamma_1^1 * (1 - g_w^1(x)) + \dots + \gamma_1^V * (1 - g_w^V(x)) \}. \quad (30)$$

An SSR [33] selection technique is applied on $M_1(t)$ to create the LM parent pool $M_2(t)$ in LM-EA. Then, the crossover and mutation operators are applied on the parent pool $M_2(t)$, which produces the LM offspring population $M_3(t)$ in LM-EA. The LM chromosomes of offspring $M_3(t)$ are evaluated using (30). SSR [33] is applied on offspring $M_3(t)$ to create the new LM parent pool for the next $(t + 1)$ th EA generation. $M_2(t)$ is replaced with the new LM parent pool which is used to create the offspring for $(t + 1)$ th EA generation in LM-EA. The EA process in LM-EA is executed over a predetermined number of iterations. The best LM chromosome γ_b is chosen from $M_3(t = \text{final generation})$ in LM-EA with the highest value for $E(^w P_{\text{avg}}, \gamma_1, g_w^m(x))$. For every offspring population $P_3(t)$ in

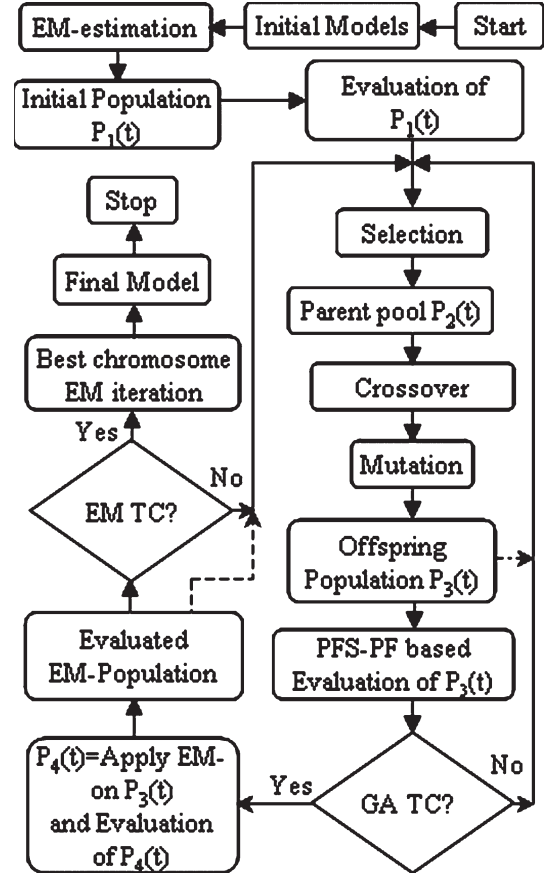


Fig. 2. Flowchart for PFS-PF-based CEL-EM.

the EA process of PFS-LM, a separate EA process for LM (LM-EA) is executed, and a new best LM is computed for the evaluation of the chromosome in $P_3(t)$. The LM offspring population $M_3(t)$ of the final EA generation in one LM-EA is taken as the initial LM population $M_1(t)$ for the LM-EA execution process in the next offspring population $P_3(t)$ of PFS-LM. Each individual in $P_3(t)$ is evaluated using the evaluation functions $E(P_{\text{avg}}, \gamma_b, g(x))$ in the following equation with corresponding values for $g(x)$ and γ_b :

$$E(P_{\text{avg}}, \gamma_b^m, g^m(x)) = P_{\text{avg}} - \{ \gamma_b^1 * (1 - g^1(x)) + \dots + \gamma_b^V * (1 - g^V(x)) \}. \quad (31)$$

The offspring individuals in $P_3(t)$ are sorted in descending order based on $E(P_{\text{avg}}, \gamma, g(x))$ where the best individual receives a position equal to the population size. The fitness of each individual in $P_3(t)$ is computed by (23). The SSR [33] is used to create the parent pool $P_2(t)$. The crossover and mutation process (same as PFS-PF) is used to create the offspring population $P_3(t)$. The EA process is executed over several iterations. Then, the final population of EA $P_3(t)$ is passed to EM reestimation.

3) *Execution of EM Reestimation in PFS-LM-Based CEL-EM:* EM reestimation is applied on each individual of population $P_3(t)$ that produces population $P_4(t)$. The evaluation of $P_4(t)$ is done using P_{avg} following the procedure described

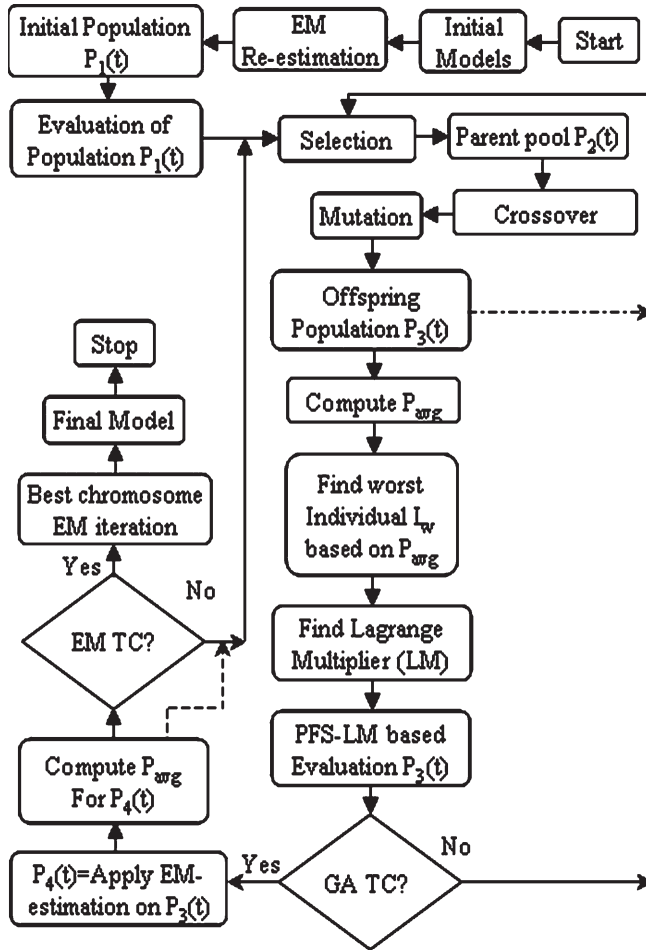


Fig. 3. Flowchart for PFS-LM-based CEL-EM.

earlier. Population $P_4(t)$ is passed to reinitialize the EA population for the next EA generation.

