Design and Identify Tubercle Bacilli Diagnosis System with TSK-type Neuro Fuzzy Controllers

[1]*Hsien-Tse Chen and [2] Sheng-Fuu Lin and [3] Yung -Chi Hsu

[1,2,3]*Department of Electrical Engineering, National Chiao Tung University, Taiwan 30010, R. O. C.*

[1]*Hsientse.ece93g@nctu.edu.tw

Abstract

This paper proposes a TSK-type Neuro Fuzzy Controllers (TFC) with a group interaction-based evolutionary algorithm (GIEA) for constructing the tubercle bacilli diagnosis system (TBDS). The proposed GIEA is designed basing on symbiotic evolution which each chromosome in the population represents only partial solution. The whole solution consists of several chromosomes. The GIEA is different from the traditional symbiotic evolution. Each population in the GIEA is divided into several groups. Each group represents a set of the chromosomes that belong to only one fuzzy rule. Moreover, in the GIEA, the interaction ability is considered that the chromosomes will interact with other groups to generate the better chromosomes by elites-base interaction crossover strategy (EICS). In the TBDS, the EICS is used to train the TBDS. After trained by the EICS, the TBDS can diagnose the visible tubercle bacilli. The performance of the GIEA achieves better than other existing models in tubercle bacilli.

Keywords: Neuro Fuzzy controllers, tubercle bacilli, symbiotic evolution, reinforcement learning.

1.Introduction

The evolutionary fuzzy model generates a fuzzy system automatically through incorporating evolutionary learning procedures [1-6], where the well-known procedure is the genetic algorithms (GAs). In [1], Karr applied GAs to the design of the membership functions of a fuzzy controller, with the fuzzy rule set assigned in advance. Many researchers have applied GAs to optimize both the parameters of the membership functions and the rule sets [2] on the basis of. Lin and Jou [3] proposed GA-based fuzzy reinforcement learning to control magnetic bearing systems. In [4], Juang et al. proposed genetic reinforcement learning in the design of fuzzy controllers. The GA adopted in [4] was based upon traditional symbiotic evolution which, when applied to fuzzy controller design, complemented the local mapping property of a fuzzy rule. In [5], Tang proposed a hierarchical genetic algorithm. The hierarchical genetic algorithm enables the optimization of the fuzzy system design for a particular application. In [6], Lin proposed a hybrid evolution learning algorithm (HELA). The HELA combines the compact genetic algorithm (CGA) and the modified variable-length genetic algorithm to perform the structure/parameter learning for constructing the network dynamically. However, these approaches encounter one or more problems as below: 1) all the fuzzy rules are encoded into one chromosome; 2) the population cannot evaluate each fuzzy rule locally.

Recently, Fuzzy Controllers have been applied to diagnosis system [7-10] in several researches. Genetic algorithm optimized fuzzy neural network (GA-FNN) proposed by Levente et al. [7] was trained on a dataset of 177 HIV-1 protease ligands with experimentally measured IC50 values. Benamrane et al. [8] propose an approach for detection and specification of anomalies present in medical images. The idea is to combine three metaphors: Neural Networks, Fuzzy Logic and Genetic Algorithms in a hybrid system. The advanced fuzzy cellular neural network (AFCNN), Wang et al. [9] proposed is a variant of the fuzzy cellular neural network (FCNN) and is proposed to effectively segment CT liver images. The Neuro-Fuzzy Controller (NFC) proposed by Jafar

et al. [10] has position control of robot arm. A five layer neural network is used to adjust input and output parameters of membership function in a fuzzy logic controller.

In this paper, as same with [7-10], we also proposed Fuzzy Controllers to diagnosis system. Therefore, TSK-type Neuro Fuzzy controllers (TFC) with a group interaction-based evolutionary algorithm (GIEA) is proposed for constructing the tubercle bacilli diagnosis system (TBDS). In TBDS, testing for acid-fast bacilli (AFB) often needs a lot of time and manpower to read a slide with phlegm merely through naked eyes and a microscope. Generally speaking, this process is time consuming and exhausting, and is likely to produce incorrect results. To reduce errors and increase analysis efficiency, this study proposes image technology and identification methods to count tubercle bacilli [11].

This paper applies digital image to identify and count tubercle bacilli on a slide through a microscope by automatic image processing and pattern recognition. This paper focuses on the Ziehl-Neelsen's [12] method of analyzing AFB. Ziehl-Neelsen's [12] method is a special bacteriological stain used to identify acid-fast organisms, especially Mycobacteria. Tubercle bacilli are the most important part of organisms. This

approach makes it easier to find tubercle bacilli since their lipid rich cell walls make them resistant to Gram stain. Ziehl-Neelsen's [12] method can also be used to stain other bacteria, such as Nocardia. If we apply it to medical clinical science, we do not need to analyze all images. Generally speaking, there are 300 images on a slide. This approach only requires a few images (9 or more tubercle bacilli in an image) to determine whether a patient suffers from tuberculosis or not. It can reduce the manpower and time to correctly diagnose a patient with tuberculosis. Foreroa [11] applied a technique of counting color steps to fluorescence micrograph. Otsu's [13] method which selects a threshold automatically from a gray level histogram was derived from discriminant analysis. Otsu's [13] method directly deals with evaluating the goodness of thresholds, and the optimal threshold (or a set of thresholds) is selected by the discriminant criterion. Liu's [14] proposed Shape Model-Based technique for cutting blood cells.

