Evolving Color Recipes

Mizutani, Eiji

Takagi, Hideyuki
九州芸術工科大学

David M. Auslander

Jyh-Shing Roger Jang

出版情報：IEEE transactions on systems, man, and cybernetics. Part C, Applications and reviews. 30 (4), pp.537-550, 2000-11. Institute of Electrical and Electronics Engineers
バージョン：

Evolving Color Recipes

Eiji Mizutani, *Student Member, IEEE*, Hideyuki Takagi, *Member, IEEE*, David M. Auslander, and Jyh-Shing Roger Jang, *Member, IEEE*

***Abstract*—This paper highlights an evolutionary computing intelligence for a computerized color recipe prediction that requires function approximation and combinatorial solution of colorants to produce color recipes for a given target color sample. We attack this real challenging problem in the color (paint) industry by using an evolutionary computing system that consists of a problem-specific knowledge and three principal constituents of soft-computing: neural networks, a fuzzy system, and a genetic algorithm. Departing from the recipe results obtained by neural networks (NN) approaches, the evolutionary system attempts to improve them in conjunction with fuzzy classification, a knowledge base and neural fitness functions. All components function synergistically in obtaining precise color recipe outputs through simulation of color paint manufacturing process. Such computational intelligence can be useful, especially when an exact mathematical model of the real-world process under consideration is not available explicitly.**

***Index Terms*—Color recipe prediction, computational intelligence, fuzzy systems, genetic algorithms, neural networks, soft computing.**

I. Introduction

COLOR is important to our daily lives; for instance, painting a room the proper color can enliven it and make it more comfortable. Painters often need to determine *color recipe* for producing a color specified by other individuals. In the color industry, it is important to develop scientific methods in calculating color recipes efficiently. For this purpose, the *Kubelka-Munk theory* has been widely used [23], [24]; however, it requires certain assumptions to formulate differential equations. In practice, those assumptions limit the situations where the theory may be applied [24]. Hence, a simple backpropagation multilayer perceptron (MLP) approach has been introduced as an alternative method to overcome practical obstacles in color recipe prediction [2], [13], [20].

This paper serves to introduce a computational intelligence technique for *color recipe prediction* that combines a knowledge base (KB) and three principal *soft computing* components: fuzzy systems (FS), neural networks (NN), and genetic algorithms (GA). When such constituents are put together, they function synergistically rather than competitively. Their mutual dependence may present unexpected performance enhancements. We shall demonstrate how the synergism of techniques surpasses the individual capacity of any one technique; color matching is an excellent test of these methods because it is difficult even for skilled human operators to do well, yet human color perception is sensitive, and therefore the matching must be done very well to meet acceptable standards.

In the next section, we explain the *color recipe prediction* task. We then present backpropagation (MLP) approaches, and a neuro-fuzzy approach in respective Sections III and IV. After that, we shall describe in detail our genetic-neuro-fuzzy approach.

Manuscript received June 9, 1996; revised February 9, 1997 and May 26, 2000. This work was supported by Kansai Paint Co., Ltd., Osaka, Japan.

E. Mizutani is with the Department of Computer Science, National Tsing Hua University, Hsinchu 30043, Taiwan, R.O.C. (e-mail: eiji@wayne.cs.nthu.edu.tw).

H. Takagi is with the Department of Art and Information Design, Kyushu Institute of Design, Fukuoka 815-8540, Japan (e-mail: takagi@kyushu-id.ac.jp).

D. M. Auslander is with the Department of Mechanical Engineering, University of California, Berkeley, CA 94720 USA (e-mail: dma@me.berkeley.edu).

J.-S. R. Jang is with the Department of Computer Science, National Tsing Hua University, Hsinchu 30043, Taiwan R.O.C. (e-mail: jang@cs.nthu.edu.tw; http://www.cs.nthu.edu.tw/~jang.).

Publisher Item Identifier S 1094-6977(00)09088-X.

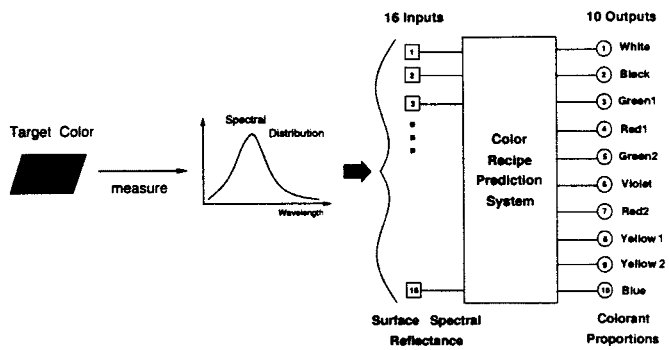

Fig. 1. Input–output relation in a color recipe prediction system. The inputs are sampled values of the surface spectral reflectance of a given target color, and the outputs are proportions of colorants.

II. Color Recipe Prediction

A real challenge in the color industry is *color recipe prediction*, which is a problem of computing a color recipe to match a sample color submitted by a customer. Technically, color recipe prediction often relates *surface spectral reflectance* of a target color to a list of *colorant proportions* that are needed to produce the same color as the given reference color, as shown in Fig. 1. In a practical situation, it is necessary to examine the color match in daylight as well as in artificial light. It is actually an arduous task even for professional *colorists*. The trained colorists have a remarkable ability to determine what colorants to be used and the direction and magnitude of the changes necessary in the colorant concentrations to improve the match by reference to their file of previous color recipes. More specifically, they first search color samples that are close enough to the given target color, and then adjust their color recipes by changing colorant proportions to match the reference color, as seen by human eyes. Those two skill-required procedures are summerized in Fig. 2, where we must emphasize two important associated criteria: *colorant proportion error* (or *color pigment concentration error*) and *color*

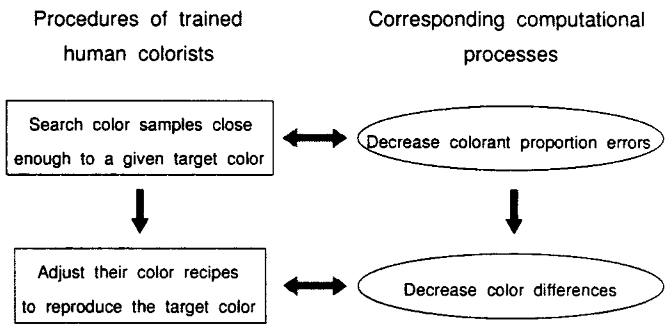

Fig. 2. Two important procedures of skilled human operators (or colorists) for color recipe prediction and their corresponding computational measures. Professional colorists first search their file of previous color recipes to find similar recipes for the given target color, and then adjust their color recipes by changing colorant proportions to match the reference color, as seen by human eyes. These two steps can be measured numerically by "colorant proportion error" and "color difference." The ultimate goal is to make the color difference small enough.

