Intelligent Feature Subset Selection with Unspecified Number for Body Fat Prediction based on Binary-GA and Fuzzy-Binary-GA

Farshid Keivanian
Department of Electrical and Computer Engineering,
University of Birjand, Birjand, South Khorasan, Birjand,
Iran
FarshidKeivanian@birjand.ac.ir , Info@keivanian.ir

Nasser Mehrshad
Department of Electrical and Computer Engineering,
University of Birjand, Birjand, South Khorasan, Birjand,
Iran
Nmehrshad@birjand.ac.ir

***Abstract*—knowing the body fat is an extremely important issue since it affects everyone's health. Although there are several ways to measure the body fat percentage (BFP), the accurate methods are often associated with hassle and/or high costs. Therefore, certain measurements or explanatory variables are used to predict the BFP. This study proposes an intelligent feature subset selection approach with unspecified number of features based on Binary GA and Fuzzy Binary GA algorithms to discover most important variable or feature and facilitate an artificial neural network (ANN) classifier model which is applied for body fat prediction (BFP). The proposed forecasting model is able to effectively predict the BFP with error of ± 3.64031% and the most effective feature of forearm circumference among total twelve features by using Fuzzy Binary GA.**

Keywords—intelligent feature subset selection; Binary GA; Fuzzy Binary GA; artificial neural network (ANN); body fat prediction

I. Introduction

The accurate measurement of the body fat percentage (BFP) is often associated with hassle and high costs since there are a variety of experiments for this purpose. To overcome this problem, BFP is predicted based on some certain measurements. In data mining point of view, the real patients and the measurements comprise the samples and related features of a data set. This real data set is fed into an artificial neural network (ANN) to train it for making a classifier model that can predict an accurate output associated with a new input data. Feature subset selection (FS) is the process of reducing the number of data features or equivalently the dimensions of input data in ANN classifiers. In general, the feature selection problem has not been proposed by a definitive solution yet. Motivated by this fact the process of feature selection will be turned into an optimization problem that is to be solved by Meta-heuristic algorithms in this article. The costs of data acquisition will be reduced when the smaller number of features is selected [1]. By using FS two important objectives are happened: (1) training of ANN classifier is facilitated, and (2) the most important features of input data are discovered [2]. Also the accuracy of the neural network which is trained by selected features is optimized by using Meta heuristic algorithms with a cost function. To quantify the quality of trained ANN classifier with the subset of features a cost function, z is defined. The Meta-heuristic algorithms are used to find a subset of most important features that allows a neural network to minimize the classifier error. There are different definitions for feature selection. The structure of classical feature selection with no intelligence is illustrated in Fig. 1. A subset of m features from a set of n original features, m < n, is selected by user, such that the classifier error is minimized over all subsets in size m [3]. The target vector, t, is originated from body fat data set. The ANN as a model and the body fat data set as a process are shown in Fig. 1 and Fig. 2:

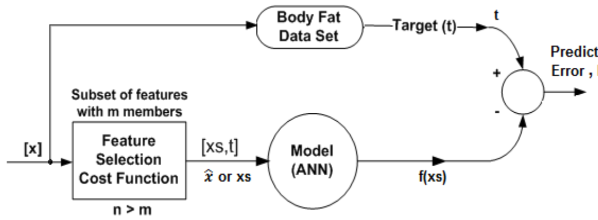

Fig. 1. Classical feature selection

Binary GA is used for intelligent feature subset selection to make the optimum ANN model with lower error as in Fig. 2:

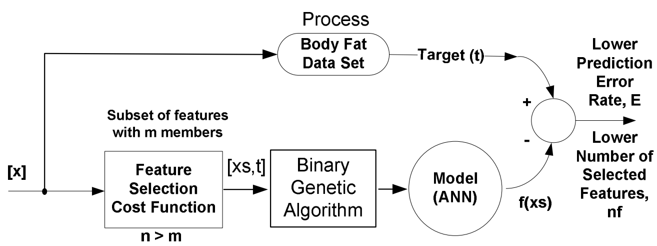

Fig. 2. Intelligent feature selection with unspecified number of features by Binary GA