The advantages of the proposed GIEA are summarized as below: 1) The GIEA uses group-based population to evaluate the fuzzy rule locally. 2) The GIEA uses the EICS method to let the better solutions from different groups cooperate to generate better solutions in the next generation. 3) It indeed can obtain better performances and converge faster than other traditional genetic methods.

2. Materials and Methods

2.1 Review A TSK-Type Neuro Fuzzy controllers

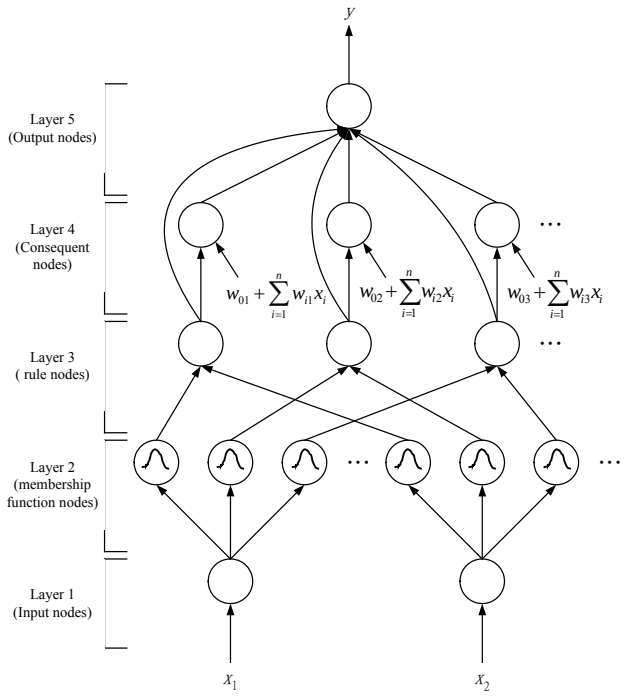
A Takagi-Sugeno-Kang (TSK) type Neuro Fuzzy controllers (TFC) [26] employs different implication and aggregation methods than the standard Mamdani controller. Instead of using fuzzy sets the conclusion part of a rule, is a linear combination of the crisp inputs.

$$\text{IF } x_1 \text{ is } A_{1j}(m_{1j},\sigma_{1j}) \text{ and } x_2 \text{ is } A_{2j}(m_{2j},\sigma_{2j}) \text{ and}\ldots\text{and } x_n \text{ is } A_{nj}(m_{2j},\sigma_{nj})$$
$$\text{THEN } y' = w_{0j} + w_{1j}x_1 + \ldots + w_{nj}x_n \quad (1)$$

where m_{ij}, and σ_{ij} represent a Gaussian membership function with mean and deviation with *i*th dimension and *j*th rule node. w_{0j} represents the first parameter of a linear combination of input variables with *j*th rule node and w_{ij} represents the *i*th parameter of a linear combination of *i*th input variable**.** Since the consequence of a rule is crisp, the defuzzification step becomes obsolete in the TSK inference scheme. Instead, the control output is computed as the weighted average of the crisp rule outputs, which is computationally less expensive then calculating the center of gravity.

The structure of TFC is shown in the fig. 1, where *n* and *M* is, respectively, the number of input dimensions and the number of rules. It is a five-layer network structure. The functions of the nodes in each layer are described as follows:

Fig.1. *The TSK-type neural fuzzy networwk.*



Layer1 (*Input Node*): No function is performed in this layer. The node only transmits input values to layer 2. That is

$$u_i^{(1)} = x_i \quad (2)$$

Layer2 (*Membership Function Node*): Nodes in this layer correspond to one linguistic label of the input variables in layer1; that is, the membership value specifying the degree to which an input value belongs to a fuzzy set is calculated in this layer. For an external input x_i, the following Gaussian membership function is used:

$$u_{ij}^{(2)} = \exp\left(-\frac{\left[u_i^{(1)} - m_{ij}\right]^2}{\sigma_{ij}^2}\right) \quad (3)$$

where m_{ij} and σ_{ij} are, respectively, the center and the width of the Gaussian membership function of the *j*th term of the *i*th input variable x_i.

Layer 3 (*Rule Node*): The output of each node in this layer is determined by the fuzzy AND operation. Here, the product operation is utilized to determine the firing strength of each rule. The function of each rule is

$$u_j^{(3)} = \prod_i u_{ij}^{(2)} \quad (4)$$

Layer 4 (*Consequent Node*): Nodes in this layer are called consequent nodes. The input to a node in layer 4 is the output delivered from layer 3, and the other inputs are the input variables from layer 1 as depicted in the fig. 1. For this kind of node, we have

$$u_j^{(4)} = u_j^{(3)}(w_{0j} + \sum_{i=1}^{n} w_{ij} x_i) \quad (5)$$

where the summation is over all the inputs and where w_{ij} are the corresponding parameters of the consequent part.

