Automatic Counting Cancer Cell Colonies using GIEA for TSK-type Neural Fuzzy Network

Hsien-Tse Chen
Dept. of Electrical Engineering
National Chiao Tung University
Hsinchu, Taiwan (R.O.C)
Hsientse.ece93g@nctu.edu.tw

Sheng-Fuu Lin
Dept. of Electrical Engineering
National Chiao Tung University
Hsinchu, Taiwan (R.O.C)
sflin@mail.nctu.edu.tw

Abstract*—This paper proposes a TSK-type neural fuzzy network (TNFN) with a group interaction-based evolutionary algorithm (GIEA) for constructing the cancer cell colonies diagnosis system (CCCDS). The proposed GIEA is designed on the basis of 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 belongs 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 CCCDS, the EICS is used to train the CCCDS. After trained by the EICS, the CCCDS can diagnose the visible cancer cell colonies automatically. The performance of the GIEA is proved to be better than other existing models in diagnosing cancer cell colonies.

Keywords: TSK-type Neural fuzzy network, Cancer Cell Colonies, group interaction-based evolutionary algorithm.

. INTRODUCTION

The concept of the fuzzy logic or artificial neural networks for control problems has become a popular research area [1]-[3]. Traditional control theory usually requires a mathematical model for designing controllers. The inaccuracy of mathematical models at plants usually reduces the performance of the controllers, especially for nonlinear and complex control problems [4]-[6]. Fuzzy logic can express the ambiguity of human thinking and translate expert knowledge into computable numerical data.

In the design of a fuzzy controller [7]-[9], adjusting the required parameters is important. Therefore, back-propagation (BP) training was widely used in [1], [2]. It is an effective training technique and applied to networks with a forward structure. Since the steepest descent technique used in BP training can minimize the error function, the algorithms may reach the local minima fast and never find the global solution. To solve these problems, several evolutionary algorithms, such as genetic algorithm (GA) [10], genetic programming [11], evolutionary programming [12], and evolution strategies [13], have been proposed. They are similar and global search techniques. They simultaneously evaluate many points in the search space so they probably converge toward the global solution. Therefore, an evolutionary method using for training the fuzzy model has changed into an important field.

The evolutionary fuzzy model generates a fuzzy system automatically throuth incorporating evolutionary learning procedures [14]-[19], where the well-known procedure is the genetic algorithms (GAs). In [14], 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 [15] on the basis of. Lin and Jou [16] proposed GA-based fuzzy reinforcement learning to control magnetic bearing systems. In [17], Juang et al. proposed genetic reinforcement learning in the design of fuzzy controllers. The GA adopted in [17] was based upon traditional symbiotic evolution which, when applied to fuzzy controller design, complemented the local mapping property of a fuzzy rule. In [18], Tang proposed a hierarchical genetic algorithm. The hierarchical genetic algorithm enables the optimization of the fuzzy system design for a particular application. In [19], 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, neural fuzzy network have been applied to diagnosis system [20]-[23] in several researches. Genetic algorithm optimized fuzzy neural network (GA-FNN) proposed by Levente et al. [20] is capable of greatly assisting medicinal chemists in the design of lead compounds for HIV-1 protease and other therapeutically important enzymes. A fuzzy neural network (FNN) proposed by Quteishat et al. [21] was trained on a dataset of 177 HIV-1 protease ligands with experimentally measured IC50 values. The advanced fuzzy cellular neural network (AFCNN), Wang et al. [22] proposed is a variant of the fuzzy cellular neural network (FCNN) and is proposed to effectively segment CT liver images. The improved fuzzy cellular neural network (IFCNN) proposed by Wang et al. [23] has the global stability and uses the experimental results for microscopic white blood cell to demonstrate its obvious advantage over FCNN in keeping the boundary integrity.