4) *TC for PFS-LM*: One complete EA process and one complete EM execution form a complete iteration of PFS-LM. Complete PFS-LM is executed over a predetermined number of iterations. The final model is the chromosome with the highest P_{avg} chosen from EM population $P_4(t)$ of the last iteration of PFS-LM.

C. Fusion Strategies for Hybridization of EA and EM in the CEL-EM

Two different staged-fusion strategies have been proposed for the CEL-EM. The first strategy (Fusion-1) follows a periodic Lamarckian evolution [18] with Darwinian evolution [28]. Both Figs. 2 and 3 for PFS-PF and PFS-LM describe the use of first fusion strategy (Fusion-1). The other fusion strategy (Fusion-2) uses a biased crossover operator with the staged-fusion technique that is conceptually similar to the hybridization techniques used in immune recruitment mechanism [29] and Simplex-Crossover [30] where the local exploitation properties of EA are enhanced by biasing the traditional reproduction operators (crossover and mutation). In the second fusion strategy (Fusion-2) of CEL-EM, the crossover operator is biased by the local knowledge of the EM algorithm. However,

the evaluation of the chromosomes in the EA population is performed after mutation and follows any of the constraint-handling mechanisms of CEL-EM (PFS-PF and PFS-LM). A complete algorithm for fusion strategy-2 (Fusion-2) based on PFS-LM is given in Algorithm 1. In fusion strategy-2 (Fusion-2), creation of the initial population $P_1(t)$, evaluation of $P_1(t)$ remains the same as in (Fusion-1). Creation of parent pools $P_2(t)$ and $P_4(t)$ follows the SSR [33] selection process described earlier in strategy-1 (Fusion-1). However, the crossover operator is biased by the local knowledge of EM. The biasing is done by the execution of EM reestimation with one iteration only on each offspring chromosome in $P_5(t)$. $P_5(t)$ is obtained by applying the crossover operator on $P_4(t)$. Then, the mutation operator is applied on the results of EM reestimation (on $P_5(t)$), which gives the offspring population $P_6(t)$. $P_6(t)$ is evaluated using the PFS-LM-based method as in Fig. 3. Constraint handling could also be used using PFS-PF as described in Fig. 2. For the PFS-LM, the LMs γ_1 and γ_2 are determined using the same procedure as described earlier by (30) and by the following equation:

$$E({}^w P_{avg}, \gamma_2, g_w^v(x)) = {}^w P_{avg} - \{\gamma_2^1 * (1 - g_w^1(x)) + \dots + \gamma_2^V * (1 - g_w^V(x))\}. \quad (32)$$

Algorithm 1 Algorithm for fusion strategy (Fusion-2) of CEL-EM (based on PFS-LM) for estimating HMM parameters in speech signal modeling

Constants *Maximum_EA_Iteration*,
Maximum_Iteration_CEL_EM
Variables *Iteration_EA*,
Iteration_CEL_EM : Integer
Highest_Pavg, *Highest_Pavg_EM* : Real
begin
Iteration_EA \leftarrow 0
Iteration_CEL_EM \leftarrow 0
Training_Data \leftarrow *D*
Features \leftarrow *Extract_Features(D)*
Models P(t) \leftarrow *Variable_Segmentation(Features)*
Initial_Population P1(t) \leftarrow *Apply EM on P(t)*
Compute P_{avg} for $P_1(t)$
Evaluation of $P_1(t)$ by P_{avg}
 $P_2(t) \leftarrow$ SSR on $P_1(t)$
 $P_3(t) \leftarrow$ crossover, mutation on $P_2(t)$
Compute P_{avg} for individuals in $P_3(t)$
 $\gamma_1 \leftarrow$ Find LM using EA and worst individual in $P_3(t)$
Evaluation of $P_3(t)$ using PFS-LM method with γ_1
repeat
repeat
 $P_4(t) \leftarrow$ SSR selection on $P_3(t)$
 $P_5(t) \leftarrow$ Crossover on $P_4(t)$
 $P_6(t) \leftarrow$ Apply One-iteration of EM on $P_5(t)$,
then mutation on the results from EM
Compute P_{avg} for individuals in $P_6(t)$
 $\gamma_2 \leftarrow$ Find LM using EA and worst individual in $P_6(t)$
Evaluation of $P_6(t)$ using PFS-LM method with γ_2
 $P_3(t) \leftarrow P_6(t)$

```

    Iteration_EA ← Iteration_EA + 1
    until (Iteration_EA < Maximum_EA_Iteration)
    Iteration_EA ← 0
     $P_7(t) \leftarrow$  EM estimation on  $P_6(t)$ 
    Compute  $P_{avg}$  and evaluation of  $P_7(t)$  by  $P_{avg}$ 
    Iteration_CEL_EM ← Iteration_CEL_EM + 1
    until (Iteration_CEL_EM < Maximum_Iteration_CEL_EM)
    Highest_ $P_{avg\_EM} \leftarrow$  Find highest  $P_{avg}$  in  $P_7(t)$ 
    Best_Model ← chromosome in  $P_7(t)$  with
    Highest_ $P_{avg\_EM}$ 
    RETURN Best_Model
end

```

γ_2 is determined using the worst individual (lowest $^w P_{avg}$) of population $P_6(t)$ by (32). The best model is determined from the final EM population $P_7(t)$ with the highest P_{avg} . The TC for Fusion-2 remain the same as the first fusion strategy (Fusion-1).

V. EXPERIMENTAL RESULTS

The TIMIT acoustic phonetic speech corpus has been used for evaluating the performance of all versions of CEL-EM. The TIMIT [35] corpus has been designed to provide speech data for the development and evaluation of the ASR systems. The TIMIT [35] contains three main categories of utterances: dialect utterances (the SA sentences), phonetically compact utterances (the SX sentences), and phonetically diverse utterances (the SI sentences). In the TIMIT, a total of 4620 utterances are used for training set. There are two test sets (test set-A and test set-B) in the TIMIT. Test set-A has 1680 utterances (SA, SX, and SI). Test set-B has 1344 utterances (SX and SI). SA sentences expose the dialectal variants of the speakers. However, SX and SI sentences are phonetically compact/diverse sentences. The test set-B does not have any SA sentence. Therefore, it is more appropriate [35] for performance tests of ASR systems [35]. The detailed description of TIMIT data set can be found in the TIMIT manual [35].