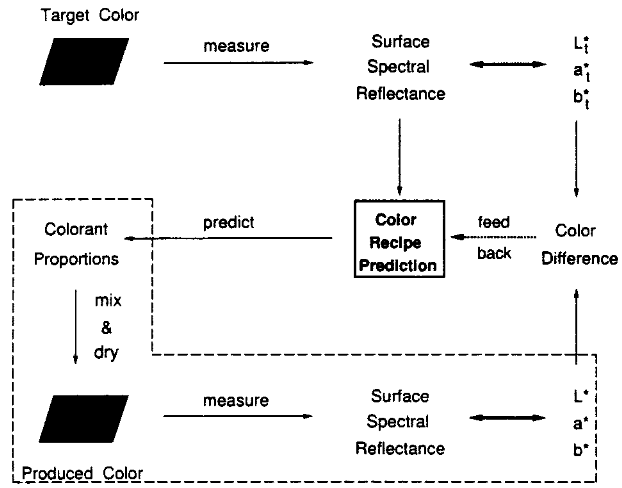

Fig. 3. Color paint manufacturing process. The dotted part includes time-consuming paint manufacturing based on predicted recipe results. The number of repetitions of this time-consuming process can be reduced by effective color recipe prediction.

difference. The *colorant error* shows how close to the previous recipe data, and the *color difference* indicates how much the newly-produced color from the predicted recipe is close to the target color, as perceived by human eyes.

In our recipe prediction problem, ten-dimensional colorant proportion vectors are considered as output (see Fig. 1). The "colorant error" is defined as

$$\text{Colorant error} \sqrt{\sum_{i=1}^{10}(t_i - o_i)^2} \tag{1}$$

where $(t_1, t_2, \ldots, t_{10})$ and $(o_1, o_2, \ldots, o_{10})$ are the colorant proportion vectors of a target color sample and of a produced color sample, respectively.

For evaluating color difference, we adopted CIE 1976 (L^*, a^*, b^*)-space, which provides a useful measure for determining "color differences" numerically [4], [24]. That is, it defines the color difference and perceptual attributes of color: "lightness," "hue," and "chroma" as shown in (2) at the bottom of the page, where L^*, a^*, and b^* are obtainable from surface spectral reflectance and (L_t^*, a_t^*, b_t^*) are the values of the target color. For details about the transformation from surface spectral reflectance to (L^*, a^*, b^*) (see [24]).

The goal of colorists is to decide the color recipe so that the color difference between a newly-produced color sample and the reference color is less than 1.0, because human eyes can hardly distinguish between smaller color differences [24]. To acheive this goal, the colorists' decision-making process inevitably involves a factor of "trial and error" to finalize color recipes until color difference becomes small enough. Fig. 3 shows the entire cycle of *color paint manufacturing*, in which the dotted part is usually time-consuming and labor-intensive.

A succinct description of the main concerns in the recipe prediction is summarized in Table I. Recall that the output vector is a list of ten colorant proportions; those ten outputs included three pairs of the same types of colorants (i.e., green, yellow, and red ones) and also complementary colorants such as "green and red," and "blue and yellow" (see Fig. 1). We must carefully determine which colorants to use, avoiding use of the same colorant types and complementary colorants to maintain acceptable cost performance. Since the desired average number of colorants required to produce any color was approximately four out of ten colorants (see Table II), this recipe prediction task involves aspects of *combinatorial* problems as well as those of nonlinear regression analysis.

For experimentation, we used 1446 training samples of Munsell color chips and 302 checking samples of standard paint color chips from the Japan Paint Manufacturers Association. Those data distributions on the $a^* - b^*$ plane are shown in Fig. 4. The input data consist of surface spectral reflectance of target colors sampled at 16 points in the visible range of color spectrum between 400 nm and 700 nm in wavelength (20-nm intervals). They were collected by using spectrophotometers [7]. All subsequent experiments were conducted using the same data sets.

$$\begin{aligned} \text{Color Difference} \quad & \sqrt{(L_t^* - L^*)^2 + (a_t^* - a^*)^2 + (b_t^* - b^*)^2} \\ \text{Lightness} \quad & L^* \\ \text{Hue} \quad & \tan^{-1}(b^*/a^*) \\ \text{Chroma} \quad & \sqrt{(a^*)^2 + (b^*)^2} \end{aligned} \tag{2}$$

TABLE I
MAIN CONCERNS IN COLOR RECIPE PREDICTION

(P1)	It is difficult to predict precise colorant concentrations. This task sometimes requires as low as 0.01%, which is the desired minimal colorant proportion level.
(P2)	It is necessary to specify use of a limited number of colorants to use to meet acceptable cost performance. In the choice of colorants, we need to avoid use of complementary colorants and of the same types of colorants.
(P3)	It is important to consider human visual sensitivity to color difference, which is closely related to perceptual attributes of color, i.e., lightness, hue, and chroma [24], [4].
(P4)	The magnitude of mean squared error of colorant vectors may not correspond exactly to that of color difference. The question is which colorant is dominant in the entire color.
(P5)	Any color can be uniquely identified by its surface spectral reflectance curve, i.e., its physical color attribute, but some combinations of colorants may have the same *perceptual* attributes of color as seen by humans.

TABLE II
NUMBER OF DATA CLASSIFIED BY THE DESIRED NUMBER OF COLORANTS REQUIRED TO PRODUCE COLOR RECIPES IN DATA SETS

	Two colorants desired	Three colorants desired	Four colorants desired	Desired average # of colorants to use
1446 training data	4	60	1,382	3.95
302 test data	0	13	289	3.96

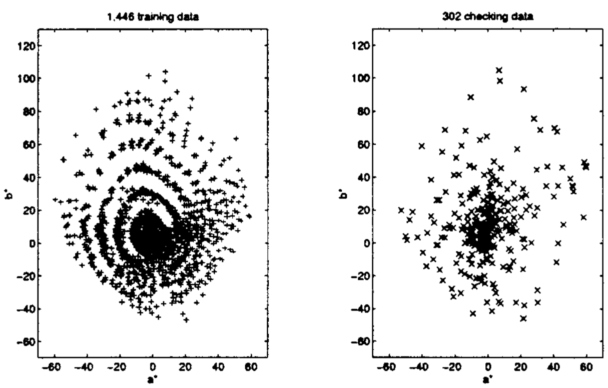

Fig. 4. Data distribution plots on the $a^* - b^*$ plane for (a) 1446 training data and (b) 302 checking data. There are relatively many data near the origin, because dark color samples are hard to predict their recipes.

III. MLP APPROACHES

Since MLPs are by far the most commonly employed NN structures for a wide range of applications, a simple MLP, $\mathrm{NN_{norm}}$, was first applied as a touchstone to the aforementioned recipe prediction to fathom its intrinsic difficulty [13]. It is then realized that the weakness of the simple MLP approach was due to the following reasons.

- Colorant selection is of great importance as indicated in (P2) of Table I, which is a sort of combinatorial problems. Table II shows the desired number of colorants in our data sets. The average number of colorants required to produce any color is fewer than five; this means six of the ten final outputs should be zero.
- We sometimes need to predict proportions with enough precision to specify levels such as 0.01% [Table I (P1)]. It is an important concern in color recipe prediction to specify such output range extremities [2], [13].

For instance, a sample color recipe might be given as follows:

White	Black	Red 1	Yellow 1
0.9754	0.0006	0.0028	0.0212

This recipe uses only four colorants and thus the other six colorant proportions must be zero. Since *black* colorant is most likely to have a significant influence on the entire color, its colorant proportion tends to be extremely small compared with the others, especially in bright color recipes.