In this paper, as same with [20]-[23], we also applied neural fuzzy network to diagnosis the system. Therefore, TSK-type neural fuzzy network (TNFN) with a group interaction-based evolutionary algorithm (GIEA) is proposed for constructing the cancer cell colonies diagnosis system (CCCDS). In CCCDS, Medical treatments for liver cancer can be divided into three general categories: chemotherapy, hyperthermia and radiotherapy. The Clonogenic Assay [24-27] is the current standard in vitro for hyperthermia and radiotherapy because its results correlate with biopsy results. The results of the MTT [24] assay showed a high degree of correlation with HTCA results in predicting the sensitivity of cancer cell lines to platinum analogues and anthracyclines/anthracenedione. N. Maximilian et al. made densitometric [25] software available with a detailed description of how to use and install the necessary features; J. Dahle et al. [26] employed a flat bed scanner to image 12 60-mm petri dishes at a time. Two major problems in automated colony counting are the clustering of colonies and edge effects; the feasibility, evaluation [27], and predictive value of the colony-forming assay with human tumor xenografts for screening anticancer drugs have been studied.

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.

. REVIEW A TSK-TYPE NEURAL FUZZY NETWORK

A Takagi-Sugeno-Kang (TSK) type neural fuzzy network (TNFN) [28] 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:

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

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.

A five-layer network structure of TFC is shown in the figure 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:

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

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

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:

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

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

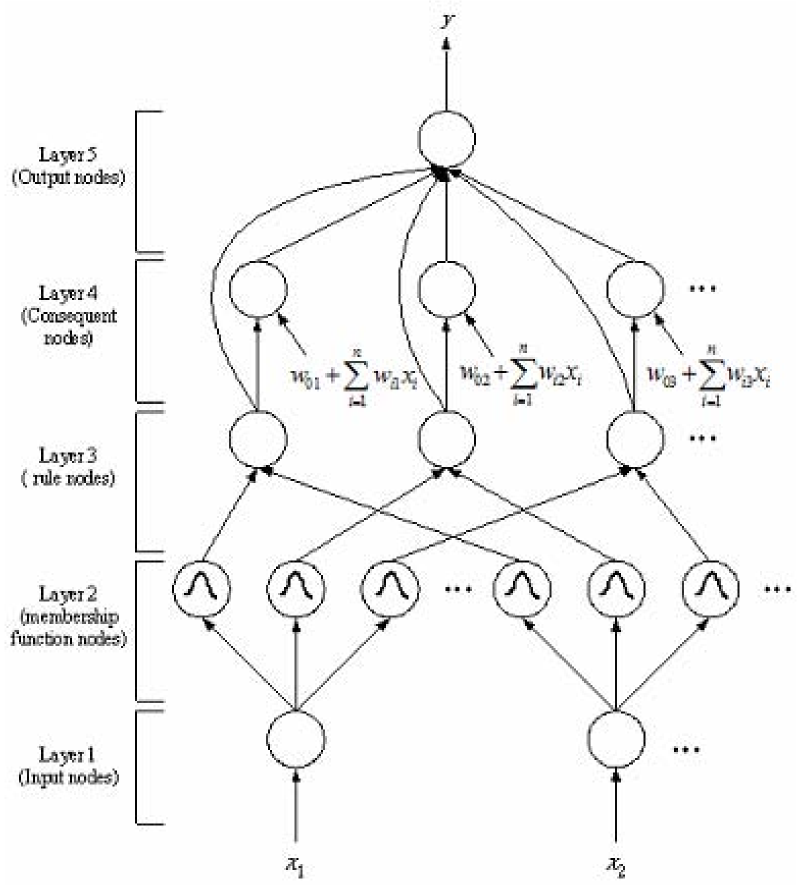

Figure 1. The TSK-type neural fuzzy networwk.

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

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

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 figure 1. For this kind of node, we have

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

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

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

where *M* is the number of fuzzy rule.

. GROUP INTERACTION-BASED EVOLUTIONARY ALGORITHM (GIEA)

This section will introduce the proposed group interaction-based evolutionary algorithm (GIEA) method. Recently, there are many research works try to improve the traditional GAs performance [29]-[32]. One category of them tries to modify the structure of a population. Examples in this category include the distributed GA [30], the cellular GA [31], and the symbiotic GA [32].