Mel frequency cepstral coefficients (MFCC) [36] feature extraction process has been used in both training and testing of CEL-EM. In the MFCC [36] feature extraction process, 39-dimensional feature vectors have been produced. Each 39-dimensional feature vector includes 12 MFCC components, one log energy component, 13 first-order differences (delta coefficients), and 13 second-order differences (delta-delta coefficients). The sampling rate for computing feature vectors was 16 kHz. The window size was 32 ms (512 samples). The window overlap was 352 samples, while the frame rate was 100 frames/s.

The TIMIT training set has been used to train all versions of the CEL-EM (PFS-LM and PFS-PF with two fusion strategies). During training, a simple left-to-right CDHMM with no node skipping has been considered for the model of each phoneme class in the training data. Training has been done considering the different numbers of Gaussian mixtures (Mixtures 3, 5, 8, 10) for HMM. The following parameters have been set during training: the EM convergence threshold: 0.50, maximum number of generations for EA: 10, maximum number of iterations for CEL-EM: 20, and maximum number of iterations for EA process for determination of the values for LM in PFS-LM: 20.

Experiments have been done for three different types of algorithm (standard EM, VIA to EM, and CEL-EM) to test and compare the performance of CEL-EM. The EM algorithm (standard EM) has been executed with initial point created by uniform segmentation. Since the performance of standard EM is strongly dependent on the initial point, EM has been repeatedly executed on the different initial points created by a VIA using variable segmentation technique to obtain the best performance from standard EM. We refer to this approach as a VIA to EM (VIA-EM), and it is used to compare the performance of CEL-EM. The best model of VIA-EM is chosen based on the highest value of the objective function among all models obtained by applying EM repeatedly on the different initial points. To test the performance of all versions of CEL-EM and compare the results to EM and VIA-EM, we tested the recognition accuracies of the HMM models for speech computed by CEL-EM, EM, and VIA-EM on TIMIT test sets. Tests have been carried out for Gaussian mixtures (3, 5, 8, 10) for the different versions of CEL-EM, EM, and VIA-EM. Two test sets of the TIMIT data set, namely, test set-A and test set-B, have been used to test the performance of the CEL-EM. A phonetic bigram grammar language model has been used in testing.

A. Results

The experimental results are described in Figs. 4–7, Tables I–IV. In Fig. 4, the values for the objective function (P_{avg}) of PFS-LM- and PFS-PF-based CEL-EM (Fusion-1) and VIA-EM have been plotted for Gaussian mixtures (3, 5, 10) for different phonemes. Recognition accuracies of the different HMM models for Gaussian mixtures (3, 5, 8, 10) obtained by CEL-EM (PFS-LM and PFS-PF for Fusion-1) for test set-A and test set-B have been depicted in Tables I and II. The accuracies have also been compared to the standard EM as well as VIA-EM. It is found in Tables I and II and Fig. 4 that both PFS-LM and PFS-PF (Fusion-1) have obtained higher values for objective function and recognition accuracies than the standard EM as well as VIA-EM. However, PFS-LM achieves higher values for recognition accuracies as well as values for P_{avg} than PFS-PF for both test sets.

This demonstrates the effectiveness of the LM-based approach (PFS-LM) constraint-handling technique in CEL-EM. The values of P_{avg} obtained by both fusion strategies (Fusion-1 and Fusion-2) of the PFS-LM-based CEL-EM and VIA-EM for Gaussian mixture-8 are shown in Fig. 5. Fig. 5 indicates that for a particular constraint-handling mechanism of CEL-EM, if we change the fusion strategy from Fusion-1 to Fusion-2, values for P_{avg} are improved. Recognition accuracies on test sets (A and B) of PFS-LM- and PFS-PF-based CEL-EM (Fusion-2) have been depicted in Tables I and II for Gaussian mixtures (3, 5, 8, 10). It is found that both PFS-LM and PFS-PF in Fusion-2 achieve higher recognition accuracies than the standard EM as well as VIA-EM in both test sets. A comparison of recognition accuracies in Figs. 6 and 7 indicates that, in both constraint-handling mechanisms (PFS-LM and PFS-PF), Fusion-2 has performed better than Fusion-1 for both test sets. This demonstrates the importance of enhancing local exploitation in the EA in Fusion-2 of CEL-EM.

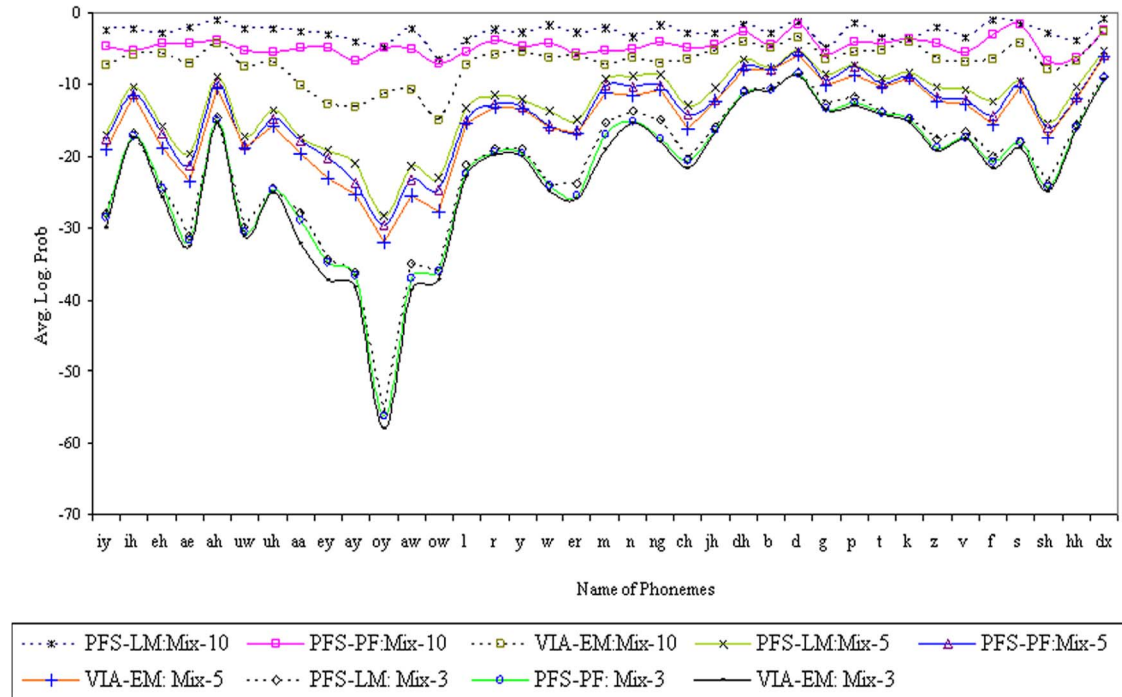


Fig. 4. Values for objective function (average log probability) for different Gaussian mixtures (3, 5, 10) for different phonemes obtained by PFS-LM- and PFS-PF-based CEL-EM for fusion strategy-1 and EM. Y-axis denotes the average log probability. X-axis denotes the name of the phonemes. PFS-LM:Mix-10 = PFS-LM for Gaussian mixture 10, PFS-LM:Mix-5 = PFS-LM for Gaussian mixture 5, and similar meaning should be taken for other legends.