II. Literature Review

While some typical studies have employed data mining techniques to classify the existence of certain diseases [4–6], the present study focuses on the development of intelligent forecasting techniques to effectively predict the BFP. The real body fat data from body fat percentage dataset of MATLAB toolbox is used in this article. These data contain the BFP which were determined by 13 body circumference measurements for 252 patients. Although one can employ the aforementioned variables to predict BFP through the use of multiple regression (MR) techniques or some machine learning approaches, the true relationship between these measurements and BFP may not be easy to determine. Several studies have used multiple regression techniques to build a forecasting model to estimate the BFP [7–9]. However, the MR models are criticized for the strong assumptions such as variation homogeneity [4]. In addition, some data mining techniques, such as artificial neural network (ANN), multivariate adaptive regression splines (MARS), and support vector regression (SVR), have become alternatives in modeling forecasting problems due to their capability to capture complex nonlinear relationships among variables [10–12]. Those data mining techniques have been reported to have better forecasting capability than the regression techniques [13–17].

To overcome the aforementioned difficulties and maintain the prediction accuracies of mentioned approaches for BFP, this study is aimed at proposing a forecasting model to predict BFP based on the idea of intelligent feature subset selection based on Binary GA and Fuzzy Binary GA.

The rest of this article is organized as follows: Section III illustrates the variables used in body fat data set. Section IV explains the structure of ANN model for BFP. Section V provides the mathematical basis of body fat prediction error and ratio of selected features to quantify the quality of ANN. Section VI, and VII demonstrate Binary-GA and Fuzzy Binary-GA. Section VIII shows the implementation results by MATLAB. Finally Section IX gives the conclusion.

III. BODY FAT DATA SET

The body fat dataset consists of 252 persons with 13 measurements for each one and it is summarized in Table I.

TABLE I. VARIABLES' DEFINITION IN THE BODY FAT DATASET

Variables	Meaning
x_1	Age (years)
x_2	Weight (pound, Ibs)
x_3	Height (inch)
x_4	Neck Circumference (cm)
x_5	Chest Circumference (cm)
x_6	Abdomen 2 Circumference (cm)
x_7	Hip Circumference (cm)
x_8	Thigh Circumference (cm)
x_9	Knee Circumference (cm)
x_{10}	Ankle Circumference (cm)
x_{11}	Biceps Circumference (cm)
x_{12}	Forearm Circumference (cm)
x_{13}	Wrist Circumference (cm)

This dataset can be used to train an ANN to estimate the body fat of someone from various measurements. This article examines if some of these features can be dropped while the accuracy of the body fat prediction is not affected seriously. Fig. 3 shows the inputs and outputs (targets):

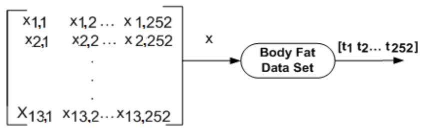

Fig. 3. Input and Output in Body Fat Data Set

The input matrix, x, is 13×252 represents 252 individuals, each one has 13 features. The output, t is the body fat target and it is a 1×252 vector. Developing an ANN solution requires previous knowledge of correct solutions for specific instances of the problem which is provided by body fat data set in this article. The data is usually split into three groups of training, testing, and validation type. In this article the test and validation data both are combined together with 2×36 = 72 elements. Always the test set consists of cases that are not used during the training phase being used for evaluation of the ANN generalization ability [18].

IV. THE ARTIFICIAL NEURAL NETWORK

For training an artificial neural network (ANN) a set of inputs and an associated set of target outputs are used. Once the ANN has fit the data, it forms a generalization of the input-output relationship and can be used to generate outputs for inputs it was not trained on. For this purpose, a two layer is feed forward network with sigmoid hidden neurons and linear output neurons is used. Levenberg-Marquardt back propagation algorithm is applied to train the neural network. This algorithm is better than the Bayesian Regularization and the Scaled Conjugate Gradient since it uses less memory in MATLAB. The standard steps for designing the ANN to be used in this article are including:

1. Collect data from body fat data set in MATLAB
2. Create the network
3. Configure the network
4. Initialize the weights and biases
5. Train the network
6. Validate the network
7. Use the network as a model for body fat prediction

The structure of ANN is illustrated in Fig. 4.

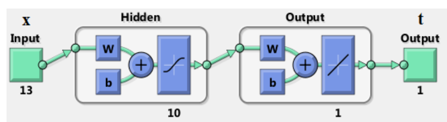

Fig. 4. A two-layer feed-forward ANN with 10 sigmoid hidden neurons and 1 linear output neuron

At first the body fat data set without any feature subset selection is applied to create and train the ANN. The total 252 samples are divided by two groups: 1. Training, 2. Combination of validation and test data as in Table II.