At research works [32] proposes the group interaction-based evolutionary algorithm (GIEA) for improving the symbiotic GA. The GIEA algorithm is developed from a symbiotic evolution. The idea of symbiotic evolution was first proposed in an implicit fitness-sharing algorithm that is used in an immune system model [33]. 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 [17] and [32], 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 [17] and [32]. 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.

There are several groups in the GIEA’s structure of the population. 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 figure 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.

The chromosomes of the GIEA are representing one fuzzy rule. A fuzzy rule that had the form of equation (1) is described in the figure 3 where m_{ij} and σ_{ij} represent a Gaussian membership function with a mean and a deviation with *i*-th dimension and *j*-th rule node.

There are six major steps in the learning process of the GIEA and described as follows: (a) initialization, (b) fitness assignment, (c) elite-based reproduction strategy (ERS), (d) elites-base interaction crossover strategy (EICS), and (f) mutation. The whole learning process is described step-by-step as follows:

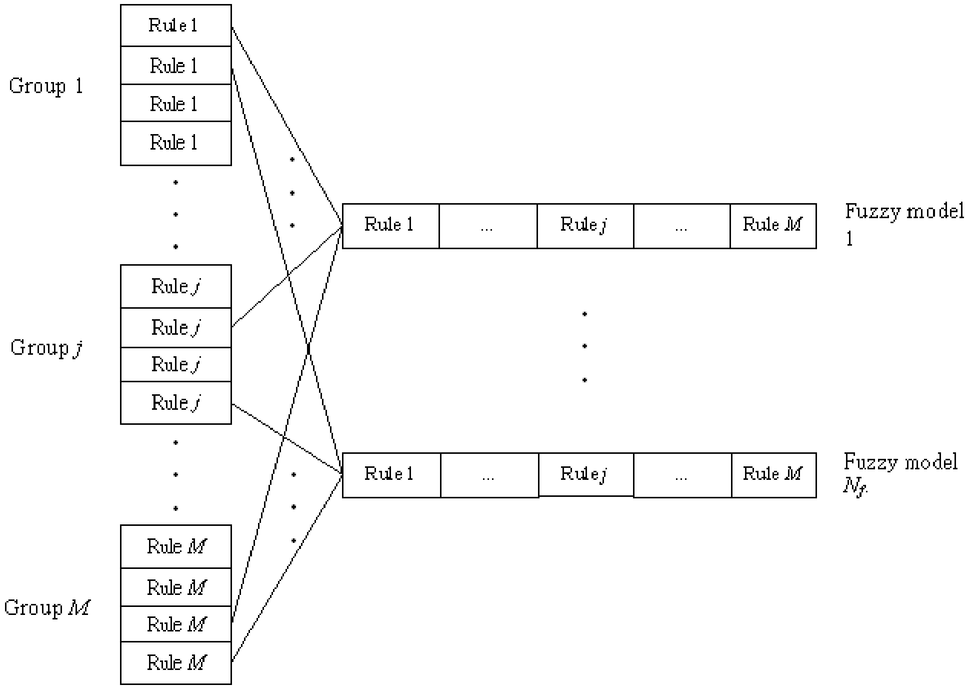

Figure 2. The structure of the chromosomes in GIEA.

m_{1j}	σ_{1j}	m_{2j}	σ_{2j}	$\cdots$	m_{nj}	σ_{nj}	w_{0j}	w_{1j}	$\cdots$	w_{nj}

Figure 3. Coding a rule of a TNFN into a chromosome in the GIEA.

a. Initialization step:

To start designing the GIEA, the several initial groups of individuals forming should be generated. Random values within a fixed range are generated for the initial groups of the GIEA. The equations which generate the initial chromosomes in each group are shown as follows:

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

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

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

where p=1, 3, …, 2n-1; (8)

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

where p=2n+1, 2n+2, …, 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 fitness value assigning steps are described as follows:

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

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

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

where y_i represents the desired value of the *i*th output, $\bar{y}_i$ represents the predicted value, $E(y,\bar{y})$ 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.** To divide the fitness value by *M* and accumulate the divided fitness value to the selected rules which 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 TNFN 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):