To handle these concerns, we have introduced in MLPs *modified sigmoidal functions* and *truncation filter functions* in the output layer [9], [13]. Here we compare the two types of MLPs: $\mathrm{NN_{norm}}$ and $\mathrm{NN_{mod}}$; $\mathrm{NN_{norm}}$ has normal sigmoidal functions and $\mathrm{NN_{mod}}$ has *modified sigmoidal functions* in the output layer. Both $\mathrm{NN_{norm}}$ and $\mathrm{NN_{mod}}$ have the same model size ($16 \times 18 \times 21 \times 10$ neurons), mapping surface spectral reflectance of a target color (16 sampled inputs) to a list of required colorant concentrations (ten outputs) (see Fig. 1); those NNs were trained by using Polak-Ribiere's conjugate gradient methods [19]. Since the modified sigmoidal function prevents an NN from exceeding the desired output range, the outputs are further processed to eliminate redundant colorants at the minimum of the desired output range. The effects of the modified sigmoidal functions can be seen clearly in Fig. 5. The $\mathrm{NN_{norm}}$ tends to specify use of more colorants than necessary; it averages almost seven specified colorants, which is far from the ideal number of *about four*. On the other hand, in Fig. 5, the $\mathrm{NN_{mod}}$ shows the predicted number of colorants

TABLE III
PERFORMANCE COMPARISON OF SINGLE MLP APPROACHES: $\mathrm{NN_{norm}}$ AND $\mathrm{NN_{mod}}$, USING 302 CHECKING DATA. $\mathrm{NN_{norm}}$ IS A SIMPLE BACKPROPAGATION MLP, AND $\mathrm{NN_{mod}}$ IS AN MLP WITH MODIFIED SIGMOIDAL FUNCTIONS. THE COLUMN, "ERROR" DENOTES THE AVERAGE COLORANT PROPORTION ERROR

	Ave. # of colorants	Error $\times 10^{-2}$	# of test data which outputs same or complementary color				
			2Green	2Yellow	2Red	Red & Green	Yellow & Blue
NN_{norm}	6.66	2.616	97	154	125	198	129
NN_{mod}	3.90	2.031	14	7	5	0	1
Ideal	3.96	0	0	0	0	0	0

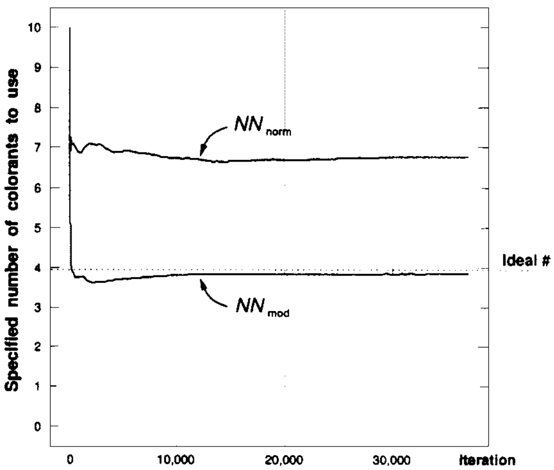

Fig. 5. Learning behaviors with respect to the number of necessary colorants predicted by an MLP with the normal sigmoidal functions ($\mathrm{NN_{norm}}$) and an MLP with the modified sigmoidal functions ($\mathrm{NN_{mod}}$).

asymptotically approached the ideal number of colorants as iterations progressed. The comparison of prediction accuracy between $\mathrm{NN_{norm}}$ and $\mathrm{NN_{mod}}$ is shown in Table III; $\mathrm{NN_{mod}}$ was more effective in avoiding use of the same types of colorants and of complementary colorants than $\mathrm{NN_{norm}}$. For more details about modified sigmoidal functions and truncation filter functions (see [11] and [13].

Recall our objective discussed in Section II; that is, the color difference should be lowered close enough to 1.0. In light of this criteria, the results obtained by $\mathrm{NN_{mod}}$ was not completely satisfactory, because the average predicted color difference was 2.847, as will be shown in Table VII. Indeed $\mathrm{NN_{mod}}$ did a better job than $\mathrm{NN_{norm}}$, but greater precision in concentration specification is desired.

IV. NEURO-FUZZY APPROACHES

Since some problem-specific knowledge can be obtained from professional colorists, we contend that knowledge-based approach, such as *fuzzy modeling*, must complement simple MLP's to enhance overall performance. In this section, we show how the knowledge is incorporated into NN models, resulting in *neuro-fuzzy* models, and how they can be generalized for application to color recipe prediction. Our neuro-fuzzy approaches are expressed within the framework of the CoActive Neuro-Fuzzy Inference System (CANFIS), detailed in [9], [11], and [12], which has enormous potential for augmenting the learning capacity of its predecessor ANFIS [5].

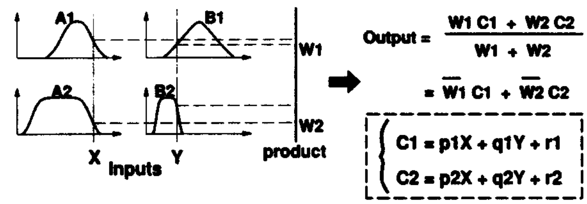

Fig. 6. Inferencing mechanism of a two-input, one-output Sugeno (TSK) fuzzy model, in which rules' consequents are linear.

A. Neuro-Fuzzy Inferencing Mechanism

This subsection briefly introduces a Sugeno-type (or TSK) fuzzy inference system [22] using Fig. 6, in which the system has two inputs (X and Y) and a single output. A typical fuzzy rule in the TSK fuzzy system has the form

$$\begin{aligned}&\text{Rule } i\text{: If } X \text{ is } A_i \text{ and } Y \text{ is } B_i,\\&\qquad \text{then } C_i = p_i X + q_i Y + r_i\end{aligned} \tag{3}$$

where A_i and B_i are linguistic terms characterized by proper fuzzy membership functions (MFs); $\{p_i, q_i, r_i\}$ are modifiable parameters.

The overall output is computed via weighted average

$$\text{Output} = \frac{W_1 C_1 + W_2 C_2}{W_1 + W_2} \tag{4}$$

where W_i are firing strengths defined as the product of membership grades on the antecedent part (see Fig. 6). In the original TSK model, C_i is a linear function of inputs. But it can be any function; for instance, an MLP (neural network) can be employed. CANFIS realizes such a rather complicated fuzzy inference model in the layered network architecture. For more details, see [12].

B. Fuzzy Partitionings

In fuzzy systems, the number of MFs should be carefully determined so that fuzzy rules can be held to meaningful limits. Considering these points, it must be a good idea to set up MFs for perceptual attributes of color such as "lightness," "hue," and "chroma" [4], [24] (see also Section V-F). Those values must be more suitable as MF inputs for treating color in a linguistically meaningful way than the 16 spectral values, which were used for MLP inputs.