Layer 5 (*Output Node*): Each node in this layer corresponds to one output variable. The node integrates all the actions recommended by layers 3 and 4 and acts as a defuzzifier with

$$y = u^{(5)} = \frac{\sum_{j=1}^{M} u_j^{(4)}}{\sum_{j=1}^{M} u_j^{(3)}} = \frac{\sum_{j=1}^{M} u_j^{(3)}(w_{0j} + \sum_{i=1}^{M} w_{ij} x_i)}{\sum_{j=1}^{M_k} u_j^{(3)}} \quad (6)$$

where *M* is the number of fuzzy rule.

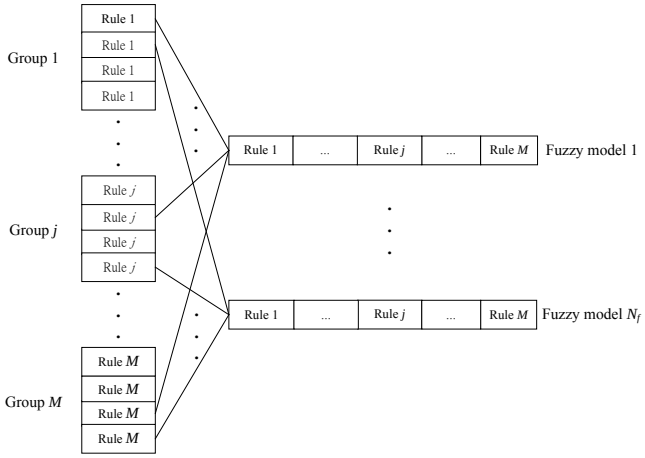
2.2 Group Interaction-based Evolutionary Algorithm

This section will introduce the proposed group interaction-based evolutionary algorithm (GIEA) method. Recently, many researches try to enhance the traditional GAs have been made [15-19]. One category of them tries to modify the structure of a population. Examples in this category include the distributed GA [16], the cellular GA [17], and the symbiotic GA [18].

This study proposes the group interaction-based evolutionary algorithm (GIEA) for improving the symbiotic GA [18]. In the proposed GIEA, the algorithm is developed from symbiotic evolution. The idea of symbiotic evolution was first proposed in an implicit fitness-sharing algorithm that is used in an immune system model [19]. The authors developed artificial antibodies to identify artificial antigens. Because each antibody can match only one antigen, a different population of antibodies is required to effectively defend against a variety of antigens. As shown in [3] and [19], partial solutions can be characterized as specializations. The specialization property ensures diversity, which prevents a population from converging to suboptimal solutions. A single partial solution cannot "take over" a population since there must exists other specializations. Unlike the standard evolutionary approach which always causes a given population to converge, hopefully at the global optimum, but often at a local one, the symbiotic evolution find solutions in different, unconverted populations [3] and [19]. The GIEA is different from the traditional symbiotic evolution; with each population in the GIEA is divided to several groups. Each group represents a set of the chromosomes that belong to a fuzzy rule.

In the proposed GIEA, the structure of the population consists of several groups. Each group represents a set of the chromosomes that belong to a fuzzy rule. The structure of the chromosome in the GIEA is shown in the fig. 2. However, to let groups that can cooperate to generate better solutions, the GIEA proposes the elites-base interaction crossover strategy (EICS) to let the better solutions form different groups can cooperate to generate better solutions in the next generation.

Fig.2. *The structure of the chromosomes in GIEA.*



In the proposed GIEA, the coding structure of the chromosomes must be suitable that each chromosome represent only one fuzzy rule. The fig. 3 describes a fuzzy rule that had the form of Eq. (1) where m_{ij} and σ_{ij} represent a Gaussian membership function with mean and deviation with *i*th dimension and *j*th rule node.

Fig.3. *Coding a rule of a TFC into a chromosome in the GIEA.*

m_{1j}	σ_{1j}	m_{2j}	σ_{2j}	…	m_{nj}	σ_{nj}	w_{0j}	w_{1j}	…	w_{nj}

The learning process of the GIEA in each group involves six major operators: initialization, fitness assignment, elite-based reproduction strategy (ERS), elites-base interaction crossover strategy (EICS), and mutation. The whole learning process is described step-by-step as follows:

a. Initialization step:

Before the GIEA is designed, individuals forming several initial groups should be generated. The initial groups of the GIEA are generated randomly within a fixed range. The following formulations show how to generate the initial chromosomes in each group:

Deviation: $Chr_{g,c}[p] = random[\sigma_{\min}, \sigma_{\max}]$

where $p=2, 4, \ldots, 2n$; $g=1, 2, \ldots, M$; $c=1, 2, \ldots, N_C$; (7)

