Fuzzy Model Based Recognition of Handwritten Hindi Numerals using Bacterial Foraging

M. Hanmandlu[1], A.V. Nath[2], A.C. Mishra[2] and V.K. Madasu[3]
[1]*Indian Institute of Technology, India;* [2]*Netaji Subhas Institute of Technology, India;*
[3]*Queensland University of Technology, Australia*

Abstract

This paper presents the recognition of Handwritten Hindi Numerals. The recognition is based on the modified exponential membership function fitted to the fuzzy sets derived from features consisting of normalized distances obtained using the Box approach. The exponential membership function is modified by two structural parameters that are estimated by optimizing the entropy subject to the attainment of membership function to unity. The optimization strategy used is the foraging model of E.coli bacteria. Two window sizes are used: one for (२, ९) and another for the rest of the numerals. Experimentation is carried out on a limited database of nearly 3500 samples. The overall recognition is found to be 96%.

1. Introduction

Most of the Indian scripts are distinguished by the presence of *matras* (or, character modifiers) in addition to main characters as against the English script that has no *matras*. Therefore, the algorithms developed specifically for English are not directly applicable to Indian scripts. Many OCRs for Indian scripts have been reported in [1,3,4,5,6]. However, none of these papers have considered Handwritten Hindi text consisting of composite characters involving *matras*. In this paper, we present a recognition system for handwritten Hindi numerals. However, the proposed recognition scheme is applicable to composite characters as well as to individual components after decomposition. The system has been tested on handwritten samples of Hindi numerals collected from different people.

Printed Devanagari character recognition is attempted based on KNN and Neural Networks [5, 1, 6]. These results are extended to Bangla [6], which also has the header line like Hindi. Structural features like concavities and inter-sections are used as features. A similar approach is tried for Gujarati [4] with limited success. Reasonable results are reported for Gurumukhi script [5]. Preliminary recognition results are also available in the literature on the two popular scripts in the south India – Tamil and Kannada [3]. Sinha et al. [2, 7] have reported various aspect ratios of Devanagari script recognition. Sethi and Chatterjee [8] have described Devanagari numeral recognition based on the structural approach. The primitives used are horizontal and vertical line segments, right and left slants. A decision tree is employed to perform the analysis depending on the presence or absence of these primitives and their interconnections. A similar strategy is applied to the constrained hand printed Devanagari characters in [9]. Neural network approach for isolated characters is reported in [10].

The perturbations due to writing habits and instruments are taken into account in the recognition of off-line handwritten English numerals in [12]. The back-propagation neural network is used in [11] for the recognition of handwritten characters. In that, feature extraction is done using three different approaches, namely, ring, sector and hybrid. The features consist of normalized vector distances and angles. The hybrid approach, which combines the ring and sector approaches, is found to yield the best results. The same features are adopted in [13]. We follow the same methodology for feature extraction as employed in [13]. The evolutionary computation technique, bacterial foraging is described in [17]. The foraging and evolutionary behavior of the E.coli bacteria is simulated using chemotaxis, reproduction and elimination-dispersal stages. We have used the bacterial foraging for the optimization of entropy associated with the modified membership function.

The paper is organized as follows. Section 2 deals with the feature extraction. In Section 3 the recognition system is presented. Section 4 reports the results and Section 5 gives the conclusions.

6th IEEE/ACIS International Conference on Computer and Information Science (ICIS 2007)
0-7695-2841-4/07 $25.00 © 2007 IEEE

2. Feature extraction

Preprocessing techniques like thinning [15], slant correction and smoothening are applied as in [16]. For extracting the features, the Box approach proposed in [13,14] is used here. This approach requires the spatial division of the character image. The major advantage of this approach stems from its robustness to variations, ease of implementation and high recognition rate.

Each character image is divided into 24 boxes so that the portions of a numeral will be in some of these boxes. There could be boxes that are empty, as shown in Fig. 1 in which numeral 3 is enclosed in the 6x4 grid. However, all boxes are considered for analysis in a sequential order. The choice of number of boxes is arrived at by experimentation. By considering the bottom left corner as the absolute origin (0,0), the coordinate distance (Vector Distance) for the k^{th} pixel in the b^{th} box at location (i,j) is computed as:

$$d_{kb} = (i^2 + j^2)^{1/2} \qquad (1)$$

By dividing the sum of distances of all black pixels present in a box with their total number, a Normalized Vector Distance (γ_b) for each box is obtained as:

$$\gamma_b = \frac{1}{n_b}\sum_{k=1}^{n_b} d_k^b,\ b=1,2,.....24 \qquad (2)$$

where, n_b is number of pixels in b^{th} box. These vector distances constitute a set of features based on distances.