When we consider one perceptual attribute of color "hue" as a linguistic variable, we can build up five fuzzy MFs according to the "hue" angle on the polar coordinates that define color gradation: "red ⇒ yellow ⇒ green ⇒ blue ⇒ violet ⇒ red." Fig. 7 (top) illustrates a *fuzzy membership value generation*; that is, if

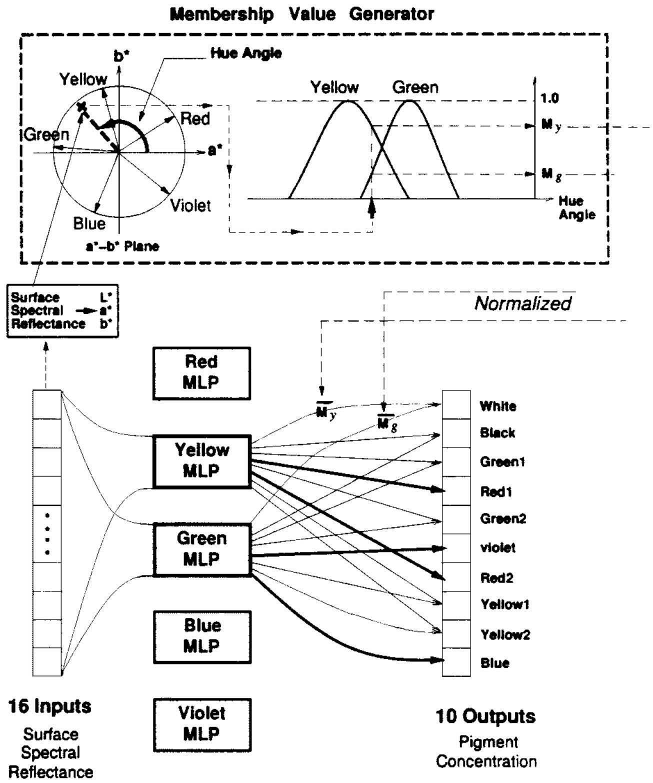

Fig. 7. CANFIS with five color rules for color recipe prediction. Each rule's consequent has been realized by an MLP. In the fuzzy membership value generator (top), five bell-shaped MF's are centered at red, yellow, green, blue, and violet regions according to the hue angle on the $a^* - b^*$ circular plane (see Fig. 4, also). If a given target color sample looks "greenish yellow," for instance, then it may be close to 125° in terms of the hue angle. In this case, the yellow MF and the green MF specify the degree of membership of their color regions; the larger degree of the yellow MF than that of the green MF $(M_y > M_g)$ is a useful measure for defining "greenish yellow" numerically.

a given color sample looks "greenish yellow," it may be close to 125° in terms of the hue angle on the $a^* - b^*$ circular plane, where the five center positions of those five color regions are indicated by arrows. The membership value of the yellow MF implies how yellow the presented color looks as seen by human eyes. In Fig. 7 (top), the yellow MF and the green MF specify the degree of membership of their color regions. The larger degree of the yellow MF than that of the green MF $(M_y > M_g)$ is a useful measure for defining "greenish yellow" numerically.

Fuzzy rules in the "if-then" format [cf. (3)] serve to determine color selection; for instance

$$\text{Yellow rule: If the target color is `yellow',}$$
$$\text{then use a yellow consequent } C_y.$$

Each of five color MFs specifies the degree of membership of its color region, and assigns the degree value to each color rule (rule's consequent) as the firing strength; see (4). In the above yellow rule, the firing strength is determined by the yellow MF because CANFIS in Fig. 7 has a single MF input "hue angle."

Introducing many more MFs might yield better results, because the number of adjustable parameters will be increased accordingly. But there is one important caveat from a fuzzy logic standpoint. When too many MFs are introduced, resulting fuzzy rules may be ill-defined, or hard to understand simply because of the difficulty of specifying the difference between "greenish yellow" and "very yellow" that humans perceive by saying "slightly greenish yellow," or using some other vague description [11]. Instead of increasing MF's, we can construct more sophisticated rules' consequents such as neural rules (or *local color expert* NN's), as suggested previously; that is, C_i in (3) can be an MLP. Fig. 7 illustrates such a CANFIS with five color neural rules. This CANFIS model can be viewed as a variant of the *modular networks* [11], [12]. The given prediction task is decomposed into five color rules using five local color expert MLPs, which form rules' consequents. For instance, the "green rule" has a neural consequent (green MLP) that has 16 spectral reflectance inputs.

CANFIS in Fig. 7 is constructed to handle an important aspect of color vision "hue angle" alone. Yet, of course, it is possible to construct another CANFIS that deals with the other two perceptual aspects: "lightness" and "chroma" so as to alleviate the problem (P3) in Table I. For details on such a CANFIS [11].

C. Knowledge-Embedded Structures

In Fig. 7, adaptive fuzzy MF's specify the degree of membership of five color regions (red, yellow, green, blue, violet) according to perceptual attributes of color. They determine what weight should be assigned to each rule's output to produce a final output. We have applied the colorist's knowledge to the CANFIS architecture so that several connections between local color experts and the final ten outputs can be pruned. For instance, the yellow expert has no effect on blue colorant proportions because of the yellow-blue complementary color relationship. This idea is pictured in Fig. 8 where the yellow MLP has just eight output units, fewer than the ten final output units; see Table IV for the size of all local experts as well as the number of output units (or neurons). As previously stated, the desired number of colorants should be about four; this means six of the ten final outputs should be zero. Reducing the number of zero outputs through the pruning procedure can possibly have a positive impact on the construction of the desired input-output mappings inside CANFIS. This modification is intended mainly to eliminate the problems of (P1) and (P2) in Table I.

V. Color Paint Manufacturing Intelligence

This section describes a cooperative hybrid system to simulate the entire manufacturing process in an attempt to construct evolutionary "manufacturing intelligence" for color recipe prediction. In particular, we integrate the three major elements of soft computing and problem-specific knowledge. To be concrete, NNs, an FS, and a GA with a KB complement each other in obtaining more precise recipe outputs than individual NN methods through simulation of the whole decision-making process of a professional colorist.

A GA may be a good choice for dealing with a combinatorial problem (P2) in Table I; hence, the GA plays a leading role in intermixing NN's, an FS, and a KB to evolve colorant recipe vectors (or chromosome).

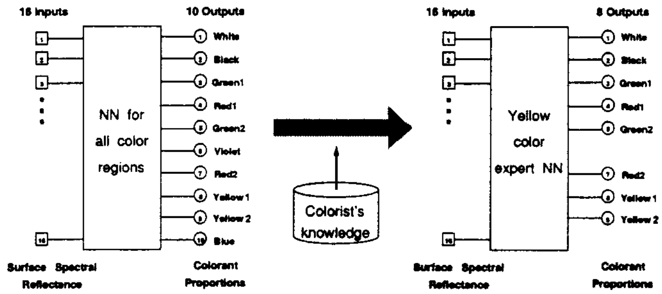

Fig. 8. Knowledge-embedded structure of the local expert NN (or MLP) for the yellow color region; the colorist's knowledge makes the yellow expert MLP have no connections to complementary colorant outputs: "blue and violet" outputs, although the 16 inputs remain the same as in the other local expert MLP's. Note that this yellow expert MLP corresponds to the yellow rule's consequent of CANFIS, denoted by "Yellow MLP" in Fig. 7.

A. Color Simulator Neural Networks

The aforementioned MLP's and CANFIS built up only the color recipe prediction system shown in Fig. 1 that does not use any feedback information concerning *color difference* explicitly. Yet, it is of great importance to consider perceived color difference during the recipe prediction process (see Table I and Fig. 3). Basically, our ultimate objective is to decrease color difference rather than colorant errors (see Fig. 2). In other words, proportion error measure might lead to rapid approach to nearly optimal recipe results, but color difference measure might be more important to fine-tune the recipe results. Practically, the color difference between pairs of presented colors should be smaller than about 1.0; human eyes can hardly distinguish between smaller color differences. If the predicted color recipe causes color difference greater than 1.0, then it may be necessary to readjust the color recipe.