Mean: $Chr_{g,c}[p] = random[m_{\min}, m_{\max}]$

where $p=1, 3, \ldots, 2n-1$; (8)

Weight: $Chr_{g,c}[p] = random[w_{\min}, w_{\max}]$

where $p=2n+1, 2n+2, \ldots, 2n+(1+n)$, (9)

where $Chr_{g,c}$ represents *c*th chromosome in *g*th group; *M* represents total number of groups and N_C is the total number of chromosomes in each group; *p* represents the *p*th gene in a $Chr_{g,c}$; and $[\sigma_{\min}, \sigma_{\max}]$, $[m_{\min}, m_{\max}]$, and $[w_{\min}, w_{\max}]$ represent the range that are predefined to generate the chromosomes.

b. Fitness assignment step:

As previously state, for the GIEA, the fitness value of a rule (an individual) is calculated by summing up the fitness values of all the possible combinations in the chromosomes that are selected randomly from *M* groups. The details for assigning the fitness value are described step by step as follows:

• **Step 1.** Randomly choose one fuzzy rule from each group such that the size is N_C.

• **Step 2.** Evaluate every TFC that is generated from step1 to obtain a fitness value. The fitness value is defined as follows:

$$Fitness\_Value = 1 \Big/ (1 + E(y, \overline{y})), \quad \text{where} \quad E(y, \overline{y}) = \sum_{i=1}^{N} (y_i - \overline{y}_i) \tag{10}$$

where y_i represents the desired value of the *i*th output, $\overline{y}_i$ represents the predicted value, $E(y, \overline{y})$ is a error function and *N* represents a numbers of the training data of each generation. The average fitness value represents the performance of a rule (individual).

• **Step 3.** Divide the fitness value by *M* and accumulate the divided fitness value to the selected rules with their fitness value records that were set to zero initially.

• **Step 4.** Repeat the above steps until each rule (chromosome) in each group has been selected a sufficient number of times, and record the number of TFC models in which each individual has participated.

• **Step 5.** Divide the accumulated fitness value of each chromosome by the number of times it has been selected. The average fitness value represents the performance of a rule.

c. Elites-based Reproduction Strategy (ERS):

Reproduction is a process in which individual strings are copied according to their fitness value. A fitness value is assigned to each chromosome in each group according to a fitness assignment method in which high numbers denote a good fit. The goal of the GIEA is to maximize the fitness value. For keeping the stable of the algorithm, this study proposes an elite-based reproduction strategy (ERS) to let the best combination of chromosomes in each group can be kept in the next generation. In the GIEA, the chromosome that has best fitness value may not be the chromosome in the best combination. About this, in the ERS, every chromosome in the best combination in each group must be kept by performing reproduction step. In the other chromosomes in each group, this study uses the roulette-wheel selection method [20] – a simulated roulette is spun – for this reproduction process. The best performing chromosomes in the top half of each group [3] advance to the next generation. The other half is generated to perform crossover operations on chromosomes in the top half of the parent generation. In the reproduction step, the top half of the population for each group must be kept the same number of chromosomes.

d. Elites-base Interaction Crossover Strategy (EICS):

Although the ERS operation can search for the best existing individuals, it does not create any new individuals. In nature, an offspring has two parents and inherits genes from both. The main operator working on the parents is the crossover operator, the operation of which occurs for a selected pair with a crossover rate. In this paper, for letting groups that can cooperate to generate better solutions, the elites-base interaction crossover strategy (EICS) is proposed to perform the crossover operation. The EICS mimics the cooperation phenomenon in society, in which individuals become more suited to the environment as they acquire and share more knowledge of their surroundings. In the EICS, the elites of each group will select to perform crossover operation in the next generation. The best performing individuals in the top half of each group that are called elites are used to select the parents for performing the EICS. Details of the EICS are shown below.

• **Step 1.** The first one of the parents that is used to perform the crossover operation is selected from the original group by using the following equations:

$$Fitness\_Ratio_{g,\,t} = \frac{\sum_{u=1}^{t} fitness_{g,\,u}}{\sum_{c=1}^{Nc} fitness_{g,\,c}}, \quad \text{where} \quad t = 1,\ 2,\ \cdots,\ Nc; \tag{11}$$

$$Rand\_Value[g] = Random[0,\ 1], \quad \text{where} \quad g = 1,\ 2,\ \cdots,\ M; \tag{12}$$

$$Parent\_SiteA[g] = t, \quad \text{if} \quad Fitness\_Ratio_{g,\,t-1} < Rand\_Value[g] \le Fitness\_Ratio_{g,\,t}, \tag{13}$$

where $Fitness\_Ratio_{g,\,t}$ is a fitness ratio of the fitness value of *t*th chromosome in the *g*th group; $Rand\_Value[g] \in [0,\ 1]$ is the random values of *g*th group; $Parent\_SiteA[g]$ is the site where the first parent is. According to Eq. (13), if the $Rand\_Value[g]$ is greater than the fitness ratio at (*t*-1)th chromosome in *g*th group and smaller or equal to the fitness ratio at *t*th chromosome in *g*th group, the site of the first parent of *g*th group is assigned to *t*.