TABLE II. TRAINING, VALIDATION AND TESTING OF MLP

Dividing 252 Samples to Two Groups	
Training	***Combination of Validation and Test Data***
176 (70%)	76 (30%)

The evaluation of ANN performance trained without any feature subset selection is shown in Fig. 5:

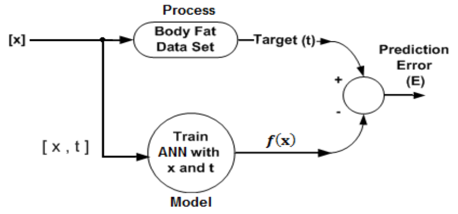

Fig. 5. Training the ANN with data set without any feature selection

The error of each sample is defined as the difference between the output of the ANN model, f(x) and the output of the process or body fat data set, t which is shown in (1):

$$e = |t - f(x)|^2 \tag{1}$$

The mean squared error for training data with N_{Train} samples is obtained in (2). Also for test data with N_{Test} samples the MSE is measured as in (3):

$$E_{Train} = MSE_{Train} = \frac{1}{N_{Train}} . \sum_{i=1}^{N_{Train}} e_i^2 \tag{2}$$

$$E_{Test} = MSE_{Test} = \frac{1}{N_{Test}} . \sum_{i=1}^{N_{Test}} e_i^2 \tag{3}$$

The total mean squared error, MSE_{Total} for all training and test data is evaluated in (4) and after substitutions, (5) is obtained as:

$$E_{Total} = MSE_{Total} = w_{Train} \times E_{Train} + w_{Test} \times E_{Test} \tag{4}$$

$$E_{Total} = MSE_{Total} = 0.8 \times \frac{1}{176} \sum_{i=1}^{176} e_i^2 + 0.2 \times \frac{1}{76} \sum_{i=1}^{76} e_i^2 \tag{5}$$

Equation (5) is obtained by substitution of wTrain = 0.8 and wTest = 0.2.

The pseudo codes for calling body fat data set and training ANN are shown in Fig. 6 and Fig. 7. The program in Fig. 7 will make an artificial neural network by receiving x and t. The ANN will be created and trained just when it receives a set of inputs, x and the target, t. The program in Fig. 7 is called by the program of Fig. 6:

- Call bodyfat_dataset with all 13 features for 252 samples
- Return the results to "[x , t]"
- Call CreateAndTrainANN with input parameters (x , t)
- Return the results e (252 errors) and E (Mean Squared Error)
- End

Fig. 6. Pseudo code: calling CreateAndTrainANN with all features (without feature selection)

The total 252 samples are divided by two groups. One has 176 samples used for training, and another has 76 samples applied for combined test and validation data. In this part no feature selection is done yet and all 13 features of body fat data set are considered for creating, training and testing ANN.

Therefore only one objective function is defined here. There are some modifications in Fig. 7 such that if x is not empty the program continues otherwise the infinitive values will be displayed for the results. By this modification, this bug will be avoided.

- **SUBMODULE** < function CreateAndTrainANN (x , t) >
  - **IF** x is not empty
    - Select Levenberg-Marquardt as training function
    - Create hidden layer with size 10
    - Create a neural network with the name of net
    - Specification of preprocessing functions
    - Division of Input Data : 70% for Training, 15% for Validation and 15% for Testing
    - Choose the performance function of Mean Squared Error (MSE or E)
    - Show off the window of training ANN
    - Train the Network with the name of net
    - Test the Network of net by : y = net(x)= f(x) , e = t – f(x)
    - Calculate the Performance of net : E = Sum (e) / Total Samples
  - **ELSE**
    - Set infinitive values to y, e, and E
  - **ENDIF**
    - Set all resulting and input data into variables for all 252 samples : Data.x=x, Data.t=t, Data.y=y, Data.e=e, Data.E=E
    - Validation and Test Data both are combined as a group of data
    - Return error (e) and mean squared error (E) as Results
- **END SUBMODULE**

Fig. 7. The Pseudo-Code for creating and training the Artificial Neural Network with all features (without feature selection)

The implementation results are shown in Table III:

The training data has lower E than the combined validation and test data. However by increasing the number of neurons in ANN the better results can be obtained.