The bird's-eye view of the paint production pictured in Fig. 3 gives us a hint about how to feed back the color difference information to the recipe prediction system in order to improve prediction accuracy. In particular, we have constructed an MLP, $\mathrm{NN}_{\mathrm{Lab}}$, specially designed to cope with the third critical problem (P3) in Table I. The $\mathrm{NN}_{\mathrm{Lab}}$ plays an important role as a color simulator in estimating color difference so that the entire system can mimic the whole paint production process, as depicted in Fig. 9. This manufacturing process can be expressed within an evolutionary framework, as illustrated in Fig. 10. We shall explain the evolutionary mechanism in subsequent sections.

B. Overlook of Manufacturing Intelligence

In the initial stage [left side of Fig. 10], the first-generation population, or starting points for a GA search are set by a fuzzy population generator and a multi-elite generator using results from the CANFIS and NN approaches. Those results must already be somewhat close to the range of ideal colorant concentrations. This initial stage corresponds to the first step of the colorist's operation described in Fig. 2. The difference is that the human colorists use their reference file of the stocked recipe records, which can be viewed as a sort of look-up table method. On the other hand, the manufacturing intelligence employs NN function approximators.

TABLE IV
OPTIMAL STRUCTURES OF FIVE LOCAL COLOR EXPERT MLPS IN CANFIS, AND THEIR INITIAL NUMBER OF TRAINING/TEST DATA CLASSIFIED INTO THE FIVE COLOR REGIONS. DURING THE LEARNING PHASE, MF PARAMETERS ARE UPDATED; HENCE, THE AMOUNT OF DATA INTO THE FIVE COLOR CATEGORIES CHANGES ACCORDINGLY. SINCE DIFFERENT AMOUNT OF TRAINING DATA GOES INTO EACH LOCAL COLOR EXPERT, EACH MLP MAY HAVE A DIFFERENT MODEL SIZE AND CAN BE OPTIMIZED FOR ITS OWN TERRITORY. THE STRUCTURES WERE OPTIMIZED BY A PROCESS OF TRIAL AND ERROR. NOTICE THAT EACH COLOR EXPERT'S OUTPUT UNITS ARE FEWER THAN THE FINAL 10 CANFIS OUTPUT UNITS, ACCORDING TO COLORISTS' KNOWLEDGE

Five Color Consequents	Model Size	Training Data	Checking Data
MLP_{Red}	16 × 16 × 16 × 8	650	138
MLP_{Yellow}	16 × 16 × 17 × 8	707	200
MLP_{Green}	16 × 21 × 7	521	105
MLP_{Blue}	16 × 15 × 8	363	65
MLP_{Violet}	16 × 17 × 6	409	48

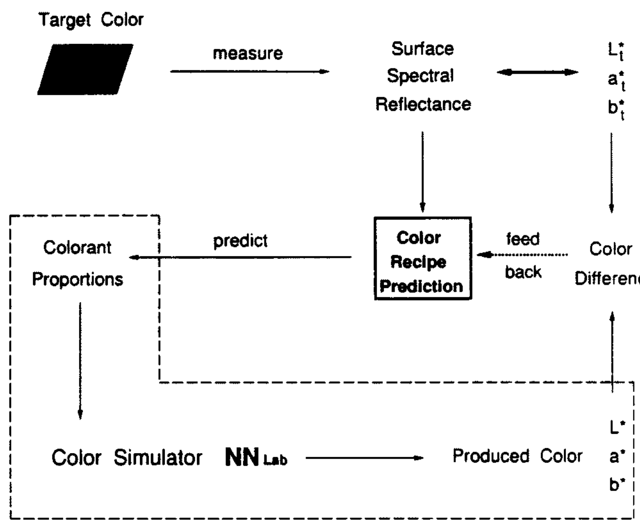

Fig. 9. Important role of $\mathrm{NN}_{\mathrm{Lab}}$ (in the paint manufacturing process) as a color simulator to predict what the produced color will look like. $\mathrm{NN}_{\mathrm{Lab}}$ replaces the time-consuming part of paint manufacturing (compare Fig. 3).

In the evolutionary phase, the system tries to improve colorant proportions encoded into chromosomes in conjunction with three functions, NNs and a KB, which form the fitness function. Genes' colorant concentrations are passed to the three functions which calculate fitness values individually, and then the three values are combined into the final fitness value. This evolutionary phase corresponds to the second step of the colorist's operation described in Fig. 2. In the following, we shed light on more details of each component in the evolutionary system.

C. Colorist's Knowledge Base

Performing the color recipe prediction task requires special knowledge, thus, a KB is constructed that has the following four main rules:

- *Rule 1*
  keep total proportions of colorants 100%;
- *Rule 2*
  keep the number of necessary colorants around the ideal number;

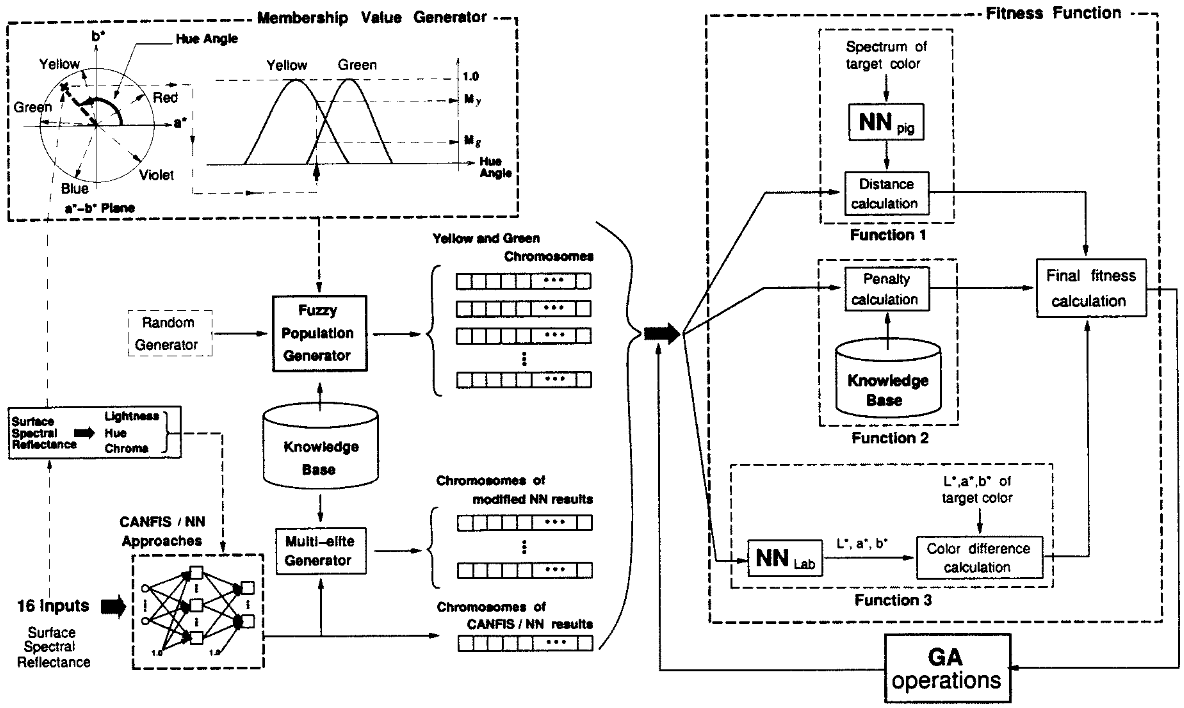

Fig. 10. Architecture of color paint manufacturing intelligence.