•**Step 2.** After determining the first parent, the best performing elites every group is used to determine the other parent. In this step, the total fitness ratio of every group is computed according to the following equations:

$$Total\_Fitness_g = \sum_{c=1}^{Nc} fitness_{g,\,c}, \quad \text{where} \quad g = 1,\ 2,\ \cdots,\ M; \tag{14}$$

$$Total\_Fitness\_Ratio_w = \frac{\sum_{u=1}^{w} Total\_Fitness_u}{\sum_{g=1}^{M} Total\_Fitness_g}, \tag{15}$$

where $w = 1,\ 2,\ \cdots,\ M;$

where $Total\_Fitness_g$ represents the summation of the fitness value of every chromosomes in gth group; $Total\_Fitness\_Ratio_w$ is a total fitness ratio of wth group.

•**Step 3.** Determine the group where the chromosome is selected from to be the other parent for performing crossover with the $Parent\_SiteA[g]$ th chromosome in gth group according to the following equations:

$$Group\_Rand\_Value[g] = Random[0,1] \tag{16}$$

where $g = 1,\ 2,\ \cdots,\ M;$

$$Parent\_Group\_SiteB[g] = w, \quad \text{if}$$

$$Total\_Fitness\_Ratio_{w-1} < Group\_Rand\_Value[g] \le Total\_Fitness\_Ratio_w, \tag{17}$$

where $Group\_Rand\_Value[g] \in [0,\ 1]$ is a random values of gth group; $Parent\_Group\_SiteB[g]$ represents the site of the group that the second parent is selected from.

•**Step 4.** After the $Parent\_Group\_SiteB[g]$ th group is selected, the ECCS determines the other present in the selected $Parent\_Group\_SiteB[g]$ th group according to the following equations:

$$Fitness\_Ratio_{Selected\_g,\,t} = \frac{\sum_{u=1}^{t} fitness_{Selected\_g,\,u}}{\sum_{c=1}^{Nc} fitness_{Selected\_g,\,c}},$$

where $t = 1,\ 2,\ \cdots,\ Nc;\ Selected\_g = Parent\_Group\_SiteB[g];$

(18)

$$Rand\_Value[g] = Random[0,\ 1], \quad \text{where} \quad g = 1,\ 2,\ \cdots,\ M; \tag{19}$$

$$Parent\_SiteB[g] = l, \quad \text{if}$$

$$Fitness\_Ratio_{Selected\_g,\,l-1} < Rand\_Value[g] \le Fitness\_Ratio_{Selected\_g,\,l}, \tag{20}$$

where $Fitness\_Ratio_{Selected\_g,\,t}$ is a fitness ratio of the fitness value of tth chromosome in the $Parent\_Group\_SiteB[g]$ th group; and $Parent\_SiteB[g]$ is the site where the second parent is.

After the EICS selects the presents form the gth group and $Parent\_Group\_SiteB[g]$ th group, the individuals ($Parent\_SiteA[g]$ th chromosome and the $Parent\_SiteB[g]$ th chromosome) are crossed and separated using a two-point crossover in the gth group. The two-point crossover exchanges the site's values between the selected sites of parents' individual create new individuals. After this operation, the individuals with poor performances are replaced by the newly produced offspring.

e. Mutation:

Although ERS and EICS methods would produce many new strings, they do not introduce any new information to the population at the site of an individual. Mutation is an operator that randomly alters the allele of a gene. Mutation can randomly alter the allele of a gene. In this paper, a uniform mutation [20] is adopted, and the mutated gene is drawn randomly from the domain of the corresponding variable.

The aforementioned steps are done repeatedly and stopped when the predetermined condition is achieved.

2.3 The Tubercle Bacilli Diagnosis System (TBDS)

The tubercle bacilli diagnosis system (TBDS) is introduced in this section. The proposed method can be divided into the following three steps. First, segment the image and collect the characteristics of the segmented images. Second, segment the images, categorize them basing on the length, roundness, and area of the tubercle bacilli. Third, classify them by TFC-GIEA which will be used to train them, and then perform that training of TFC-GIEA according to the form and characteristics of the tubercle bacilli.

The experiments in this study used a microscope eyepiece with 10x magnification and an objective magnification of 100x. A digital camera or Charge Coupled Device (CCD) was placed above the microscope eyepiece to take digital images of tubercle bacilli for analysis. The digital camera used in this study had a horizontal and vertical resolution of 150 dots per inch (dpi). Each image is 1388 pixels wide and 1040 pixels tall, with 24-bit color imaging.

To obtain the characteristic value of the tubercle bacilli, it is necessary to segment the image. The image after segmentation is a binary image. The purpose of segmentation is to separate the tubercle bacilli from background impurities. This study uses the Otsu [18] automatic best value to search algorithm and color information to carry out segmentation. Otsu utilized statistical analysis to identify the method that yields the within-class minimum and the critical value of the between-class maximum of each group. This approach enables the image of a tubercle bacilli cell to be separated from the background [22-23].