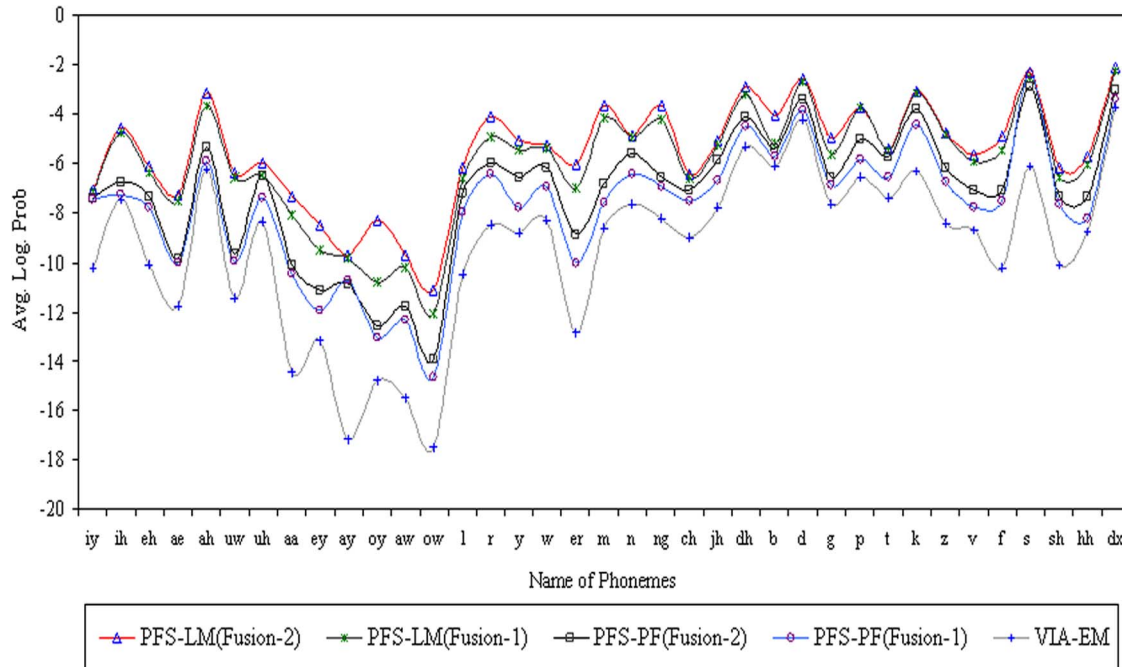


Fig. 5. Values for objective function (average log probability) for different Gaussian mixtures-8 for different phonemes obtained by PFS-LM- and PFS-PF-based CEL-EM for fusion strategy-2 and EM. Y-axis denotes the average log probability. X-axis denotes the name of the phonemes. PFS-LM (Fusion-2) = PFS-LM for Fusion strategy-2 for Gaussian mixture-8, and similar meaning should be taken for other legends.

VI. DISCUSSION AND SIGNIFICANCE OF THE RESULTS

Experimental results demonstrate that all versions of CEL-EM (PFS-PF and PFS-LM with two fusion strategies) obtain higher values for the objective function and recognition accuracies compared to standard EM and VIA-EM. Therefore, CEL-EM computes better estimation for HMM parameters than

standard EM and VIA-EM. From Figs. 4 and 5, it is found that PFS-LM obtains higher values for objective function than PFS-PF for all different mixtures. While comparing the recognition accuracies, Tables I and II show that PFS-LM obtains higher accuracies than PFS-PF for all different mixtures. Therefore, PFS-LM computes better HMM models than PFS-PF. This

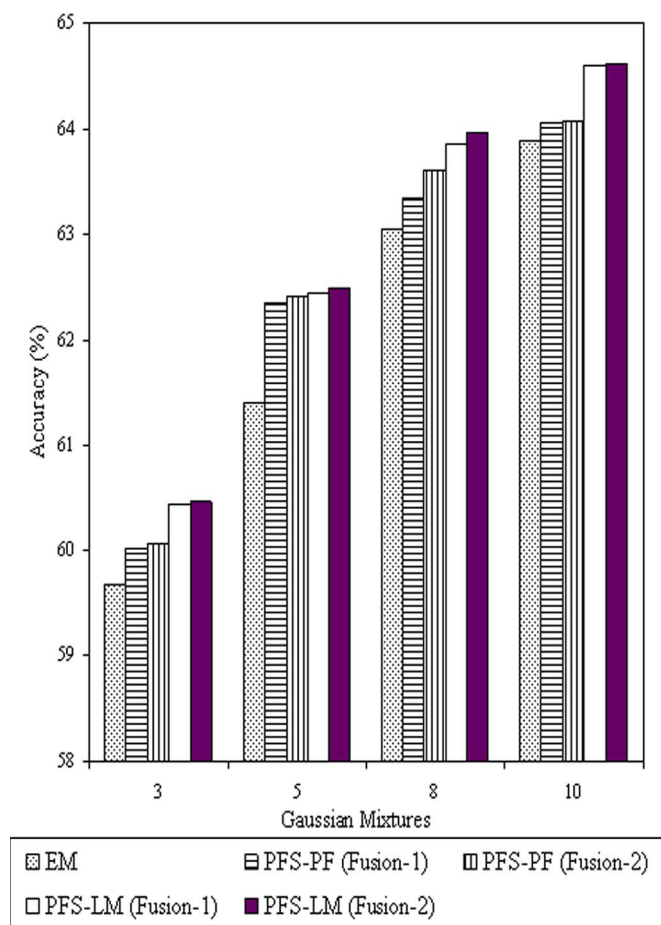


Fig. 6. Comparison of recognition accuracies between different versions of CEL-EM (PFS-PF and PFS-LM for both fusion strategies) and EM for test set-A.

is due to the improved constraint-handling technique (LM-based) used in the PFS-LM-based method. This also proves the advantages of the use of LM-based technique in CEL-EM. The advantages of the use of LM-based techniques have been verified in [26] and [27] where LM-based techniques have been applied on some other problems (not related to speech recognition) and improved results have been found. Compared to fusion strategy-1, fusion strategy-2 obtains higher recognition accuracies. Therefore, PFS-LM with Fusion-2 is more suitable to estimate HMM parameters than any other version of CEL-EM. Accuracies of CEL-EM have also been compared with other systems. Yung and Oh [37] achieve a baseline accuracy of 56% for the EM algorithm in HMM systems and 60.6% recognition accuracy for SFHMM system [37] for test set-B. For test set-B, the PFS-PF-based CEL-EM (Fusion-2) obtains an accuracy of 61.16%, and the PFS-LM-based CEL-EM (Fusion-2) obtains an accuracy of 61.59% for Gaussian mixture-10, which is higher than the SFHMM system [37]. The significance of the improvement in performance of CEL-EM over VIA-EM and EM has been verified using the standard “Matched-Pair test” [38].