In this reproduction process, the individuals are copied according to their fitness values. A high fitness value denotes 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 to 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 [34] – a simulated roulette is spun – for this reproduction process. The best performing chromosomes in the top half of each group [17] advance to the next generation. The other half will be generated by crossover operations with the top half of the parent generation. In this 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):

Form above step, we know the ERS operation can keep the best individuals but it does not create any new individuals. In nature, an offspring has two parents and inherits genes from both. A crossover operator is the main process with the parents' chromosome. The probability of the crossover operation is according to 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.** To Choice the first one of the parents from the original group by using the following equations:

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

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

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

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 equation (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.** If the first parent is chosen, then the best performing elites in each group are used to determine the other parent. In this step, we use the following equation to calculate the all fitness ratio of every group:

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

$$Total\_Fitness\_Ratio_w = \frac{\sum_{u=1}^{w} Total\_Fitness_u}{\sum_{g=1}^{M} Total\_Fitness_g}, \quad \text{where} \quad w = 1,\ 2,\ \cdots,\ M; \tag{15}$$

where $Total\_Fitness_g$ represents the summation of the fitness value of every chromosomes in *g*-th group; $Total\_Fitness\_Ratio_w$ is a total fitness ratio of *w*-th 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 *g*-th group according to the following equations:

$$Group\_Rand\_Value[g] = Random[0,1] \quad \text{where} \quad g = 1,\ 2,\ \cdots,\ M; \tag{16}$$

$$\begin{gathered} Parent\_Group\_SiteB[g]=w, \quad \text{if} \\ Total\_Fitness\_Ratio_{w-1} < Group\_Rand\_Value[g] \le Total\_Fitness\_Ratio_{w}, \end{gathered} \tag{17}$$

where $Group\_Rand\_Value[g] \in [0,\ 1]$ is a random values of *g*-th 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:

$$Fitness\_Ratio_{Selected\_g,\ t} = \frac{\sum_{u=1}^{t} fitness_{Selected\_g,\ u}}{\sum_{c=1}^{Nc} fitness_{Selected\_g,\ c}}, \tag{18}$$

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

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

$$\begin{gathered} Parent\_SiteB[g]=l, \quad \text{if} \\ Fitness\_Ratio_{Selected\_g,\ l-1} < Rand\_Value[g] \le Fitness\_Ratio_{Selected\_g,\ l}, \end{gathered} \tag{20}$$

where $Fitness\_Ratio_{Selected\_g,\ t}$ is a fitness ratio of the fitness value of *t*-th 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 *g*-th 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 *g*-th group. The two-point crossover exchanges the site's values between the selected sites of parents' individual create new individuals. A new offspring will replace the individuals with poor performances.

e. Mutation:

Although ERS and EICS methods would produce many new strings, they do not introduce any new information to the population. An individual Mutation is an operator for altering a randomly allele of a gene. In this paper, a uniform mutation [34] 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.

. The cancer cell colonies diagnosis system (CCCDS)

In this section the cancer cell colonies diagnosis system (CCCDS) is introduced. In this paper, a scanner [26] is used to capture the culture dish image. As the culture dish was made from transparent acrylic, an even light source was required to avoid reflection and diffraction, precluding the use of a digital camera for image capture. Another advantage of using a scanner is its fixed focal distance. It will make the user unnecessary to spend time adjusting the focus during image capture. A cancer cell colony can only appear in six circular wells within one culture dish image. A image processing method will recognize the six circular wells.

A binarization process is used to separate a cancer cell colony from the background inside the well region.

This paper uses an image differencing method in which the image of an empty culture dish is processed to provide a background image *Ib* without any cancer cell colonies. The image *I* of a culture dish with potential cancer cell colonies was then used as the input image and compared against the background image to form the foreground mask image *Im*. *Im* is defined as a binary image, and can be derived using equations (21) and (22) below , where τ *f* is a suitable threshold value.

$$Im(x,y)=\begin{cases}1, & \text{if } \|I(x,y)-Ib(x,y)\| > \tau_f, \\ 0, & \text{otherwise,}\end{cases} \tag{21}$$

$$\|I(x,y)-Ib(x,y)\|=\sqrt{(I_r(x,y)-Ib_r(x,y))^2+(I_g(x,y)-Ib_g(x,y))^2+(I_b(x,y)-Ib_b(x,y))^2}, \tag{22}$$