Although the Otsu algorithms probably still find tubercle bacillus, it may require an excessive amount of searching time if the background is complicated. To address this problem, this study uses a hue, intensity and saturation (HIS) in color space to dye the tubercle bacillus and perform color analysis. Hue can be defined as below：

$$H=\begin{cases}\theta & \text{if } B\le G\\ 360-\theta & \text{if } B>G\end{cases}, \tag{21}$$

where $\theta=\cos^{-1}\left\{\dfrac{\frac{1}{2}\left[(R-G)+(R-B)\right]}{\left[(R-G)^2+(R-B)(G-B)\right]^{1/2}}\right\}$,

Experiment results show that a difference between the *R* value and the *G* value. The difference increases if there is tubercle bacillus in the image. The *H* value also increases meanwhile, but the *B* value does not change.

The proposed image segmentation method initially applies the Otsu algorithm to find the probable tubercle bacilli. The next step is to utilize these tubercle bacilli to determine the difference between the *R* value and the *G* value, and then use *H* value to calculate quantitative value (see equation 22). This process accurately identifies tubercle bacilli.

$$\begin{aligned}&R-G>10,\\ &H>260,\\ &B\cong 200,\end{aligned} \tag{22}$$

Even after roughly highlighting the tubercle bacilli, fracturing of the highlighted tubercle bacilli or discontinuity may occur because the dye on the slide may be stained or faded. Therefore, this study uses a morphology closing filter to repair fractured parts of a cell.

The most realistic method of differentiating between true tubercle bacillus and excessively dyeing tubercle bacillus is to extract the eigenvalue of true tubercle bacillus. First, according to the examination results, the tubercle bacillus has a light red, bar shaped appearance, and is aligned in various directions. Then, digitize these tubercle bacilli features and identify tubercle bacillus through the computer.

This method focuses on extracting and calculating the forms and features of tubercle bacillus, including length, roundness, Hu's invariant moment, area, and perimeter. These parameters are regarded as training type neural parameters, as follows:

1. Area (A): the total number of pixels in the target area.
2. Perimeter (P): the total number of pixels from any point to the tubercle bacilli along an edge in the target area.
3. Roundness (R): this value ranges between 1 and 0. The rounder the body is, the closer the roundness is to 1.

$$R=\frac{4\pi A}{(P)^2} \tag{23}$$

4. Length: the longest route after the target area gets thinned.
5. All length: the total length of all routes after the target area gets thinned.
6. Invariant moment (hu_i) [23]: the tubercle bacilli in this study always have the same form, but with different angles and sizes. Thus, the invariant moment is included as extracted feature to utilize the fact that the invariant moment does not change with the variation, translation, rotation, and size of an image.

We can deduce seven invariant moments as follows:

$$\begin{aligned}
&\phi_1=\eta_{20}+\eta_{02}\\
&\phi_2=(\eta_{20}-\eta_{02})^2+4\eta_{11}^2\\
&\phi_3=(\eta_{30}-3\eta_{12})^2+(3\eta_{21}-\eta_{03})^2\\
&\phi_4=(\eta_{30}+\eta_{12})^2+(\eta_{21}+\eta_{03})^2\\
&\phi_5=(\eta_{30}-3\eta_{12})(\eta_{30}+\eta_{12})[(\eta_{30}+\eta_{12})^2-3(\eta_{21}+\eta_{03})^2]+(3\eta_{21}-\eta_{03})(\eta_{21}+\eta_{03})[3(\eta_{30}+\eta_{12})-(\eta_{21}+\eta_{03})^2]\\
&\phi_6=(\eta_{20}-\eta_{02})[(\eta_{30}+\eta_{12})^2-(\eta_{21}+\eta_{03})^2]+4\eta_{11}(\eta_{30}+\eta_{12})(\eta_{21}+\eta_{03})\\
&\phi_7=(3\eta_{21}-\eta_{03})(\eta_{30}+\eta_{12})[(\eta_{30}+\eta_{12})^2-3(\eta_{21}+\eta_{03})^2]+(3\eta_{21}-\eta_{03})(\eta_{21}+\eta_{03})[3(\eta_{30}+\eta_{12})-(\eta_{21}+\eta_{03})^2]
\end{aligned} \tag{24}$$

The invariant moment is not affected by translation, rotation and size. Equation 24 shows that its output values will be very large. Because the logarithm function is an increasing function, this study uses the following equations instead of equation 24.

$$hu_i=\left|\log\left(\left|\phi_i\right|\right)\right|, i=1,2,\ldots,7 \tag{25}$$

In all these equations, ϕ_i is the value of the invariant moment after calculation.

The directions and sizes of most of tubercle bacilli are similar, but only the directions of single tubercle bacilli are different. Thus, we can utilize the fact that the invariant moment does not change along with variations in angle and size to identify true tubercle bacilli. There is a fixed area within each tubercle bacilli. If the area of the eigenvalue exceeds the maximal area range, the object is overlapping tubercle bacilli. However, an excessively dyed area only can be the same as the area of single tubercle bacilli, so this study does not regard excessive dyeing as overlapping tubercle bacilli.