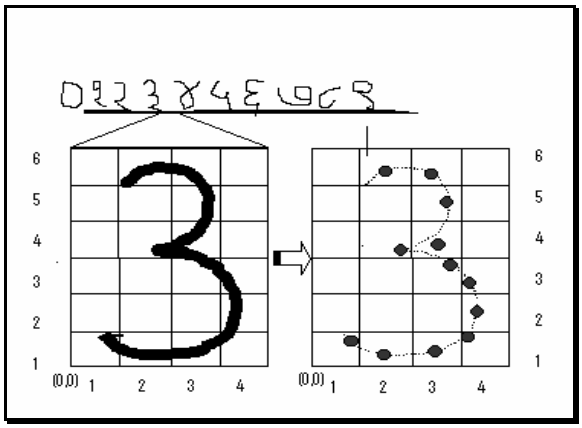

Figure 1. The handwritten Hindi numeral '3' enclosed using the box method

Therefore, 24 γ_b's corresponding to 24 boxes will constitute a feature set. However, for empty boxes the feature value will be zero.

3. Recognition scheme

In order to recognize the unknown numeral set using fuzzy logic, an exponential variant of fuzzy membership function is selected. The fuzzy membership function is constructed using the normalized vector distance. The concept of a fuzzy set arising from a set of features is now explained. If there are 'n' possible features for each numeral and if there are 'm' such samples, then a particular feature from each of the samples forms a fuzzy set. The means and variances are computed for each of the 24 fuzzy sets and these constitute the knowledge base (KB). Here, we use the training dataset which contains reference numerals for generating the KB.

3.1. Creation of KB and formulation of fuzzy membership function

The means and variances for each of the 24 fuzzy sets of KB are computed from the formulae:

$$m_i = \frac{1}{N_i}\sum_{j=1}^{N_i}\gamma_{ij},\ \sigma_i^2 = \frac{1}{N_i}\sum_{j=1}^{N_i}(\gamma_{ij} - m_i)^2 \qquad (3)$$

where,

N_i is the number of samples in the i^{th} fuzzy set with i=1,2,…,24 and γ_{ij} stands for j^{th} feature value of the reference character in the i^{th} fuzzy set (i.e. box).

For an unknown input numeral x, the 24 features are extracted using the Box method. The membership function is chosen as,

$$\mu_{x_i} = e^{\frac{-|x_i - m_i|}{\sigma_i^2}} \qquad (4)$$

where, x_i is the i^{th} feature of the unknown numeral.

If all x_i's are close to m_i's which represent the known statistics of a reference character, then the unknown numeral is identified with this known numeral because all membership function values are close to 1 and hence the average membership function is almost 1. Let, $m_j(r)$, $\sigma^2_j(r)$ belong to the r^{th} reference numeral with r = 0,1…9. We can then calculate the average membership as,

$$\mu_{av}(r) = \frac{1}{c}\sum_{j=1}^{c} e^{\frac{-|x_j - m_j(r)|}{\sigma_j^2(r)}} \qquad (5)$$

where, c denotes for the number of fuzzy sets. Then $x \in r$ if $\mu_{av}(r)$ is the maximum for r=0,1…9. It is observed that some of the fuzzy sets have a very small variance and others, a large variance. This has led to

the choice of a new membership function in [14] involving the structural parameters *s* and *t* given by,

$$\mu_{xi} = e^{-\Delta x_i^{'} / \sigma_i^{2'}} \quad (6)$$

where,

$$\sigma_i^{2'} = (1+t) + t^2\sigma_i^2$$
$$\Delta x_i^{'} = \left|(1-s) + s^2\Delta x_i\right|$$
$$\Delta x_i = \left|x_i - m_i\right|$$

The new mean and the new variance are functions of the mean and variance of the reference fuzzy set. Thus the structural parameters *s*, *t* models the variations in the mean and variance overall 24 boxes. The choice of these parameters has reasoning. That is, if *s*=1, $\Delta x_i^{'} = \Delta x_i$. Thus, *s* would be perturbed around 1 to reflect changes in the means. Similarly, if *t*=-1, then $\sigma_i^{2'} = \sigma_i^2$ thus, *t* would reflect the changes in the variances.