Once the norm of the input image *I* and the background exceeds a certain threshold value at (*x*,*y*), that point will be the foreground point and equation (23) will tag it as 1 The definition of the *Ip* that can be acquired from *Im* is given below:

$$Ip(x,y)=\begin{cases}I(x,y), & \text{if } \ Im(x,y)=1, \\ 0, & \text{otherwise,}\end{cases} \tag{23}$$

In other words, *Ip* consists only of image pixels defined as 1 in *Im*. All other image pixels, where *Im* is 0, are considered to be a part of the background.

For the cancer cell colony identification system, to determine whether an object is a cancer cell colony or not, it must base on that object's features. Therefore, the system which proposed in this paper uses color information to identify cancer cell colonies instead of shape for identification. This color information includes the relationship between RGB values, the hue (*H*), and whether there are more than 50 cells in a colony or not. Surface area is another important feature. The expression for *H* can be derived from equation (24):

$$H(x,y)=\begin{cases}\theta(x,y), & \text{if} \quad B(x,y)<G(x,y), \\ 360-\theta(x,y), & \text{if} \quad B(x,y)\geq G(x,y),\end{cases} \tag{24}$$

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

Manual empirical assessment shows that stained cancer cell colonies tend to be purple tinted. Equation (25) can be used to remove noise in an effective manner. In this normal cancer cell colony, the RGB difference value was 152.

$$R_{rgb}(area)=G_{mean}(area)-\left|R_{mean}(area)-B_{mean}(area)\right|, \tag{25}$$

The proposed system uses a high-powered Charge Coupled Device (CCD) to observe 100 cancer cell colonies at 200X magnification. The individual cells in each colony are then mapped to their corresponding image pixels in the culture dish image. A clinical experience indicates there are 50 cells approximate to 25 pixels in a culture dish image.

. SIMULATION

Simulation is discussed in this section. The example was run to evaluate the cancer cell colonies diagnosis system (CCCDS). For the simulation, the initial parameters are given in Table 1. The initial parameters are determined by practical experimentation or trial-and-error tests [35].

TABLE 1 THE INITIAL PARAMETERS BEFORE TRAINING

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

Example. Evaluating the CCCDS

In this example, the simulation of GIEA-CCCDS is demonstrated. A total of 25 culture dish specimens were prepared. Each dish contained six 35mm-sized wells, yielding a total of 150 single cell images to process. The resolution of image is 500 dots per inch (dpi), the width and height of each image is 2262 pixels and is 1492 pixels for color image. 50 samples are used to train the TNFN-GIEA and 150 samples are used to test TNFN-GIEA. These were divided according to their incubation conditions into37℃, 41℃and42.5℃. Treatment times were 0.5h, 1h, 2h and 4h.

The values are floating-point numbers assigned with the GIEA initially. The fitness function in this example is defined in equation (10) to train TNFN. Ten fuzzy rules are used to construct TNFN. The evolution learning processed for 500 generations is repeated 50 times. As for comparative analysis, this paper uses the accuracy of three grades to evaluate the performance of the CCCDS. After 50 runs, the final average training and testing accuracy of three grades gets up to 94% and 93%.

In order to demonstrate the effectiveness and efficiency of the proposed TNFN-GIEA, the SE, GA, and ESP are applied to the same problem in this example. There are ten rules to construct the TNFN. The parameters set in three methods are as follows: 1) the numbers of fuzzy rules are all set as 6; 2) the population sizes of SE and GA are 100 and 50; 3) the population size of the SE and ESP are both set as 50; 4) the crossover rates of SE, ESP, and GA are 0.57, 0.36, and 0.61; 5) the mutation rate of SE, ESP, and GA are 0.09, 0.15, and 0.14. The learning curves of the four methods (GIEA, SE, ESP, and GA) are shown in the figure 4. As shown in this figure, the TNFN-GIEA obtains better fitness value than others.

After 50 runs, the final average training accuracy of the SE [17], ESP [36], and GA [14] approximates 75%, 76%, and 70%. The final average testing accuracy of the SE [17], ESP [36], and GA [14] approximates 71%, 72%, and 68%.