We use three eigenvalue to conduct training with TFC-GIEA：length, all length, and invariant moment's first equation $\left|\log(|\phi_1|)\right|$ respectively.

3. Results

Simulation is discussed in this section. The example was run to evaluate the tubercle bacilli diagnosis system (TBDS). For the simulation, the initial parameters are given in Table 1. The initial parameters are determined by practical experimentation or trial-and-error tests [24].

Table 1. *The initial parameters before training.*

Parameters	Value	Parameters	Value
Group Size	10	*Time_Value*	10100
Crossover Rate	0.5	*Desired_Times*	10000
Mutation Rate	0.3	$[\sigma_{\min},\ \sigma_{\max}]$	[0, 2]
$[m_{\min},\ m_{\max}]$	[0, 2]	$[w_{\min},\ w_{\max}]$	[-20, 20]

Example. Evaluating the TBDS

In this experiment, a Pentium III chip with a 800MHz CPU, a 512MB memory, and the visual C++ 6.0 simulation software are applied to this experiment. In this example, the simulation of GIEA-TBDS is discussed. There are 50 samples used to train the TFC-GIEA and 200 samples used to test TFC-GIEA. Based on the standard for clinically diagnosing the severity of tubercle bacilli, the complications can be classified in three grades. Grade I is normal cases and slight cases complicated with gastritis. Grade II is acid reflux complicated with reflux esophagitis A and B which are included at the same level. Grade III is the more severe classification, which includes acid reflux complicated with reflux esophagitis C; duodenal ulcer; gastric ulcer; acid reflux complicated with reflux esophagitis and duodenal ulcer; acid reflux complicated with reflux esophagitis and gastric ulcer; and duodenal and gastric ulcer.

The values are floating-point numbers assigned using the GIEA initially. The fitness function in this example is defined in equation 4 to train TFC. There are ten fuzzy rules used to construct TFC. The evolution learning processed for 500 generations and is repeated 50 times. For comparative analysis, this paper uses the accuracy of three grades to evaluate the performance of the TBDS. After 50 runs, the final training and testing average accuracy of three grades approximates 93% and 92%.

4. Discussion

In order to demonstrate the effectiveness and efficiency of the proposed TFC-GIEA, the SE, GA, and ESP are applied to the same problem in this example. There are ten rules to construct the TFC. The parameters set in three methods are as follows: 1) the numbers of fuzzy rules are all set for 6; 2) the population sizes of SE and GA are 100 and 50 respectively; 4) the population size of the SE and ESP are both set for 50; 3) the crossover rates of SE, ESP, and GA are 0.57, 0.36, and 0.61, respectively; 3) the mutation rate of SE, ESP, and GA are 0.09, 0.15, and 0.14,respectively. The evolution learning processes for 500 generations and is repeated 50 times.

After 50 runs, the final training average accuracy of the SE, ESP, and GA approximate75%, 76%, and 70% and the final testing average accuracy of the SE, ESP, and GA approximate 73%, 70%, and 68%.

The comparison about the training accuracy, testing accuracy and CPU times of proposed method and other methods are shown in table2.

5. Conclusion

This paper proposed a TSK-type Neuro Fuzzy controllers (TFC) with group interaction-based evolutionary algorithm for constructing tubercle bacilli diagnosis system (GIEA-TBDS). The proposed GIEA-TBDS can be divided into two parts. The first part is the learning algorithm and the group interaction-based evolutionary algorithm (GIEA) is proposed. The GIEA can evaluate the fuzzy rule locally and interact with each group to produce the better chromosomes by elites-base interaction crossover strategy (EICS). The second part is the diagnosis system and the TBDS trained by GIEA is proposed. The tubercle bacilli can be diagnosed by TBDS automatically. The summarization of the advantages of the proposed GIEA-TBDS are as follows: 1) the GIEA-TBDS evaluates the fuzzy rule locally with group-based population; 2) the GIEA-TBDS uses the EICS to make the better solutions form different groups and interact each other to generate better solutions in the next generation; 3) the GIEA-TBDS can detect the tubercle bacilli automatically. Computer simulations have been proved that the proposed method is provided with a better performance than the other methods.

Table 2. *Comparison of performance for different methods.*

Method	Training Accuracy			Testing Accuracy			CPU Time (Seconds)
	Best	Worst	Mean	Best	Worst	Mean	
GIEA	**95%**	**90%**	**93%**	**94%**	**89%**	**92%**	**95.58**
HELA	88%	84%	86%	84%	82%	83%	130.35
GSE	80%	75%	78%	80%	70%	75%	141.62
ESP	78%	74%	76%	75%	68%	73%	155.74
SE	77%	73%	75%	72%	68%	70%	165.13
GA	72%	68%	70%	73%	63%	68%	148.82

6. Acknowledgements

This work was supported partially by the National Science Council under grant NSC 99-2221-E-009-148.