- *Rule 3*
  avoid use of complementary colorants, e.g., Red and Green;
- *Rule 4*
  avoid use of the same type of colorants at the same time: e.g., Red_1 and Red_2.

Note that we have ten colorants (ten outputs) that include three pairs of the same kind of colorants: green, yellow, and red ones (see Fig. 1); each pair, such as Red_1 and Red_2, has different characteristics. In the color recipe prediction task, the 100% rule (Rule 1) was also emphasized in [20].

Knowledge may be useful in reinforcing some favorable aspects of genetic searches [8]. Thus, the KB might play an important role in helping the hybrid system evolve to recognize specific features of a target color.

D. Multi-Elites Generator

The resultant color recipes obtained by $\mathrm{NN_{norm}}$, $\mathrm{NN_{mod}}$, and CANFIS are encoded into the initial population as elite members. Then, a multi-elite generator produces more elites by modifying those results according to Rule 4 in the KB. That is, the concentrations of the same type of colorants are summed into one or another of them, e.g.,

$$\mathrm{Red}_1 + \mathrm{Red}_2 \Rightarrow \mathrm{Red}_1, \text{ or}$$
$$\mathrm{Red}_1 + \mathrm{Red}_2 \Rightarrow \mathrm{Red}_2.$$

This is derived from the fact (in Table III) that the simple backpropagation MLP, $\mathrm{NN_{norm}}$, tends to specify use of more than six colorants although the desired number of colorants to produce any color in our data sets is fewer than five. Table V shows two sample color recipes obtained by the aforementioned NN approaches. Again, it is important to keep the number of colorants used at a practical level.

The following table shows a sample of initial multiple elites produced by the multi-elite generator, and their associated color difference predicted by $\mathrm{NN_{Lab}}$:

	Colorant error	Color difference by $\mathrm{NN_{Lab}}$	Fitness value
CANFIS	0.000 748	1.405	0.816 236
$\mathrm{NN_{mod}}$	0.008 070	1.706	0.836 720
$\mathrm{NN_{norm}}$	0.006 765	7.121	0.435 954
Elite 1	0.005 266	5.833	0.800 402
Elite 2	0.029 944	7.072	0.734 058
Elite 3	0.030 417	9.961	0.689 702
Elite 4	0.007 502	8.660	0.733 926

In this example, four new elites are generated by modifying the $\mathrm{NN_{norm}}$'s recipe vector. The four newly generated elites (Elites 1 through 4) have different fitness values, which are higher than $\mathrm{NN_{norm}}$'s fitness value, because knowledge has been applied to improve the recipe vector obtained by $\mathrm{NN_{norm}}$.

Multiple elite colorant vectors offer several different good starting points for GA searches. The number of encoded elites depends on the quality of the CANFIS/NN results; we take the results of three approaches ($\mathrm{NN_{norm}}$, $\mathrm{NN_{mod}}$, CANFIS), and so at least three elite members always exist at the initial stage. The combination of several solutions may be effective in finding an optimal solution [6]. The other members are initialized by a fuzzy population generator. This seeding procedure is shown in the left side of Fig. 10.

TABLE V
TWO SAMPLE COLOR RECIPES, COMPARED WITH THE RECIPES OBTAINED BY $\mathrm{NN}_{\mathrm{norm}}$, $\mathrm{NN}_{\mathrm{mod}}$, AND CANFIS; THOSE ARE ENCODED INTO THE INITIAL POPULATION

Sample 1

Colorant	target	NN_{norm}	NN_{mod}	CANFIS
White	0.9561	0.9518	0.9545	0.9580
Black	0.0121	0.013	0.0134	0.0119
Green_1	0	0.009	0	0
Red_1	0	0.0011	0	0
Green_2	0.0317	0.0233	0.0321	0.0275
Violet	0	0	0	0
Red_2	0	0	0	0
Yellow_1	0.0001	0.0008	0	0.0023
Yellow_2	0	0.0004	0	0.0003
Blue	0	0	0	0

Sample 2

Colorant	target	NN_{norm}	NN_{mod}	CANFIS
White	0.0933	0.0887	0.0939	0.0936
Black	0.0352	0.0332	0.0410	0.0363
Green_1	0	0	0	0
Red_1	0.2673	0.2643	0.2608	0.2664
Green_2	0	0.0051	0	0
Violet	0	0	0	0
Red_2	0	0.0003	0	0
Yellow_1	0.6042	0.5988	0.6043	0.6037
Yellow_2	0	0.0097	0	0
Blue	0	0	0	0

E. Fuzzy Population Generator

The CANFIS concept in Fig. 7 is adapted to initialization process [see the left side of Fig. 10]. The idea is to generate the initial population according to the fuzzy classification of a target color, which serves to determine colorant selection. First, we classify the target color into one of five color categories (red, yellow, green, blue, and violet) on the $a^* - b^*$ plane, which shows hue and chroma [see (2)], and decide to what extent the desired color belongs to each color category using fuzzy MFs, as discussed in Section IV-B. We then generate initial color chromosomes by modifying chromosomes generated by a random number generator according to rules in the KB. For example, when a target color looks greenish yellow, green chromosomes and yellow ones are generated; Green chromosomes have zero values in either Green_1 or Green_2 colorant concentration and in red colorant concentrations because of the red-green complementary color relationship (see Rule 3 and Rule 4 in Section V-C). It is effective to inactivate some genes which have information on the same type of colorants and complementary colorants in order to eliminate redundant colorants at the initial stage.

The number of green chromosomes ($\mathrm{Num}_{\mathrm{Green}}$) and that of yellow ones ($\mathrm{Num}_{\mathrm{Yellow}}$) are decided according to the following calculations:

$$\begin{aligned}\mathrm{Pop}_{\mathrm{rest}} &= \mathrm{Pop}_{\mathrm{total}} - \mathrm{Pop}_{\mathrm{NN}},\\ \mathrm{Num}_{\mathrm{Green}} &= \frac{M_g}{M_y + M_g}\mathrm{Pop}_{\mathrm{rest}},\\ \mathrm{Num}_{\mathrm{Yellow}} &= \frac{M_y}{M_y + M_g}\mathrm{Pop}_{\mathrm{rest}}\end{aligned}$$

where two membership values, M_y and M_g, signify to what extent the target color belongs to the yellow category and the green one, respectively. $\mathrm{Pop}_{\mathrm{total}}$ denotes the total population number, and $\mathrm{Pop}_{\mathrm{NN}}$ signifies the number of elite chromosomes from the CANFIS/NN results including the chromosomes generated by the multi-elite generator.

F. Fitness Function

The fitness function consists of three functions; two neural fitness functions (Function 1 and Function 3), and the KB-based fitness function (Function 2). Its form can be expressed as

$$\mathrm{fitness} = \alpha \cdot \mathrm{fitness}_1 + \beta \cdot \mathrm{fitness}_2 + \gamma \cdot \mathrm{fitness}_3 \tag{5}$$

where α, β, and γ are scaling factors such that the total fitness value is scaled to 1.0.