The comparison about the training accuracy, testing accuracy and CPU times of proposed method and other methods ([14, 17, 19, 36, and 37]) are shown in table2. In this experiment, a Pentium III chip with a 400MHz CPU, a 512MB memory, and the visual C++ 6.0 simulation software are applied to this experiment. This paper can detect the visible cancer cell colonies diagnosed by the proposed GIEA-CCCDS automatically. Moreover, the performance of the GIEA-CCCDS is proved to be better than other methods.

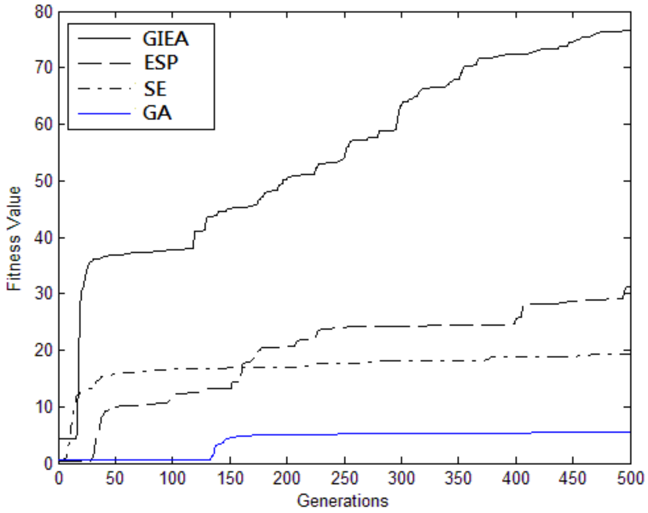

Figure 4. The learning curves of the GIEA, ESP [36], SE [17] and GA [14].

TABLE 2 COMPARISON OF PERFORMANCE FOR DIFFERENT METHODS.

Method	Training Accuracy			Testing Accuracy			CPU Time (Seconds)
	Best	Worst	Mean	Best	Worst	Mean	
GIEA	**95%**	**93%**	**94%**	**95%**	**90%**	**93%**	**94.22**
HELA [19]	87%	83%	85%	85%	82%	84%	131.26
GSE [37]	81%	77%	79%	81%	71%	76%	140.51
ESP [36]	77%	75%	76%	75%	69%	72%	154.65
SE [17]	76%	74%	75%	73%	69%	71%	163.96
GA [14]	73%	67%	70%	71%	65%	68%	147.38

. CONCLUSION

This paper proposed a TSK-type neural fuzzy network (TNFN) with group interaction-based evolutionary algorithm for constructing cancer cell colonies diagnosis system (GIEA-CCCDS). The proposed GIEA-CCCDS 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 CCCDS trained by GIEA is proposed. The cancer cell colonies can be diagnosed by CCCDS automatically. The summarization of the advantages of the proposed GIEA-CCCDS are as follows: 1) the GIEA-CCCDS evaluates the fuzzy rule locally with group-based population; 2) the GIEA-CCCDS 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-CCCDS can detect the cancer cell colonies automatically. Computer simulations have been proved that the proposed method is provided with a better performance than the other methods.

ACKNOWLEDGEMENTS

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

REFERENCES

[1] G. G. Towell, and J. W. Shavlik, “Extracting refined rules from knowledge-based neural networks.” Machine Learning, vol. 13, 1993, pp. 71-101.

[2] C. J. Lin, and C. T. Lin, “An ART-based fuzzy adaptive learning control network,” IEEE Trans. Fuzzy systs., vol. 5, no. 4, 1997, pp. 477-496.

[3] C. F. Juang, and C. T. Lin, “An on-line self-constructing neural fuzzy inference network and its applications,” IEEE Trans. Fuzzy Systs., vol. 6, no. 1, 1998, pp. 12-31.

[4] C. J. Lin, and C. C. Chin, “Prediction and identification using wavelet-based recurrent fuzzy neural networks,” IEEE Trans. Syst., Man, Cybern., Part B, vol. 34, no. 5, 2004, pp. 2144-2154.