TABLE III. MSE (E) FOR ALL GROUPS OF DATA IN ANN WITHOUT ANY FEATURE SELECTION

Data	Samples	MSE (E)	E_{Total}
Training Data	176	E_{Train} =13.1582	E_{Total} = 14.76194
Combined Validation and Test Data	76	E_{Test} = 21.1769	

V. THE OBJECTIVE FUNCTIONS OF INTELLIGENT FEATURE SUBSET SELECTION

Many different evaluation functions have been used for the feature selection. A categorization of these functions according to their theoretical basis is firstly proposed by Dash and Liu in 1997 [19]. The categories are including: distance, information, dependency, consistency, and classification error measurements. In this article the total prediction error, E_{Total} and the number of selected features, n_f both are presented as the objective functions. Their combination is defined as the cost function, z, which is obtained through the equations (6) to (14). Also (6) shows the difference of target, t and the output of the ANN, f (xs), that presents the error, e:

$$e = |t - f(xs)|^2 \tag{6}$$

For each sample the error is shown with index of i in (7):

$$\begin{cases} y_i = f(xs_i) \\ e_i = t_i - y_i \end{cases} \tag{7}$$

The cardinality of the sub set of xs is in (8) that shows the number of selected features, n_f. It is another objective function in this article. Both E_{Total} in (5) and n_f (the number of selected features) in (9) are to be minimized in this article.

$$n_f = \|x_s\| \tag{8}$$

The linear combination of these objective functions; (5), and (9) is represented in (10). The coefficients w_1 and w_2 are used for the cost function, z:

$$z = w_1.E_{Total} + w_2.n_f \tag{9}$$

If (10) is divided by w_1 then (11) with only one coefficient is obtained:

$$z \approx E_{Total} + \frac{w_2}{w_1}.n_f = E_{Total} + w.n_f \tag{10}$$

The weighting factor w is chosen the way that $w.n_f$ becomes proportional to E_{Total} in scale. Equation (12) provides this matter.

$$w \propto E_{Total} \rightarrow w = \beta.E_{Total} \tag{11}$$

Based on (11) and (12) the resulting cost function, z is obtained in (13) which is to be minimized by the Meta heuristic algorithm of Binary-GA in this article:

$$z = E_{Total} + \beta.E_{Total}.n_f \tag{12}$$

The important parameter of β is independent of E_{Total} and the value of 0.04 is used for β. After simplification, (14) is obtained:

$$z = E_{Total}.(1 + \beta.n_f) \tag{13}$$

The lower value for n_f means the higher value for E_{Total} because these two objective functions are in conflict with each other. Some of features that are more important are selected with the number of n_f. This number is not exactly known and only the parameter of β affects on the minimization of z. The parameter β determines the balance point between two objective functions. When it decreases, the number of features n_f is increased while the mean squared error, E_{Total} is decreased. The minimization of z causes the minimization of both E_{Total} and n_f . The goal of this article is to create and train an ANN classifier with minimum number of features, n_f and minimum total error, E_{Total}. In some articles the ANN is usually trained only one time but in this article it is trained 5 times to increase the reliability and accuracy of the ANN results. This manipulation is provided by a loop with nRun iterations for creating and training the ANN which is similar to Monte Carlo simulation. The resulting E is calculated as in (14):

$$E = Average(EE) = \frac{1}{nRun}\sum_{i=1}^{nRun} E^i_{Total} \tag{14}$$

After substitution of nRun = 5, (15) is obtained:

$$E = Average(EE) = \frac{1}{5}.\sum_{i=1}^{5} E^i_{Total} \tag{15}$$

In fact the real number of iterations is obtained by the multiplication of ANN train number (nRun) and the population size of the Meta heuristic algorithm. When the number of features, n_f is divided by the total number of features, n_x the ratio of the feature subset selection, r_f, which is a number between 0 and 1 is obtained as in (16):

$$r_f = \frac{\text{Number of selected features}(n_f)}{\text{Total number of features}(n_x)} = \frac{n_f}{13} \tag{16}$$

Since r_f is 13 times lower than n_f then β should be chosen 13 times bigger than its previous value which is shown in (17).

$$r_f = \frac{1}{13}n_f \tag{17}$$

$$\beta_{new} = 13 \times \beta_{old} = 13 \times 0.04 = 0.52 \tag{18}$$

$$z = E_{Total}.(1 + \beta_{new}.r_f) = E_{Total}.(1 + 0.52 \times r_f) \tag{19}$$