In the “Matched-pair” test [38], let z be the difference of phoneme recognition error per utterance between CEL-EM and EM. The mean and variance of z are considered as μ_z and Σ_z which are computed from the recognition results of

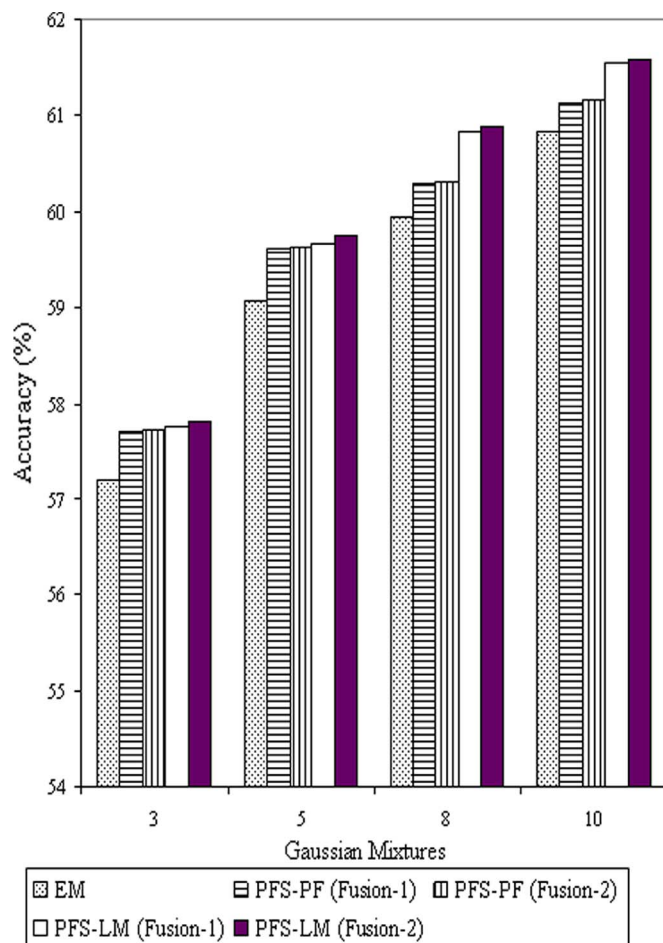


Fig. 7. Comparison of recognition accuracies in different versions of CEL-EM (PFS-PF and PFS-LM for both fusion strategies) and EM for different mixtures for test set-B.

CEL-EM, VIA-EM, and EM. If the total number of utterances “ u_t ” be sufficiently large ($u_t > 50$), then $W = \mu_z / (\sigma_z \sqrt{u_t})$ tends to a normal distribution $N(0, 1)$. Considering $z = w^r$, $P = 2P(z \geq |w^r|) = 2 \int_z^\infty \phi(t) dt$ is computed from $N(0, 1)$ table where w^r is the realized value of W and $\phi(t)$ is the corresponding density to normal distribution $N(0, 1)$. For a standard significance level [38], if $P < 0.05$, the results are deemed to be significantly different; otherwise, they are not. Tables III–VI give the matched pair test results for PFS-PF and PFS-LM for two fusion strategies on the two test sets. The results of the matched pair tests show that when compared to the VIA-EM only, the PFS-LM demonstrates an improvement in the recognition for both fusion strategies, which is significant; however, when compared to the standard EM, the improvements in recognition for both PFS-PF and PFS-LM are significant for both fusion strategies in all Gaussian mixtures.

Computational performance of CEL-EM is given in Fig. 8 for different mixtures. The experimental platform was 1.80-GHz Pentium-4 CPU with 512 MB of RAM. There are two main computational phases in any speech recognition system. One is training phase, and the other is the decoding phase (testing phase). In Fig. 8, only the training time of CEL-EM has been compared with EM. CEL-EM takes the same decoding time as EM, because the decoding algorithm is the same for both EM

TABLE I
TEST SET-A: RECOGNITION ACCURACY (PFS-LM:F-2, PFS-LM: F-1, PFS-PF:F-2, PFS-PF:F-1, VIA-EM, AND EM).
F-2 = FUSION STRATEGY-2. F-1 = FUSION STRATEGY-1

	Percent correct %						Accuracy %					
Mix- ture	PFS-LM (F-2)	PFS-LM (F-1)	PFS-PF (F-2)	PFS-PF (F-1)	VIA- EM	EM	PFS-LM (F-2)	PFS-LM (F-1)	PFS-PF (F-2)	PFS-PF (F-1)	VIA- EM	EM
3	63.06	63.01	62.68	62.7	62.44	62.32	60.47	60.43	60.06	60.01	59.81	59.67
5	65.04	64.95	64.92	64.83	64.53	64.11	62.49	62.45	62.42	62.35	61.93	61.41
8	66.61	66.44	66.29	65.95	65.74	65.67	63.96	63.86	63.61	63.35	63.16	63.05
10	67.11	67.11	66.53	66.52	66.43	66.31	64.62	64.59	64.07	64.05	63.97	63.89

TABLE II
TEST SET-B: RECOGNITION ACCURACY (PFS-LM: F-2, PFS-LM: F-1, PFS-PF: F-2, PFS-PF: F-1, VIA-EM, AND EM).
F-2 = FUSION STRATEGY-2. F-1 = FUSION STRATEGY-1