[5] K. S. Narendra, and K. Parthasarathy, “Identification and control of dynamical systems using neural networks,” IEEE Trans. Neural Networks, vol. 1, no. 1, 1990, pp. 4-27.

[6] G. Barto, R. S. Sutton, and C. W. Anderson, “Neuron like adaptive elements that can solve difficult learning control problem,” IEEE Trans. Syst., Man, Cybern., vol. 13, no. 5, 1983, pp. 834-847.

[7] J. Lin, “A GA-based neural network with supervised and reinforcement learning,” Journal of the Chinese Institute of Electrical Engineering, vol. 9, no. 1, 2000, pp. 11-25.

[8] T. Lin and C. P. Jou, “GA-based fuzzy reinforcement learning for control of a magnetic bearing system,” IEEE Trans. Syst. Man Cybern. Part B, vol. 30, no. 2, 2000, pp. 276-289.

[9] H. R. Berenji, and P. Khedkar, “Learning and tuning fuzzy logic controllers through reinforcements,” IEEE Trans. Neural Networks, vol. 3, no, 5, 1992, pp. 724-740.

[10] D. E. Goldberg, “Genetic Algorithms in Search Optimization and Machine Learning. Reading,” MA: Addison-Wesley, 1989.

[11] J. K. Koza, “Genetic Programming: On the Programming of Computers by Means of Natural Selection,” Cambridge, MA: MIT Press, 1992.

[12] L. J. Fogel, “Evolutionary programming in perspective: The top-down view,” in Computational Intelligence: Imitating Life, 1994.

[13] Rechenberg, “Evolution strategy, in Computational Intelligence: Imitating Life,” J. M. Zurada, R. J. Marks II, and C. Goldberg, Eds. Piscataway, NJ: IEEE Press, 1994.

[14] C. L. Karr, “Design of an adaptive fuzzy logic controller using a genetic algorithm,” Proc. The Fourth Int. Conf. Genetic Algorithms, 1991, pp. 450–457.

[15] M. Lee, and H. Takagi, “Integrating design stages of fuzzy systems using genetic algorithms,” Proc. 2nd IEEE Int. Conf. Fuzzy Systems, 1993, pp. 612-617.

[16] C. T. Lin, and C. P. Jou, “GA-based fuzzy reinforcement learning for control of a magnetic bearing system,” IEEE Trans. Syst., Man, Cybern., Part B, vol. 30, no.2, 2000, pp. 276-289.

[17] C. F. Juang, J .Y Lin, and C. T. Lin, “Genetic reinforcement learning through symbiotic evolution for fuzzy controller design,” IEEE Trans. Syst., Man, Cybern., Part B, vol. 30, no. 2, 2000, pp. 290-302.

[18] S. Tang, “Genetic algorithms in modeling and optimization. Ph.D. Dissertation,” Dep. Electron. Eng., City Univ. Hong Kong, 1996.

[19] C. J. Lin and Y. C. Hsu, “Reinforcement Hybrid Evolutionary Learning for Recurrent wavelet-based neuro-fuzzy controllers,” IEEE Trans. Fuzzy Systs., vol. 15, no. 4, 2007, pp. 729-745.

[20] F. A. Levente, A. Razvan, and J. C. Catharine, “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, vol. 16, no. 6, 2008, pp. 2903-2911.

[21] H. C. Kuo, H. K. Chang, and Y. Z. Wang, “Symbiotic evolution-based design of fuzzy-neural diagnostic system for common acute abdominal pain,” Expert Systems with Applications, vol. 27, no. 3, 2004, pp. 391-401.

[22] S. Wang, D. Fu, M. Xu, and D. Hu, “Advanced fuzzy cellular neural network: Application to CT liver images,” Artificial Intelligence in Medicine, vol. 39, no. 1, 2007, pp. 65-77.

[23] S. Wang, F. L. C. Korris, D. Fu, “Applying the improved fuzzy cellular neural network IFCNN to white blood cell detection,” Neurocomputing, vol. 70, no. 7, 2007, pp. 1348-1359.