The mathematical basis for using r_f instead of n_f is shown through the equations (20) to (27). Equation (20) shows the cost, z_1 for the ratio of feature selection r_1 and the total error of E_1. Also the second state of selection is shown in (21). The substitutions are done through the equations (22) to (27). And the resulting equation is obtained in (27). Equation (27) shows that with a reduction in r_f how much the prediction error, E will be increased. By setting the values of β and Δr if the amount of increase in E (mean squared error) becomes equal or lower than ΔE then reducing the number of features has been a positive work. The maximum acceptable increase in E is defined as ΔE in (27) that can be achieved for the maximum decrease in r_f , Δr:

$$z_1 = E_1 \times (1 + \beta.r_1) \tag{20}$$

$$z_2 = E_2 \times (1 + \beta.(r_1 - \Delta r)) \tag{21}$$

$$E_1 \times (1 + \beta.r_1) = E_2 \times (1 + \beta.(r_1 - \Delta r)) \tag{22}$$

$$r_2 = r_1 - \Delta r \ , \ E_2 = E_1 + \Delta E \tag{23}$$

$$E_1 \times (1 + \beta.r_1) = (E_1 + \Delta E) \times (1 + \beta.(r_1 - \Delta r)) \tag{24}$$

$$E_1 + E_1\beta.r_1 = E_1 + E_1\beta r_1 - E_1\beta\Delta r + \Delta E.(1 + \beta.(r_1 - \Delta r)) \tag{25}$$

$$E_1\beta\Delta r = \Delta E.(1 + \beta.r_1 - \beta\Delta r) \tag{26}$$

$$\Delta E = \frac{E_1\beta\Delta r}{(1 + \beta.r_1 - \beta\Delta r)} \tag{27}$$

Equation (27) shows that the changes in body fat prediction error, ΔE is proportional to the amount of error, E1, parameter β, and the decrease amount in the number of features, Δr.

The relationship between Δr and ΔE is shown in (28).

$$\Delta r \downarrow \Rightarrow \Delta E \uparrow \tag{28}$$

Equations (23) and (28) indicate that by decreasing Δr, the number of features is increased yielding the increase in ΔE, and the amount of error becomes more.

VI. BINARY GENETIC ALGORITHM (BINARY-GA)

The Binary-GA begins, like any other optimization algorithm, by defining the optimization variable, and the cost function. It ends like others too, by testing for convergence. In between, however, the Binary-GA is quite different [20]. The fitness evaluator is the program of Fig. 8 and the Binary-GA is in Fig. 9.

```
- SUBMODULE < function FeatureSelectionCost ( x , t , s ) >
    - Make the vector S based on the nonzero elements of vector s
    - Count the number of S elements as selected features, "nf = numel (S)"
    - Calculate the ratio of the selected features by "rf=nf/numel(s)"
    - Feature Selection by "xs=x(S,:)"
    - Calculate the weighting coefficients for errors of Train and Test Data
      by "wTrain=0.8 , wTest=1-wTrain"
    - Define the number of loop iterations for training ANN, "nRun=5"
    - Define EE as prediction error for each iteration of ANN
  - For r <= nRun
    - Call CreateAndTrainANN with input parameters ( xs , t )
    - Calculate the prediction error for each iteration by,
     "EE(r) = wTrain*results.TrainData.E + wTest*results.TestData.E"
  - END
    - Calculate the average prediction error, E, nf, rf
    - Calculate the Cost Function, z = E * (1 + beta * rf)
    - Return the cost function z, the prediction error E, and the ratio of
    feature selection, rf to Results "[ z, nf, rf, E, xs] "
- END SUBMODULE
```

Fig. 8. The Pseudo-Code of the Feature Selection Cost

```
- Call bodyfat_dataset
- Return its results to "[ x , t ]"
- Make the vector, s with random integer values either 0 or 1, in the size of nx
- Call FeatureSelectionCost with the input parameters ( x , t , s )
- Return its results to "[ z, nf, rf, E, xs ]"
- Define the number of variables as the number of features,"nVar=data.nx"
- Set GA Parameters, MaxIt, nPop, pc, pm, and mu
  - FOR i <= nPop
     - Generate initial population
     - Evaluation with Cost Function
  - END
- Sort the Population based on Cost value
- Store Best Solution
- Store Best Cost and Worst Cost value
- WHILE  it < = Maxit
    - FOR k < = nc/2
        - Select Parents Indices
        - Select Parents
        - Do Crossover
        - Evaluate the Off springs
    - END (k)
    - FOR k < = nm
        - Select Parent
        - Do Mutation
        - Evaluate the Mutant
    - END (k)
    - Create Merged Population
    - Sort Population
    - Update worst Cost
    - Truncation
    - Store Best Solution Ever Found
    - Store Best Cost Ever Found
- ENDWHILE
- Show the best results  ("Best Cost in each iteration")
```

Fig. 9. The Pseudo-Code of the Binary Genetic Algorithm for intelligent feature selection with unspecified number of features

The program of Fig. 8 calls the program of Fig. 9.