	Percent correct %						Accuracy %					
Mix- ture	PFS-LM (F-2)	PFS-LM (F-1)	PFS-PF (F-2)	PFS-PF (F-1)	VIA- EM	EM	PFS-LM (F-2)	PFS-LM (F-1)	PFS-PF (F-2)	PFS-PF (F-1)	VIA- EM	EM
3	60.49	60.42	60.38	60.40	60.26	60.01	57.82	57.76	57.73	57.71	57.42	57.19
5	62.46	62.34	62.31	62.26	62.1	61.82	59.75	59.67	59.63	59.61	59.46	59.08
8	63.72	63.60	63.07	63.06	62.86	62.71	60.88	60.83	60.31	60.30	60.12	59.94
10	64.22	64.22	63.78	63.74	63.67	63.4	61.59	61.55	61.16	61.12	61.06	60.83

TABLE III
RESULTS OF MATCHED-PAIR TEST OF PFS-PF AND PFS-LM (FUSION-1) ON RECOGNITION RESULTS FOR TIMIT TEST SETS-A: $u_t = 1680$

	PFS-PF / EM		PFS-PF / VIA-EM		PFS-LM / EM		PFS-LM /VIA-EM	
Mixture	W	P	W	P	W	P	W	P
3	3.79181	0.0001447	2.172153	0.03572	3.055	0.0027	3.59397033	0.0003182
5	8.0085	$1.44E^{-15}$	2.92189	0.003732	7.619924	$6.382E^{-14}$	3.89777794	0.0000962
8	2.03959	0.04556	2.0157	0.0455	5.546	$3.798E^{-8}$	5.22585060	$5.734E^{-7}$
10	1.99190	0.0466	1.194251	0.234	4.399	0.0000108	4.30429055	0.00001708

TABLE IV
RESULTS OF MATCHED-PAIR TEST OF PFS-PF AND PFS-LM (FUSION-2) ON RECOGNITION RESULTS FOR TIMIT TEST SETS-A: $u_t = 1680$

	PFS-PF / EM		PFS-PF / VIA-EM		PFS-LM / EM		PFS-LM /VIA-EM	
Mixture	W	P	W	P	W	P	W	P
3	2.2045404	0.0278	2.195325	0.03572	3.35353	0.0009668	3.88809026	0.0000962
5	8.7231128	$1.896E^{-17}$	3.951553	0.0000962	7.88	$6.382E^{-14}$	4.01428313	0.00006334
8	3.622862	0.0003182	3.1677157	0.0019352	5.900403	$3.798E^{-8}$	5.7704312	$3.798E^{-8}$
10	2.0102883	0.0444	1.2486044	0.215	4.7735975	0.0000026	4.68981754	0.000003212

and CEL-EM. Feature extraction time is the same for both EM and CEL-EM. From Fig. 8, it is found that CEL-EM takes more training time than standard EM. This is due to the hybridization of EM with EA in the CEL-EM. It is well known that the EA takes more computational time, thereby hybrids of EA and local search will also demand more time [8]–[10], [39], [40]. In [10], it is seen that the hybrid algorithm (EA + EM for a single GMM for image processing) takes 30 times longer training time than EM; another version of the algorithm takes 90 times longer training time than standard EM. CEL-EM takes approximately 30–50 times longer computational time than standard EM for different mixtures.

In practical speech recognition, real-time performance is measured in the decoding (actual recognition or testing), not in the training of the HMM models themselves. Indeed, commercial speech recognizers provide already pretrained models, and these are usually trained offline to produce the most accurate models. Due to the offline training arrangement of HMM in the ASR systems (where accuracy is more important than computational time), additional training time in CEL-EM is negligible.

The training time of CEL-EM can be reduced by using a parallel implementation of CEL-EM with a master-slave model [41]–[43] where the master process maintains the global

TABLE V
RESULTS OF MATCHED-PAIR TEST OF PFS-PF AND PFS-LM (FUSION-1) ON RECOGNITION RESULTS FOR TIMIT TEST SETS-B: $u_t = 1344$

	PFS-PF / EM		PFS-PF / VIA-EM		PFS-LM / EM		PFS-LM /VIA-EM	
Mixture	W	P	W	P	W	P	W	P
3	3.0982	0.0026	1.5620	0.19	3.792	0.0001447	2.2460	0.025
5	4.1093	0.00004	1.5934	0.118	4.2461	0.0000267	1.9697	0.04
8	2.8840	0.00511	1.7420	0.0818	5.758	$3.798E^{-8}$	4.5849	0.000004224
10	2.18010	0.034	0.883	0.234	4.599	0.000004224	3.07887	0.0022

TABLE VI
RESULTS OF MATCHED-PAIR TEST OF PFS-PF AND PFS-LM (FUSION-2) ON RECOGNITION RESULTS FOR TIMIT TEST SETS-B: $u_t = 1344$

	PFS-PF / EM		PFS-PF / VIA-EM		PFS-LM / EM		PFS-LM /VIA-EM	
Mixture	W	P	W	P	W	P	W	P
3	3.6226	0.00031	2.07040	0.0384	4.160192	0.00004132	2.638647	0.023
5	4.2170	0.00002	1.79663	0.0734	4.447	0.000010826	2.096863	0.036
8	2.9743	0.0037	1.77767	0.0768	5.9017341	$3.798E^{-8}$	4.907487	$9.5E^{-7}$
10	2.42380	0.0164	1.3051	0.19356	4.83948475	$1.58E^{-6}$	3.353678	0.000966

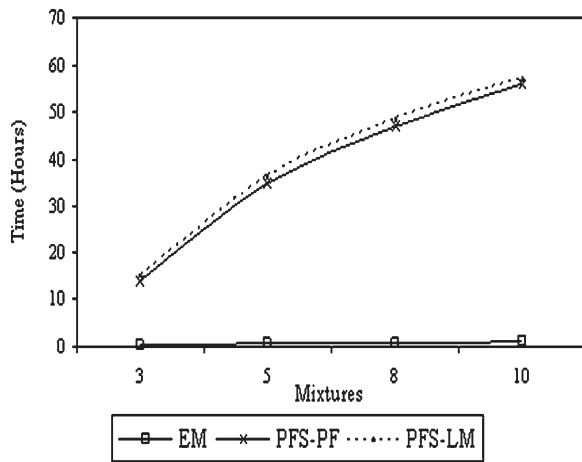


Fig. 8. Computational performance for different versions of CEL-EM (PFS-PF-based CEL-EM and PFS-LM-based CEL-EM) and EM for different numbers of Gaussian mixtures.

population and executes selection and the slave processes execute crossover, mutation, evaluation, and EM reestimation. We will implement a parallel version of CEL-EM in future work.