7. References

1. Karr CL (1991) Design of an adaptive fuzzy logic controller using a genetic algorithm. *Proc. The Fourth Int. Conf. Genetic Algorithms.* 450–457.
2. Lee M and Takagi H (1993) Integrating design stages of fuzzy systems using genetic algorithms. *Proc. 2nd IEEE Int. Conf. Fuzzy Systems*. 612-617.
3. Lin CT and Jou CP (2000) GA-based fuzzy reinforcement learning for control of a magnetic bearing system. *IEEE Trans. Syst., Man, Cybern., Part B*. 30(2), 276-289.
4. Juang CF, Lin JY and Lin CT (2000) Genetic reinforcement learning through symbiotic evolution for fuzzy controller design. *IEEE Trans. Syst., Man, Cybern., Part B.* 30(2), 290-302.
5. Tang KS (1996) *Genetic algorithms in modeling and optimization*. Ph.D. Dissertation. Dep. Electron. Eng., City Univ. Hong Kong.
6. Lin CJ and Hsu YC (2007) Reinforcement Hybrid Evolutionary Learning for Recurrent wavelet-based

neuro-fuzzy controllers. *IEEE Trans. Fuzzy Systs.* 15(4), 729-745.

7. Levente FA, Razvan A and Catharine JC (2008) A.W. Sarah and S. Nicholas, A genetic algorithm optimized fuzzy neural network analysis of the affinity of inhibitors for HIV-1 protease. *Bioorganic & Medicinal Chemistry*. 16(6), 2903-2911.
8. Benamrane N, Aribi A and Kraoula L (2006) Fuzzy Neural Networks and Genetic Algorithms for Medical Images Interpretation. *IEEE Proc. of the conf. on Geometric Modeling and Imaging: New Trends*.
9. Wang S, Fu D, Xu M and Hu D (2007) Advanced fuzzy cellular neural network: Application to CT liver images. *Artificial Intelligence in Medicine.* 39(1), 65-77.
10. Jafar T, Afshar SJ and Muhammad AD (2012) A new method for position control of a 2-DOF robot arm using neuro–fuzzy controller. *Indian J.Sci.Technol.* 5(3), 2253-2257.
11. Forero M, Sroubek F and Cristobal G (2004) Identification of tuberculosis bacteria based on shape and color. *Real-Time Imaging,* 10(4), 251-262.
12. Ziehl-Neelsen stain, http://en.wikipedia.org/wiki/Ziehl-Neelsen_stain.
13. N. Otsu, A threshold selection method from gray-level histogram, *IEEE Trans. on Systems, Man and Cybernetics*, **9**(1): 62-66, 1979.
14. Liu L and Sclaroff S (2001) Medical image segmentation and retrieval via deformable models,.*Proc. IEEE Int. Conf. on Image Processing*. 3, 1071-1074.
15. Takagi T and Sugeno M (1985) Fuzzy identification of systems and its applications to modeling and control. *IEEE Trans. Systems Man Cybern.* 15, 116-132.
16. Michalewicz Z (1999) *Genetic Algorithms+Data Structures=Evolution Programs*. New York: Springer-Verlag.
17. Tanese R (1989) *Distributed genetic algorithm*. Proc. Int. Conf. Genetic Algorithms. 434–439.
18. Arabas J, Michalewicz Z and Mulawka J (1994) GAVaPS—A genetic algorithm with varying population size. *Proc. IEEE Int. Conf. Evolutionary Computation, Orlando.* 73–78.
19. Moriarty DE and Miikkulainen R (1996) Efficient reinforcement learning through symbiotic evolution. *Mach. Learn.* 22, 11–32.
20. Smith SE, Forrest S and Perelson AS (1993) Searching for diverse, cooperative populations with genetic algorithms, *Evol. Comput.* 1(2), 127–149.
21. Cordon O, Herrera F, Hoffmann F and Magdalena L (2001) Genetic fuzzy systems evolutionary tuning and learning of fuzzy knowledge bases. *Advances in Fuzzy Systems-Applications and Theory*, NJ: World Scientific Publishing. 19.
22. Weaver JR and Au JL (1997) Application of automatic thresholding in image analysis scoring of cells in human solid tumors labeled for proliferation markers, *Cytometry A*. 29, 128-135.
23. Wu K, Gauthier D and Levine MD (1995) Live Cell Image Segmentation. *IEEE Trans. on Biomedical Engineering*. 42, 1-12.
24. Grefenstette JJ (1986) Optimization of control parameters for genetic algorithms. *IEEE Trans. Syst. Man Cybern.* 6(1) 122–128.
25. Gomez FJ (2003) *Robust non-linear control through neuroevolution.* Ph.D. Disseration, The University of Texas at Austin.
26. Lin CJ and Xu YJ (2006) A Self-Adaptive Neural Fuzzy Network with Group-Based Symbiotic Evolution and Its Prediction Applications, *Fuzzy Sets and Systems*. 157, 1036-1056.