1) Function 1: The first function evaluates genes' colorant concentration vectors according to the use of colorants specified by $\mathrm{NN}_{\mathrm{pig}}$. The $\mathrm{NN}_{\mathrm{pig}}$ ($16 \times 18 \times 21 \times 10$ neurons) maps surface spectral reflectance to a list of required colorants (see Fig. 11). It gives just ON/OFF values to each output unit to predict which colorants should be used to produce the same color as the target color, where ON means "colorant needed" and OFF means "not needed." Function 1 evaluates each chromosome by calculating the Euclidean distance in binary space (ON/OFF) after each chromosome's representation has been transformed into the ON/OFF format as follows:

$$\mathrm{fitness}_1 = 10.0 - \sqrt{\sum_{i=1}^{10}(t_i - b_i)^2} \tag{6}$$

where $(t_1, t_2, \ldots, t_{10})$ and $(b_1, b_2, \ldots, b_{10})$ are ten-dimensional binary vectors of $\mathrm{NN}_{\mathrm{pig}}$ output and of an evolving color chromosome, respectively. The calculated $\mathrm{fitness}_1$ is plugged into (5).

Fig. 11 describes this procedure. Table VI shows the capability of this trained $\mathrm{NN}_{\mathrm{pig}}$.

2) Function 2: The second function calculates a fitness value based on the KB described in Section V-C. The fitness value depends on the extent to which genes' colorant concentration vector obeys the rules in the KB. To keep the GA search moving in a consistent direction, the KB is used in both the initial stage and in the calculation of fitness values as illustrated in Fig. 10. Function 2 computes the following:

$$\begin{aligned}\mathrm{fitness}_2 = {} & \left(k_1 - \left\|1.0 - \sum_{i=1}^{10} o_i\right\|\right)\\ & + \left(k_2 - \left\|3.95 - \sum_{i=1}^{10} b_i\right\|\right)\\ & + (k_3 - \#\text{ of complementary colorants})\\ & + (k_4 - \#\text{ of the same type of colorants})\end{aligned} \tag{7}$$

where vector $(o_1, o_2, \ldots, o_{10})$ is a ten-dimensional colorant proportion vector encoded in an evolving color chromosome;

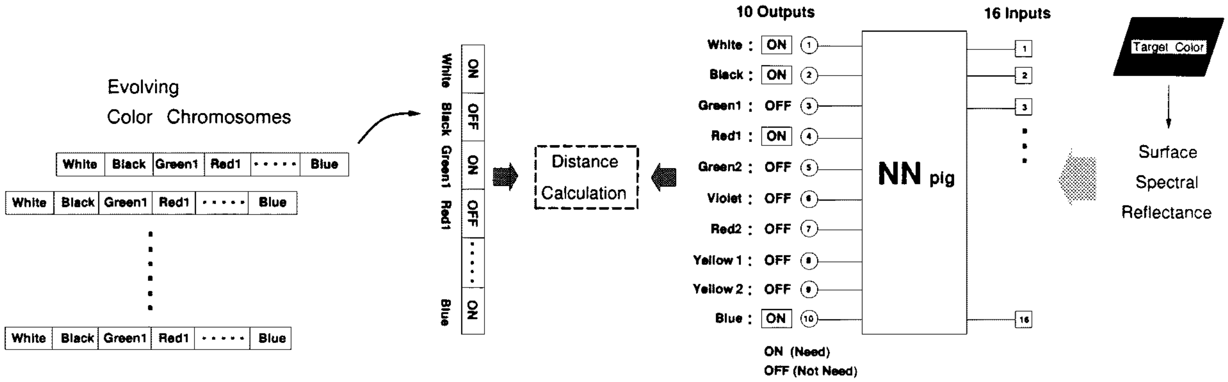

Fig. 11. Component of the fitness function based on $\mathrm{NN_{pig}}$ [see (6)].

TABLE VI
CAPABILITIES OF DIFFERENT NN APPROACHES IN SPECIFYING NECESSARY COLORANTS. $\mathrm{NN_{norm}}$ IS THE SIMPLE BACKPROPAGATION MLP, AND $\mathrm{NN_{mod}}$ IS THE IMPROVED $\mathrm{NN_{norm}}$ AS DISCUSSED IN SECTION III; CANFIS IS THE NEURO-FUZZY MODEL DESCRIBED IN SECTION IV. $\mathrm{NN_{pig}}$ IS THE SPECIAL NN THAT PREDICTS NECESSARY COLORANTS AS SHOWN IN FIG. 11

	NN_{norm}	NN_{mod}	CANFIS	NN_{pig}
# of unmatched patterns in 302 test patterns	299	74	73	27
# of unmatched units in 3,020 output units	911	106	98	48
Predicted avg. # of required colorants	6.66	3.90	3.89	3.96

vector $(b_1, b_2, \ldots, b_{10})$ is a corresponding transformed binary (ON/OFF) vector; and k_1, k_2, k_3, k_4 are some positive coefficients that make the four parenthesized values *positive*. The computed fitness$_2$ is used in (5).

3) Function 3: The third function, based on $\mathrm{NN_{Lab}}$, generates a fitness value with respect to color difference between a target color and each member's color whose colorant concentrations are predicted by the system. Because it is time-consuming to manufacture actual color paint by mixing colorants specified by genes' values, the $\mathrm{NN_{Lab}}$ plays a crucial role as a color simulator to predict what color will be produced (see Fig. 9). The $\mathrm{NN_{Lab}}$ ($10 \times 11 \times 14 \times 3$ neurons) maps colorant concentrations to L^*, a^*, and b^*; that is, by plugging each member's colorant proportions into $\mathrm{NN_{Lab}}$, we can obtain L^*, a^*, and b^* to calculate the color difference between a target color and an individual color (see Fig. 12).

The calculated color difference shows how satisfactorily the predicted color matches the reference color. Recall that human eyes can hardly distinguish two color samples if their color difference is smaller than 1.0. The use of $\mathrm{NN_{Lab}}$ provides a way to calculate color differences numerically, and thus to take into account human visual sensitivity to color differences. Table VII shows the potential of the color simulator $\mathrm{NN_{Lab}}$.

Function 3 determines the fitness value ($\mathrm{fitness_3}$) of each chromosome based on

$$\mathrm{fitness_3} = \exp(-E) \tag{8}$$

where E denotes the color difference calculated by (2). Other decreasing functions can be employed alternatively. The calculated fitness$_3$ goes to (5).

G. Genetic Strategies

GA search is controlled by genetic operations, which might have a significant effect on the quality of solutions. We have embodied some ideas special to the color recipe prediction in both mutation and crossover operations.

1) Mutation Strategy: Usual mutation operation as in a simple GA [3], [21] is applied to all members with a changeable mutation rate scheme such that a fixed mutation rate (0.01) is adopted with a probability of 0.4, and otherwise, a mutation rate ranging from 0.09 to 0.69, is decided using a random number. Moreover, the following modified operations are also considered.