VII. CONCLUSION

This paper proposes a hybrid training algorithm (CEL-EM) using constraint-based EA and EM for better estimation of HMM parameters for speech signal modeling. In CEL-EM, EA explores the search space more thoroughly than EM and finds globally competitive solutions irrespective of local optima which are potentially improved on by the EM by ascending the hill to the optima of their corresponding attraction basins. This hybrid approach enables CEL-EM to avoid many local maxima which arise in standard EM-based training. While estimating the HMM parameters by taking the benefit of hybridization, a staged-fusion approach has been proposed in the CEL-EM to minimize the interference between the two algorithms, main-

tain the global sampling capabilities of EA, and meet the EA population-diversity requirement for optimization of high-dimensional objective function.

A constraint-based EA has been introduced in CEL-EM to satisfy the HMM constraints. Two constraint-based versions of CEL-EM with two different fusion strategies have been proposed. An LM-based constrained approach (PFS-LM) has been developed to avoid the problems (due to many HMM constraints) of traditional constraint-handling method (PFS-PF). A simple staged-fusion strategy and a staged fusion with a biased-crossover-operator strategy have been used in the CEL-EM. A VIA has been developed using a variable segmentation technique to provide better initialization to CEL-EM. Experimental results show that CEL-EM obtains a better estimation for HMM than EM because all versions of CEL-EM obtain higher recognition accuracies than a standard EM as well as a top standard EM developed by applying the VIA to EM (VIA-EM). However, PFS-LM performs better than PFS-PF, whereas staged fusion with biased crossover outperforms simple staged fusion. Therefore, staged fusion (biased crossover) with PFS-LM-based CEL-EM is more suitable than all other versions of CEL-EM for HMM training for speech signal modeling. In the future, we will implement a parallel version of CEL-EM to reduce the training time and apply CEL-EM to other HMM-based signal modeling techniques used in signal classification/phoneme recognition.

REFERENCES

- [1] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proc. IEEE*, vol. 77, no. 2, pp. 257–286, Feb. 1989.
- [2] S. E. Levinson, L. R. Rabiner, and M. M. Sondhi, "An introduction to the application of the theory of probabilistic functions of a Markov process to automatic speech recognition," *Bell Syst. Tech. J.*, vol. 62, no. 4, pp. 1035–1074, 1983.
- [3] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. R. Stat. Soc.*, vol. 39, no. 1, pp. 1–38, 1977.

- [4] C. F. J. Wu, "On the convergence properties of the EM algorithm," *Ann. Stat.*, vol. 11, no. 1, pp. 95–103, Mar. 1983.
- [5] N. Thatphithakkul and K. Supphanat, "HMM parameter optimization using tabu search," in *Proc. Int. Symp. Commun. Inf. Technol.*, 2004, pp. 904–908.
- [6] T. Y. Chen, X. D. Mei, J. S. Pan, and S. H. Sun, "Optimization of HMM by the tabu search algorithm," *J. Inf. Sci. Eng.*, vol. 20, no. 5, pp. 949–957, Sep. 2004.
- [7] J. Simonin and C. Mokbel, "Using simulated annealing expectation maximization algorithm for hidden Markov model parameter estimation," in *Proc. Eurospeech*, 1997, pp. 449–452.
- [8] A. M. Martinez and J. Vitria, "Learning mixture models using a genetic version of the EM algorithm," *Pattern Recognit. Lett.*, vol. 21, no. 8, pp. 759–769, Jul. 2000.
- [9] F. Pernkopf, D. Bouchaffra, P. Tefl, and C. Arth, "Genetic-based EM algorithm for learning Gaussian mixture models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 8, pp. 1344–1348, Aug. 2005.
- [10] N. Majdi-Nasab, M. Analoui, and E. J. Delp, "Decomposing parameters of mixture Gaussian model using genetic and maximum likelihood algorithms on dental images," *Pattern Recognit. Lett.*, vol. 27, no. 13, pp. 1522–1536, Oct. 2006.
- [11] B. Liu, L. Wang, and Y.-H. Jin, "An effective PSO-based memetic algorithm for flow shop scheduling," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 37, no. 1, pp. 18–27, Feb. 2007.
- [12] H. D. Nguyen, I. Yoshihara, K. Yamamori, and M. Yasunaga, "Implementation of an effective hybrid GA for large-scale traveling salesman problems," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 37, no. 1, pp. 92–99, Feb. 2007.
- [13] S.-M. Tse, Y. Liang, K.-S. Leung, K.-H. Lee, and T. S.-K. Mok, "A memetic algorithm for multiple-drug cancer chemotherapy schedule optimization," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 37, no. 1, pp. 84–91, Feb. 2007.
- [14] A. M. Martinez and J. Vitria, "Clustering in image space for place recognition and visual annotations for human-robot interaction," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 31, no. 5, pp. 669–682, Oct. 2001.
- [15] J. Yen, C. L. James, B. Lee, and D. Randolph, "A hybrid approach to modeling metabolic systems using a genetic algorithm and simplex method," *IEEE Trans. Syst., Man Cybern.*, vol. 28, no. 2, pp. 173–191, Apr. 1998.
- [16] Y.-S. Ong, M.-H. Lim, N. Zhu, and K.-W. Wong, "Classification of adaptive memetic algorithms: A comparative study," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 36, no. 1, pp. 141–152, Feb. 2006.
- [17] J. E. Smith, "Coevolving memetic algorithms: A review and progress report," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 37, no. 1, pp. 6–17, Feb. 2007.
- [18] D. Whitley, G. Scott, and M. Keith, "Lamarckian evolution, the baldwin effect and function optimization," in *Proc. PPSN*, 1994, pp. 6–15.
- [19] L. R. Rabiner, B. H. Juang, S. E. Levinson, and M. M. Sondhi, "Recognition of isolated digits using hidden Markov models with continuous mixture densities," *AT&T Tech. J.*, vol. 64, no. 6, pp. 1211–1222, 1986.
- [20] K. E. Mathias, D. Whitley, C. Stork, and T. Kusuma, "Staged hybrid genetic search for seismic data imaging," in *Proc. 1st IEEE Conf. Evol. Comput.*, Orlando, FL, 1994, pp. 356–361.
- [21] S. Kimura and A. Konagaya, "High dimensional function optimization using a new genetic local search suitable for parallel computers," in *Proc. IEEE Int. Conf. Syst., Man Cybern.*, 2003, pp. 335–342.
- [22] J. H. Holland, *Adaptation in Natural and Artificial Systems*. Ann Arbor, MI: Univ. Michigan Press, 1975.
- [23] D. Whitley, "A genetic algorithm tutorial," *Stat. Comput.*, vol. 4, pp. 65–85, 1994.
- [24] C. Houck, J. Joines, and M. Kay, "Utilizing Lamarckian evolution and the Baldwin effect in hybrid genetic algorithms," *Meta-Heuristic Res. Appl. Group*, Dept. Ind. Eng., North Carolina State Univ., Raleigh, NC, Tech. Rep. NCSU-IE TR 96-01, 1996.
- [25] Z. Michalewicz and M. Schoenauer, "Evolutionary algorithms for constrained parameter optimization problems," *Evol. Comput.*, vol. 4, no. 1, pp. 1–32, 1996.
- [26] H. Myung and J. H. Kim, "Multiple Lagrange multiplier method for constrained evolutionary optimization," *Int. J. Advance Comput. Intell.*, vol. 4, no. 2, pp. 158–163, 2000.
- [27] Y. Yoon, Y. H. Kim, and B. R. Moon, "An evolutionary lagrangian multiplier method for the 0/1 multiple knapsack problem," in *Proc. GECCO*, 2005, pp. 629–635.
- [28] D. E. Goldberg, *Genetic Algorithm in Search, Optimization and Machine Learning*. Reading, MA: Addison-Wesley, 1989.
- [29] H. Bresisni and F. J. Varela, "The immune recruitment mechanism: A selective evolutionary strategy," in *Proc. 4th Int. Conf. Genetic Algorithm*, 1991, pp. 520–526.
- [30] H. Bresini and G. Seront, "In search of a good evolution-optimization crossover," in *Proc. Parallel Problem Solving From Nature-2*, 1992, pp. 479–488.
- [31] L. R. Rabiner, B. H. Juang, S. E. Levinson, and M. M. Sondhi, "Some properties of continuous hidden Markov model representation," *AT&T Tech. J.*, vol. 64, no. 6, pp. 1251–1270, Jul./Aug. 1985.
- [32] F. K. Soong and T. Sevensen, "On the automatic segmentation of speech signals," in *Proc. ICASSP*, 1987, pp. 77–80.
- [33] H. Pholheim, "Genetic and evolutionary algorithm toolbox for use with MATLAB," Dept. Comput. Sci., Univ. Ilmenau, Ilmenau, Germany, 1998. Tech. Rep.
- [34] H. Muhlenbein, "Predictive models for the breeder genetic algorithm continuous parameter optimization," *Evol. Comput.*, vol. 1, no. 1, pp. 25–49, 1993.
- [35] S. J. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, D. S. Pallett, and N. L. Dahlgren, *TIMIT Acoustic-Phonetic Continuous Speech Corpus*. Philadelphia, PA: Linguistic Data Consortium.
- [36] L. R. Rabiner and B. H. Juang, *Fundamentals of Speech Recognition*. Englewood Cliffs, NJ: Prentice-Hall, 1993.
- [37] Y. S. Yung and Y. H. Oh, "A segmental-feature HMM for continuous speech recognition based on a parametric trajectory model," *Speech Commun.*, vol. 38, no. 1, pp. 115–130, Sep. 2002.
- [38] L. Gillick and S. Cox, "Some statistical issues in the comparison of speech recognition algorithms," in *Proc. IEEE ICASSP*, 1989, pp. 532–535.
- [39] L. Ingber and B. Rosen, "Genetic algorithms and very fast simulated reannealing: A comparison," *Math. Comput. Model.*, vol. 16, no. 11, pp. 87–100, 1992.
- [40] A. Soke and Z. Bingul, "Hybrid genetic algorithm and simulated annealing for two-dimensional non-guillotine rectangular packing problems," *Eng. Appl. Artif. Intell.*, vol. 19, no. 5, pp. 557–567, Aug. 2006.
- [41] V. S. Gordon and D. Whiteley, "Serial and parallel genetic algorithms as function optimizers," in *Proc. 5th Int. Conf. Genetic Algorithm*, 1993, pp. 177–183.
- [42] E. Cantu-Paz, "A survey of parallel genetic algorithms," Univ. Illinois, Urbana, IL, Tech. Rep. ILLGAL 97003, 1997.
- [43] E. Cantu-Paz, "Designing efficient master-slave parallel genetic algorithms," Univ. Illinois, Urbana, IL, Tech. Rep. ILLGAL 97004, 1997.