[24] Kawada, T. Yonei, H. Ueoka, K. Kiura, M. Tabata, N. Takigawa, M. Harada, M. Tanimoto, “Comparison of Chemosensitivity Tests: Clonogenic Assay versus MTT Assay,” Acta Medica Okayama, vol. 56, 2002, pp. 129-134.

[25] N. Maximilian, N. Ismat, B. Claus, “Counting Colonies of Clonogenic Assay by Using Densitometric Software,” Radiation Oncology, vol. 2, 2007, pp. 2-4.

[26] J. Dahle, M. Kakar, H. B. Steen, O. Kaalhus, “Automated Counting of Mammalian Cell Colonies by Means of a Flat Bed Scanner and

Image Processing," Cytometry Part A, vol. 60, 2004, pp. 182-188.

[27] D. P. Berger, H. Henb, B. R. Winterhalter, H. H. Fiebig, "The Clonogenic Assay with Human Tumor Xenografts: Evaluation, Predictive Value and Application for Drug Screening," Annals of Oncology, vol. 1, 1990, pp. 333-341.

[28] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control," IEEE Trans. Systems Man Cybern., vol. 15, 1985, pp. 116-132.

[29] Z. Michalewicz, "Genetic Algorithms+Data Structures=Evolution Programs," New York: Springer-Verlag, 1999.

[30] R. Tanese, "Distributed genetic algorithm," Proc. Int. Conf. Genetic Algorithms, 1989, pp. 434–439.

[31] J. Arabas, Z. Michalewicz, and J. Mulawka, "GAVaPS—A genetic algorithm with varying population size," Proc. IEEE Int. Conf. Evolutionary Computation, Orlando, 1994, pp. 73–78.

[32] D. E. Moriarty and R. Miikkulainen, "Efficient reinforcement learning through symbiotic evolution. Mach. Learn, vol. 22, 1996, pp. 11–32.

[33] S. E. Smith, S. Forrest, A. S. Perelson, "Searching for diverse, cooperative populations with genetic algorithms," Evol. Comput., vol. 1, no. 2, 1993, pp. 127–149.

[34] O. Cordon, F. Herrera, F. Hoffmann, and L. Magdalena, "Genetic fuzzy systems evolutionary tuning and learning of fuzzy knowledge bases," Advances in Fuzzy Systems-Applications and Theory, NJ: World Scientific Publishing, vol. 19, 2001.

[35] J. J. Grefenstette, "Optimization of control parameters for genetic algorithms," IEEE Trans. Syst. Man Cybern., vol. 6, no. 1, 1986, pp. 122–128.

[36] F. J. Gomez, "Robust non-linear control through neuroevolution," Ph.D. Disseration, The University of Texas at Austin, 2003.

[37] C. J. Lin and Y. J. Xu, "A Self-Adaptive Neural Fuzzy Network with Group-Based Symbiotic Evolution and Its Prediction Applications," Fuzzy Sets and Systems, vol. 157, 2006, pp. 1036-1056.

AUTHORS PROFILE

Sheng-Fuu Lin (S'84–M'88) was born in Taiwan, R.O.C., in 1954. He received the B.S. and M.S. degrees in mathematics from National Taiwan Normal University in 1976 and 1979, respectively, the M.S. degree in computer science from the University of Maryland, College Park, in 1985, and the Ph.D. degree in electrical engineering from the University of Illinois, Champaign, in 1988. Since 1988, he has been on the faculty of the Department of Electrical and Control Engineering at National Chiao Tung University, Hsinchu, Taiwan, where he is currently a Professor. His research interests include image processing, image recognition, fuzzy theory, automatic target recognition, and scheduling

Hsien-Tse Chen was born in Taipei, Taiwan, R.O.C., in 1977. He received the B.S. degree in department of marine engineering from National Taiwan Ocean University, Keelung, Taiwan, R.O.C, in 1999 and the M.S. degree in department of automatic control engineering from Feng Chia University, Taichung, Taiwan, R.O.C, in 2002. He is currently pursuing the Ph. D. degree in the Department of Electrical and Control Engineering, the National Chiao Tung University, Hsinchu, Taiwan. His current research interests include image recognition, medicine image processing, machine vision.