We need to select appropriate optimization (mainly minimizing) function to estimate the structural parameters. The form of this function can be taken as,

G = [Entropy function]+ λ [Some function of the modified membership function] (7)

Next, we eliminate λ by choosing G as,

G = [Some function of the modified membership function]*[Entropy function] (8)

The test function is chosen as,

J = [Some function of the modified membership function] (9)

Using the proposed modified membership function, different minimizing functions and the test functions will be experimented in Section 6 on results. We will now present a bacterial foraging for the purpose of minimization.

3.2. Bacterial Foraging (BF) minimization strategy

Bio-mimicry of foraging activities of Bacteria such as the E.coli provides a robust algorithm for distributed non-gradient global optimization. Natural selection tends to eliminate animals with poor foraging strategies and favor those that have good food locating and foraging capabilities through reproduction. Over a period of time the entire population is comprised of only the fittest animals (those having the best food gathering ability).

Generally, a foraging strategy involves finding a patch of food, deciding whether to enter it and search for food and when to leave the patch. Bacteria like the E.coli use the salutatory approach to foraging. They move about a region changing directions and stopping wherever there is a good concentration of food. The E.coli bacteria using its flagella can either run (move straight) or tumble (change direction). These allowable movements of the bacteria determine its chemo tactic actions.

Evolution of the bacteria leads to an ever more efficient population, in terms of foraging abilities. Each successive generation arises as a result of reproduction the previous generation's best lot, while those animals doing poorly are eliminated.

We can begin by assuming that ɵ is the instantaneous position of a bacterium. Then, $J(\theta) \in R^P$ represents an attractant profile, i.e., where nutrients are present or the medium is neutral or noxious for $J<0, J=0, J>0$ respectively. And 'p' is the dimensional spread of $J(\theta)$, viz., number of parameters for optimization. Let,

$$P(j,k,l) = \{\theta_i(j,k,l) | i = 0,1,.....S\} \quad (10)$$

represent the position of each member in a population of S bacteria at the j^{th} chemo tactic step, k^{th} reproduction step and the l^{th} elimination-dispersal event. Further, let $J(i,j,k,l)$ denote the cost at the location of the i^{th} bacterium $\theta_i(j,k,l) \varepsilon R^p$. To represent a tumble, a unit length random direction, say φ(j), is generated then,

$$\theta_i(j+1,k,l) = \theta_i(j,k,l) + C(i)\varphi(j) \quad (11)$$

such that a step of size C(i) is taken in the direction φ(j). If the cost $J(i,j+1,k,l)$ is better at $\theta_i(j+1,k,l)$ than at $\theta_i(j,k,l)$ then another chemo tactic step of size C(i) is taken in that direction and this may go on up to a maximum of N_s steps.

Each bacterium has a finite lifetime measured in number of chemo tactic steps. Let N_c be the length of bacteria measured in number of chemo tactic steps. Once it expires, it is time for a reproductive/elimination step. After the entire population has covered the requisite number of chemo tactic steps, a health assessment is done, whereby the cost functions of the bacteria are listed out in increasing order. The top half is chosen for reproduction while those in the bottom are eliminated. Reproduction involves splitting of the bacteria so that for every healthy bacterium (one that is doing more favorably in terms of foraging) two new bacteria are placed at the same location in the nutrient

plane. Thus during this process the population size remains the same.

Suppose there are, N_{re} reproduction steps. After N_{re} such steps, each having N_c number of chemo tactic steps, an elimination dispersal event is scheduled in which some members of the population are randomly relocated with probability P_{ed} .We may have N_{ed} number of such events.

3.3. Hindi Numeral Recognition using BF

Let us denote our optimization function by G = f(μ,other variable parameters present in G). As μ=f(s,t), G=f(s, t, other variable parameters present in G). Thus to minimize G, the parameters on which G depends, must be tuned. The structural parameters that are to be optimized are initially taken as random numbers on the optimization domain. These parameters are denoted by P(1,i,j,k,l) and P(2,i,j,k,l) respectively, for the i^{th} bacteria at the j^{th} chemo tactic step, k^{th} reproduction step and the l^{th} elimination-dispersal event. If any other parameters are included in the optimization function, the dimension of the search space increases correspondingly.