Shamsul Huda received the B.Sc.(Hons.) degree in electrical and electronic engineering from Khulna University of Engineering and Technology (KUET), Khulna, Bangladesh and the M.Sc. degree in computer engineering from Bangladesh University of Engineering and Technology, Dhaka, Bangladesh, in 2003. He is currently working toward the Ph.D. degree in the Center for Informatics and Applied Optimization, University of Ballarat, Ballarat, Victoria, Australia.

In 2003, he was with the Department of Computer Science, KUET, as a Faculty Member. He has published more than eight academic papers in pattern recognition area. His current research interest includes hybrid evolutionary computation, training algorithms for hidden Markov models, and their applications in signal processing and classification, sequential pattern recognition, and data mining.



John Yearwood received the B.Sc. degree from Monash University, Victoria, Australia, the M.Sc. degree from Sydney University, Sydney, Australia, and the Ph.D. degree from RMIT University, Melbourne, Australia.

In 1989, he was with the School of Information Technology and Mathematical Science, University of Ballarat, Ballarat, Victoria, as a Lecturer, where he was appointed as a Professor in 2007 and has been the Director of the Center for Informatics and Applied Optimization since 2007. His research interest includes modern optimization theory and techniques and their applications in pattern recognition, signal processing, and decision support systems. He has been the Chief Investigator on a number of ARC projects in these areas. He is an Associate Editor for the *Journal of Research and Practice in Information Technology*. He has published over 140 refereed journals, book chapters, and conference articles.



Roberto Togneri (M'89–SM'04) received the B.E. and Ph.D. degrees from the University of Western Australia (UWA), Perth, Australia, in 1985 and 1989, respectively.

Since 1998, he has been with the School of Electrical, Electronic and Computer Engineering, UWA, where he was first a Senior Tutor and later appointed to Lecturer in 1992 and then Senior Lecturer in 1997. He is a member of the Signals and Systems Engineering Research Group and heads the Signal and Information Processing Laboratory. His research

activities include signal processing and robust feature extraction of speech signals, statistical and neural network models for speech and speaker recognition, and related aspects of communications, information retrieval, and pattern recognition. He has published over 50 papers in the areas of spoken language and information systems and was the coauthor of the book *Fundamentals of Information Theory and Coding Design* (Chapman and Hall/CRC, 2003).