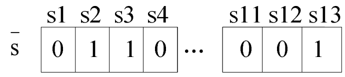

Fig. 10. The chromosomes in Binary-GA

$$\text{Cost Function}: z = f(\bar{s}) = f(s_1, s_2, \ldots, s_{12}) \quad (29)$$

The chromosomes as the search agents are shown in Fig. 10 and Fig. 9 is called to evaluate the cost function in (29). Fig. 9 returns the results including the number of selected features, n_f, the ratio of selected features, r_f, the total mean squared error, E_{Total}, and the selected features, xs. Binary-GA is in Fig. 9. The vector s has random integers either zero or one. The number of selected features is not specified in advance. The goal is to find the least number of features, n_f that provides the least amount of prediction error, E_{Total} and subsequently the least cost function, z. The binary chromosomes are the elements in s vector that have off springs with each other and their size is the same as nx. So the number of genes in each chromosome, nVar is the same as the number of features, nx in body fat data set. Based on Fig. 10 the values of genes are set either one or zero randomly. By Fig. 8 the different artificial neural networks are trained until ANN meets the lowest possible value for E_{Total} and z.

VII. FUZZY BINARY GENETIC ALGORITHM (FUZZY BINARY-GA)

Fuzzy Binary-GA is shown in Fig. 11. The parameters of mutation ("pm=0.3"), and crossover percentage ("pc=0.8"), also the number of population ("nPop=20") all are optimally controlled to get the better results [21]. The Fuzzy-Binary-GA rules are illustrated in Table IV [22–24]. The pseudo code of Fuzzy Binary-GA is in Fig. 12. The membership functions are trapmf type. The implementation results are shown in part VII.

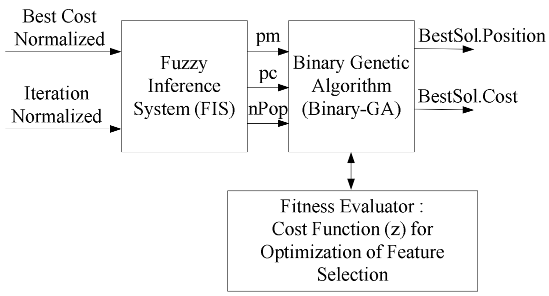

Fig. 11. The block diagram of Fuzzy Binary-GA system used in this article

TABLE IV. FUZZY RULES IN FUZZY BINARY-GA

If	Then
Itnormalized is High and BestCostnormalized is High	pm is High and pc is Low and nPop is High
Itnormalized is High and BestCostnormalized is Low	pm is Low and pc is High and nPop is Low
Itnormalized is Low and BestCostnormalized is High	pm is High and pc is Medium and nPop is High
Itnormalized is Medium and BestCostnormalized is Medium	pm is Medium and pc is Medium and nPop is Medium

- Call bodyfat_dataset
- Return its results to "[x , t]"
… # the same as Box 4 (Binary-GA)
- **WHILE** it <= Maxit
… # the same as Box 4 (Binary-GA)
- Normalize the Iteration and the BestCost value
  itnormalized = it / Maxit,
  BestCostnormalized = [WorstCost -- BestCost(it)] / WorstCost
- Read Fuzzy Inference System File and Fuzzy Rules (Fuzzy_BGA_FIS.fis)
- Define Input Variables for FIS and Fire the Rules
- **ENDWHILE**
- Show the best results ("Best Cost in each iteration")

Fig. 12. Fuzzy-Binary GA Pseudo-Code similar to Binary-GA

VIII. Implementation Results and Discussion

The obtained implementation results by MATLAB are illustrated in Table V. The objective functions, nf and E_{Total} and the cost function, z all are minimized in three states: **1.** Without using any loop for training of ANN and without using Binary-GA, **2.** With using loop for training of ANN and using Binary-GA, and **3.** With using loop for training of ANN and using Fuzzy-Binary-GA and the maximum iteration of 100.