The total cost function, $J = \sum G_{si}$, where 'si' varies from 0 to number of images in training set (300 in our case). Now J is minimized by using FBF optimization strategy to tune the parameters. Program run is stopped once value of J function averaged for all bacteria's (AVGJ) falls within a predefined limit. The algorithm for the bacterial foraging optimization technique is given below.

Algorithm

I) *Initialization*:
1. Initialize the following parameters:
   i) Number of Chemotactic steps (N_c)
   ii) Number of Reproductive steps (N_{re})
   iii) Number of Bacteria (S)
   iv) Number of elimination & dispersal steps (N_{ed})
   v) Running Length which specifies the step size during one tumble/swim
   vi) Chemotactic behavior of the Bacteria: C(S,N_{re})
   vii) Maximum swim length (Ns) where $N_s<N_c$
   viii) Average minimized function for all Bacteria (AVGJ)
   ix) P_{ed}: Probability with which Ned will take place
   x) w: dimension of search space
   xi) flag: To break out of loop when satisfactory minimizing function is obtained.
2. Variables to be optimized are represented by P(:,i,j,k,l), which are initialized randomly within certain limits.
   where,
   i: corresponds to a particular Bacteria
   j: corresponds to a particular Chemotactic Step
   k: corresponds to a particular Reproduction Step
   l: corresponds to a particular Elimination and Dispersal step

II) *Iterative Optimization Algorithm*:
1) While flag=0
2) Elimination and dispersal loop: l = l+1
   ▪ If flag = 0
3) Reproduction loop: k = k+1
   ▪ If flag = 0
4) Chemotaxis loop: j = j+1
   ▪If flag = 0
5) Loop for each Bacteria: i = i+1
   ▪ If flag = 0
6) Loop run for each image: si = si+1
   ▪ If Lowerlimit < AVGJ < Upperlimit
   Set Flag=1
   Else J(i,j,k,l) = J(i,j,k,l)+ G_{si}
   ▪ If si < last (final image count), go to 6 (next image)
7) Chemotactic behavior for each bacterium
   i. Jlast = J(i,j,k,l)
   ii. Initiate a tumble in a random direction (Δ (:,i)) scaled by the running length, where, Δ (:,i) ε [-1,1]
   iii. Update position:
   P(:,i,j+1,k,l) = P(:,i,j,k,l)+C(i,k) × Δ (:,i)/sqrt(Δ (:,i) × Δ (:,i))
   iv. m = 0(counter for limiting length of swim)
   while m<N_s
   m = m+1
   If J(i,j+1,k,l)<Jlast
   Jlast = J(i,j+1,k,l)
   P(:,i,j+1,k,l)=P(:,i,j+1,k,l)+C(i,k) × Δ (:,i)/sqrt(Δ (:,i)'× Δ (:,i))
   v. If i < S, go to 5 (next bacterium)
8) Calculate value of AVGJ
   i. For B=1:S
   AVGJ = AVGJ+J(B,j,k,l)
   ii. AVGJ = AVGJ/10
   iii. If j<N_c, go to 4 (chemotaxis loop)

9) Reproduction
   a) Sort Bacteria in the order of ascending J values
   b) Carry over the chemotactic parameters and sort the bacteria positions according to indices obtained.
   c) Split healthy bacteria and eliminate half the population of less healthy bacteria
      Sr = S/2
      For i=1:Sr
        P(:,i+Sr,1,k+1,l)=P(:,i,1,k+1,l)
        C(i+Sr,k+1)=C(i,k+1)
   d) If k < Nr, go to 3 (reproduction loop)
10) Elimination and dispersion
   a) Based on P_{ed}, disperse some bacteria's and keep others in the same position.
   b) For m=1:S
      If P_{ed}>rand,
        P(:, m,1,1,l+1)=10 × rand(w,1)
      Else P(:,m,1,1,l+1)=P(:,m,1,Nre+1,l)
   c) If l < N_{ed}, go to 2 (Elimination-dispersion loop)
11) While loop
      If flag=1, save parameters and exit

4. Results

It is always a difficult task to compare the results of handwritten character recognition with the methods presented in the literature. The main problems that arise are the differences in the experimental methodology, experimental settings and the database used. For handwritten English numerals recognition, the benchmark dataset used by researchers is the CEDAR (Centre of Excellence for Document Analysis and Recognition) numeral database [1].

As there is no standard database available for Hindi numerals, a database of totally unconstrained handwritten numerals is considered for this work. This database was created from Hindi numeral samples extracted from railway reservation forms. The samples are of different writing styles with different sizes and stroke widths. The database also includes some complex samples that are even hard to recognize by careful inspection. The database is divided into two disjoint sets, one for training and another for testing.

The best results with a recognition rate of 96% were obtained with Shannon Entropy as tabulated in Table 1. In Table 2, the results with other entropy functions are listed. All minimizing functions presented in Table 2 gave better results with the Entropy used as the test function.

Table 1: Results using Shannon entropy in G

Numeral	S	T	Recognition rate (%)	Errors noted
0	1.079	9.860	98	7
1	1.073	10.82	100	-
2	0.910	10.89	92	3
3	0.889	11.05	90	2, 4
4	0.975	10.62	96	3
5	0.943	10.93	100	-
6	0.946	11.05	96	5,7
7	0.934	10.97	98	6
8	0.988	10.35	96	4
9	1.094	10.91	94	1

Net Recognition Rate= 96.0%, Average Running Time: 1 to 2 minutes, AVGJ: 308.3 to 308.7
Test function used for Recognition: J= $(\mu_{xj})^2$

Table 2: Recognition Rates with different minimizing functions

Minimizing Function	R.R (%)
$G = 1/(c\ln 2)(\Sigma-(\mu_{xj}\ln\mu_{xj} + (1-\mu_{xj}) \ln(1\mu_{xj}))) + \lambda J_1$ where, $J = \frac{1}{C}\sum_{j=1}^{c}\mu_{xj}$, $J_1 = [1-J]^2$, c=24	92.6
$G = (((\mu_{xj})^2 \ln\mu_{xj})/ \Sigma(\mu_{xj} \ln\mu_{xj}))^{-1}$	94.4
$G = (\Sigma(\text{Entropy} \times \text{Energy})/\Sigma\text{Entropy})^{-1}$ Entropy = $\ln\mu_{xj}$	94.0
$G = (\Sigma(\text{Entropy} \times \text{Energy})/\Sigma\text{Entropy})^{-1}$	93.4
$G = (\Sigma((\mu_{xj})^3 \ln\mu_{xj})/\Sigma\ln\mu_{xj})^{-1}$	95.0
$G = (\Sigma(\text{Entropy} \times \text{Energy})/\Sigma\text{Entropy})^{-1}$ Pal & Pal Entropy = $\mu_{xj}e^{(1-\mu xj)}$	92.6
$G = (\Sigma(\text{Entropy} \times \text{Energy})/\Sigma\text{Entropy})^{-1}$ Reny Entropy = $\ln(\mu_{xj})^{\alpha}$	93.8
$G = (\Sigma(\text{Entropy} \times \text{Energy})/\Sigma\text{Energy})^{-1}$ Shannon Entropy = $\mu_{xj}\ln\mu_{xj} + (1-\mu_{xj})\ln(1-\mu_{xj})$	92.0
$G = (\Sigma(\text{Entropy} \times \text{Energy})/\Sigma\text{Entropy})^{-1}$ Entropy = $x\mu_{xj}$	93.0

where, Energy= $(\mu_{xj})^2$

The recognition rate is found to increase with Shannon Entropy on decreasing the running time and increasing the tolerance for AVGJ to converge to the specified value. There is always a recognition problem between Hindi numerals 2 and 3, where 2's are mistaken as 3's and vice versa. The other problem is with the numerals 0 and 9. However these numerals show better recognition rates with Shannon Entropy function.

5. Conclusions

This paper introduces a new recognition scheme employing a new learning technique. It uses the box approach for feature extraction. In this each numeral is partitioned into 24 boxes from which 24 normalized distance features are extracted. These 24 features give rise to 24 fuzzy sets on collecting them over several samples. The knowledge base consists of means and variances of 24 fuzzy sets, which constitute the statistics of the reference numerals from the training dataset. We also use structural parameters s and t to take account of changes in the means and variances. These changes are likely to occur while testing the features of the unknown numerals to ascertain their identity using the known statistics of reference numerals in the knowledge base. Note that our approach uses two functions: one for estimating the parameters and another for testing the unknown numerals. The learning involves the choice of the optimization/minimizing function. The structural parameters are estimated using the optimization/ minimizing function. The unknown numerals are then tested using the test function making use of the learned parameters.