TABLE V. The Comparison Of The Results

Results	1	2	3
Number of Features (n_f)	6	1	1
Ratio of Selected Features	46 %	7.69 %	7.69 %
Selected Features, S	x_2 x_6 x_8 x_{10} x_{11} x_{12}	x_{12}	x_{12}
PPE	± 4.06787 %	± 3.73816 %	± 3.64031%
Prediction Error (E_{Total})	16.5475	13.9739	13.2519
Cost Function: z	115.8327	14.5329	13.7820

The percentage of prediction error (PPE) for body fat is calculated by (30):

$$PPE = \pm\sqrt{E_{Total}}\ \% \qquad (30)$$

The minimization of the cost function, z is iteratively done by Binary-GA in blue and Fuzzy-Binary-GA in green after 100 iterations and their plotted in Fig. 13:

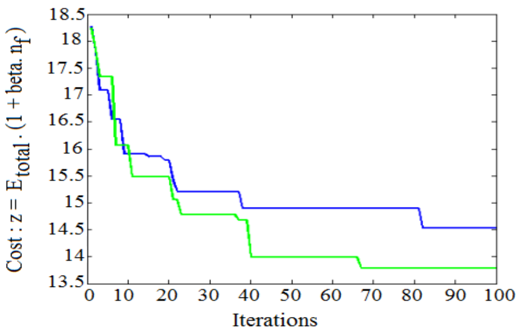

Fig. 13. the convergence of the cost function, z, by using Binary GA and Fuzzy Binary GA

Based on Fig. 13 and Table V, even though six features are selected as in column **1,** but the cost function, z which is combination of selected features ratio and prediction error is more than others. The promising results obtained by Fuzzy Binary GA in column number **3** are the best since fuzzy inference system has improved the accuracy and convergence speed of algorithm. After 100 iterations, the most effective feature of x_{12} is discovered and more accurate PPE ± 3.64031%

IX. Conclusion

The feature subset selection is intelligently optimized by Binary GA and Fuzzy BGA in body fat prediction problem. An artificial neural network with reduced features of BFP is trained then its forecasting performance is evaluated through the cost function, z presenting a linear combination of two objective functions: 1. the ratio of selected features, r_f, and 2. the total error, E_{Total}. By iterative meta-heuristic algorithms the ratio of selected features, r_f, the prediction error, E_{Total}, and subsequently the cost function z are minimized. Also Fuzzy Binary GA showed more accurate result in lower time by using the fuzzy rules for convergence improvement. Finally, the least percentage of prediction error of 3.64% with the lowest cost function of 13.782, and the minimum ratio of selected features of 7.69 % are obtained by Fuzzy Binary GA with the most important feature of forearm circumference, x12.

Acknowledgment

Appreciations are extended to the following esteemed individuals: Professor Seyed-Hamid Zahiri, and Professor Seyed Mohammad Razavi at University of Birjand.

References

[1] Antonio Arauzo-Azofra, J. M. Benitez, J. L. Castro, "Consistency measures for feature selection", Journal of Intelligent Information Systems, 30:273–292, 2008.

[2] R. Nilsson, J. M. Pena, J. Bjorkegren, and J. Tegner "Consistent Featrure Selection for Pattern Recognition in Polynomial Time", Journal of Machine Learning Research, 8,pp. 589–612, 2007.

[3] I.S. Jacobs and C.P. Bean, "Fine particles, thin films and exchange anisotropy," in Magnetism, vol. III, G.T. Rado and H. Suhl, Eds. New York: Academic, pp. 271–350, 1963.

[4] Y.E.Shao, C. D. Hou, and C.C.Chiu,"Hybridintelligent modeling schemes for heart disease classification,"Applied Soft Computing, vol.14, partA, pp.47–52, 2014.

[5] J. de Jes´ us Rubio, F. Ortiz-Rodriguez, C. R. Mariaca-Gaspar, andJ.C.Tovar,"Amethodforonlinepatternrecognition of abnormal eyemovements,"Neural Computing andApplications, vol. 22, no. 3-4, pp. 597–605, 2013.

[6] D. M. Vazquez, J. J. Rubio, and J. Pacheco, "A characterization framework for epileptic signals," IET Image Processing, vol.6, pp.1227–1235, 2013.