We have presented the bacterial foraging for learning the parameters in the recognition of Hindi numerals by the use of the test function. A set of optimization function and test functions has been explored leading to the particular combination that gives an overall 96% recognition rate which is obtained with Shannon entropy function and other entropy functions have failed to yield good results. The choice of different optimization function and test function is the main finding from the present study. Another outcome is that the parametric entropy function like Reny entropy didn't fare well over the non-parametric Shannon entropy function.

The advantage of bacterial foraging is that there is no need of initialization of parameters and no need to compute the derivatives of the optimization function with respect to parameters but we need to provide a bound on the optimization function. This can be done by making a few trail runs and getting a feel of the range of functional values during learning.

6. References

[1] http://www.cedar.buffalo.edu/ilt/research.html

[2] S.S. Marwah, S.K. Mullick and R.M.K. Sinha, "Character Recognition of Devanagari characters using a hierarchial binary decision tree classifier," Proc. IEEE International Conference on SMC, 1994, pp. 414-420.

[3] R. Bajaj, L. Dey and S. Chaudhury, "Devnagari numeral recognition by combining decision of multiple connectionist classifiers," Sadhana, Vol. 27 (1), 2002, pp. 59–72.

[4] S. Antanani and L. Agnihotri, "Gujarati character recognition," Proc. International Conference on Document Analysis & Recognition, 1999, pp. 418–421.

[5] V. Bansal and R.M.K. Sinha, "A Devanagari OCR and a brief review of OCR research for Indian scripts," Proc. STRANS, 2001.

[6] B. B. Chaudhuri and U. Pal, "An OCR system to read two Indian language scripts: Bangla and Devanagari (Hindi)," Proc. International Conference on Document Analysis & Recognition, 1997, pp. 1011–1015.

[7] R.M.K. Sinha and H.N. Mahabala, "Machine recognition of Devanagari script," IEEE Transactions on SMC, Vol. 9, 1979, pp. 435-441.

[8] I.K. Sethi and B. Chatterjee, "Machine recognition of Handprinted Devangari Numerals," Journal of Institution of Electronics and Telecommunication Engineers, Vol. 22, 1976, pp. 532-535.

[9] I.K. Sethi, "Machine recognition of constrained handprinted Devanagari," Pattern Recognition, Vol. 9, 1977, pp.69-75.

[10] J.C. Sant and S.K. Mullick, "Handwritten Devanagari Script recognition using CTNNSE algorithm," Proc. International Conference on Application of Information Technology in South Asian Language, 1999.

[11] M. Hanmandlu, K.R.M. Mohan and H. Kumar, "Neural-based handwritten character recognition," Proc. Fifth IEEE International Conference on Document Analysis and Recognition, 1999, pp. 241-244.

[12] T.M. Ha and H. Bunke, "Offline handwritten numeral recognition by perturbation method," IEEE Transactions on PAMI, Vol. 19(5), 1997, pp.535-539.

[13] M. Hanmandlu, K.R.M. Mohan, S. Chakrobarty, S. Goyal and D. Roy-Choudhary, "Unconstrained handwritten character recognition based on fuzzy logic," Pattern Recognition, Vol. 36(3), 2000, pp. 603-623,

[14] M. Hanmandlu, M.H.M. Yusof and V.K. Madasu, "Off-line signature verification and forgery detection using fuzzy modeling," Pattern Recognition, Vol. 38(3), 2005, pp. 341-356.

[15] R. Stefanelli and A. Rosenfeld, "Some parallel thinning algorithms for digital pictures", Journal of Association for computing Machinery, Vol. 18(2), 1971, pp. 255-264

[16] M. Hanmandlu and O.V.R. Murthy, "Fuzzy logic based handwritten Hindi numeral Recognition," Proc. International Conference on Cognition and Recognition, 2005.

[17] K.M. Passino, "Biomimicry of Bacterial Foraging for Distributed Optimization and Control", IEEE Control Systems Magazine, 2002, Vol. 22(3), pp 52-67.

[18] S. Mishra, "A Hybrid Least Square-Fuzzy Bacterial Foraging Strategy for Harmonic Estimation", IEEE Transactions on Evolutionary Computations, Vol. 9(1), 2005, pp. 61-73.

0-7695-2841-4/07 $25.00 © 2007 IEEE