[7] O.L.Svendsen, J. Haarbo,B.L.Heitmann, A. Gotfredsen,and C. Christiansen, "Measurement of body fat in elderly subjects by dual-energy x-ray absorptiometry, bioelectrical impedance, and anthropometry," The American Journal of Clinical Nutrition, vol. 53, no.5, pp. 1117–1123, 1991.

[8] D. S. Freedman, W. H. Dietz, S. R. Srinivasan, and G. S. Berenson, "Risk factors and adult body mass index among overweight children: the bogalusa heart study," Pediatrics,vol. 123, no. 3, pp. 750–757, 2009.

[9] T. Sakai, S. Demura, and K. Fujii, “Validity of fat percentage evaluation for fat-thin judgement derived fromstandardweight in preschool children,”Sport Sciences for Health, vol.5, no.3, pp. 105–112, 2009.

[10] Y. E. Shao and C. D. Hou, “Fault identification in industrial processes using an integrated approach of neural network and analysis of variance,” Mathematical Problems in Engineering,vol. 2013, Article ID 516760, 7 pages, 2013.

[11] C. J. Lu and Y. E. Shao, “Forecasting computer products sales by integrating ensemble empirical mode decomposition and extreme learningmachine,”Mathematical Problems inEngineering,vol.2012, ArticleID831201, 15pages, 2012.

[12] R. Abirami and M. S. Vijaya, “Stock price prediction using support vector regression,”Communications in Computer and Information Science, vol. 269, pp. 588–597, 2012.

[13] X. Ren and X. Lv, “Identification of extended hammerstein systems using dynamic self-optimizing neural networks,” IEEE Transactions on Neural Networks,vol.22, no.8,pp.1169–1179, 2011.

[14] J. H. Perez-Cruz, J. J. Rubio, E. Ruiz-Velazquez, and G. SolisPerales, “Tracking control based on recurrent neural networks for nonlinear systems with multiple inputs and unknown deadzone,”Abstract and Applied Analysis,vol.2012, Article ID 471281,18pages, 2012.

[15] J. J. Rubio, “Modified optimal control with a backpropagation network for robotic arms,”IET Control Theory and Applications, vol. 6, pp. 2216–2225, 2012.

[16] J. P. Donate, X. Li, G. G. S´ anchez, and A. S. de Miguel, “Time series forecasting by evolving artificial neural networks with genetic algorithms, differential evolution and estimation of distribution algorithm,”Neural Computing and Applications, vol. 22, no. 1, pp. 11–20, 2013.

[17] Y. E. Shao, “Prediction of currency volume issued in Taiwan using a hybrid artificial neural network andmultiple regression approach,” Mathematical Problems in Engineering,vol.2013, Article ID 676742, 9 pages, 2013.

[18] Manuel R. Barbosa, Teresa Amaral, Maria F. Chouzal, Maria T. Restivo, "NEURAL NETWORKS BASED APPROACH TO ESTIMATE BODY FAT", Symposium on Instrumentation, modeling and control, 2010.

[19] Dash, M., & Liu, H., "Feature Selection for classification", International Journal of Intelligent Data Analysis, Elsevier, 131-156, 1997.

[20] R. L. Haupt, S. E. Haupt, "Practical genetic algorithms", second edition, Wiley Interscience Publication, New Jersey, 2004.

[21] Mitchell, Melanie, "An Introduction to Genetic Algorithms," Cambridge, MA: MIT Press. ISBN 9780585030944 , 1996.

[22] F. Keivanian, N. Mehrshad, S. H. Zahiri, "Optimum layout of multiplexer with minimal average power based on IWO, Fuzzy - IWO, GA, and Fuzzy GA", ACSIJ Advances in Computer Science: an International Journal, Vol. 3, Issue 5, No. 11, September 2014.

[23] F. Keivanian, "Minimization of Average Power Consumption in 3 Stage CMOS Ring Oscillator based on MSFLA, FuzzyMSFLA, GA, and Fuzzy-GA", International Journal of Computer Applications (0975–8887), Vol. 104, No. 16, October 2014.

[24] F. Keivanian, A. Yekta, and N. Mehrshad, "Optimization of JK Flip Flop Layout with Minimal Average Power of Consumption based on ACOR, Fuzzy-ACOR, GA, and Fuzzy-GA," Vol. 14, Issue 1, pp. 1–15, November 2014.