- *Chromosome Template*:
  To avoid specifying use of more colorants than necessary, we set out to inactivate some genes using the fuzzy population generator as described in Section V.E. This has made it possible to use a chromosome itself as a template to do the mutation operation. That is, before the mutation operation, it is decided whether to mutate an inactivated gene or not; the mutation is applied with low probability (0.1) to inactivated genes, which have zero values of concentrations. If the mutation is applied to an inactivated gene, this leads to an increase in the number of necessary colorants.
- *Local Search and Preservation of Multi-elites*:
  Multi-elites, i.e., chromosomes from the results of CANFIS/NN approaches, are mutated only at the lower bits of each gene to keep traits similar to the NN results. Those mutant copies of the multi-elites may stay in the vicinity of the original multi-elites. In this way, local search of the NN results is realized. In addition, the offspring of multi-elites always advance to the next generation. The mutant copies of multi-elites are preserved throughout the entire evolution. Note that this manipulation of low-order bits is applied only to multi-elites.
- *Exchanging Mutation*:
  After the usual mutation, with low probability, members are subjected to another mutation: exchanging genes

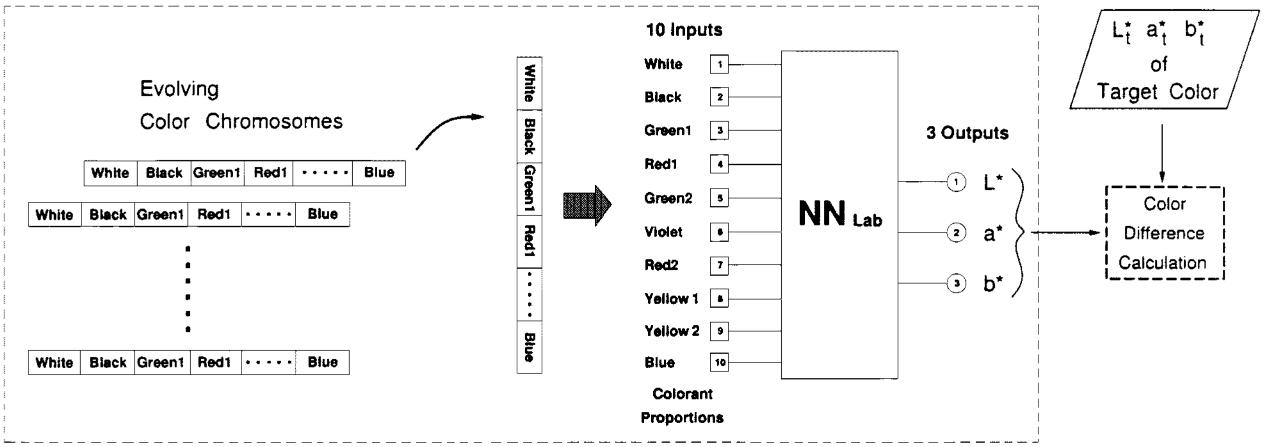

Fig. 12. Component of the fitness function based on $\mathrm{NN_{Lab}}$. The $\mathrm{NN_{Lab}}$ predicts what color will be produced after genes' specified colorant proportions are mixed.

TABLE VII
AVERAGE COLOR DIFFERENCE PREDICTED BY $\mathrm{NN_{Lab}}$ USING THE IDEAL COLORANT CONCENTRATIONS, AND THE RESULTS OF THREE NN APPROACHES: $\mathrm{NN_{norm}}$, $\mathrm{NN_{mod}}$, AND CANFIS. THIS TABLE SHOWS THE POTENTIAL CAPABILITY OF $\mathrm{NN_{Lab}}$ FOR 302 CHECKING DATA

Colorant vectors	Ideal	NN_{norm}	NN_{mod}	CANFIS
Color difference	0.567	5.921	2.847	1.976

that have the same type of colorant information. This mutation is illustrated in Fig. 13. Among ten output colorant proportions, we have three pairs of the same types of colorants (e.g., $\mathrm{Red_1}$ and $\mathrm{Red_2}$); so, we must decide which one to use. This exchanging mutation allows us to explore such colorant choices. This may lead to an escape from local optima in the initial CANFIS/NN and $\mathrm{NN_{pig}}$ results; their choices may not match the final choice determined by the system. Later in Table IX, we will show the agreement with $\mathrm{NN_{pig}}$; namely, how much the resultant choice of colorants optimized by the system matched the colorant choices specified by $\mathrm{NN_{pig}}$.

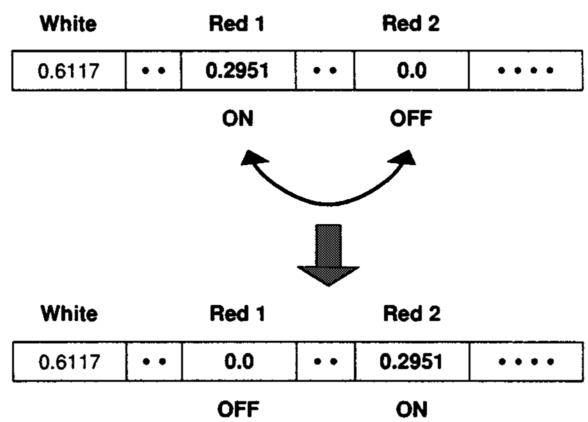

Fig. 13. Exchanging mutation.

2) Modified Simplex Crossover: Instead of using ordinary two-point crossover method, we employed the *simplex crossover* method, detailed in [1]. We have modified the selection method of the original simplex crossover, resulting in the following three procedures:

1) select one good chromosome with respect to fitness value;
2) pick, with high probability, a multi-elite, i.e., one of the mutant copies from the initial CANFIS/NN results, as a good chromosome;
3) choose one bad chromosome with respect to fitness value.

The procedures share an idea of the *Nelder-Mead downhill simplex method* [18], based on a reflection away from a bad chromosome, as illustrated in Fig. 14. Procedure 1) lights a direction toward minimizing color difference since a chromosome with high fitness most likely has small color difference due to (8) with $\mathrm{NN_{Lab}}$. In this GA search, it is desirable to find a direction that minimizes both color difference and colorant errors. The problem is that we cannot calculate colorant errors directly; however, the CANFIS/NN results provide a clue as to better colorant concentrations since they must already be within some range of the ideal colorant concentrations. That is why mutant copies from the CANFIS/NN results, including ones originally generated by the multi-elite generator, should be involved in guiding the search toward better colorant proportion vectors as in procedure 2). And then procedure 3) completes the simplex crossover, as depicted in Fig. 14. These three procedures were motivated by the colorists' skillful procedures described in Fig. 2.

VI. EXPERIMENTS OF MANUFACTURING INTELLIGENCE

The performance of the evolutionary color paint "manufacturing intelligence" was evaluated by actually manufacturing color paint samples according to the experimental results. Due to the time constraints in the usual production schedule and limited manufacturing capacity, 111 checking data were randomly selected for the performance evaluation.

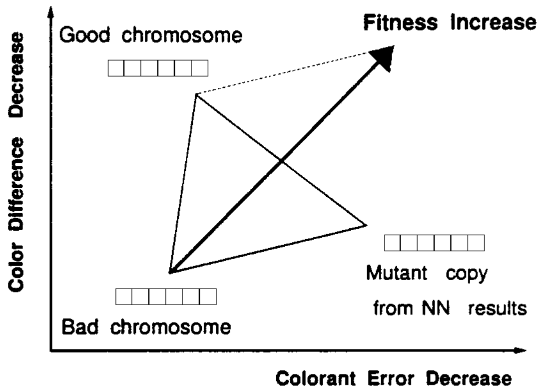

Fig. 14. GA search control by the modified simplex crossover. Two selected chromosomes are called "good" or "bad" according to their fitness value, and the third chromosome is a mutant copy from the CANFIS/NN results.

The configuration of the GA was as follows:

Population size	80 members
Mutation rate	flexible
Crossover method	simplex crossover [1]
Simplex crossover rate	0.85
Maximum generations	10 000.

Table VIII shows the comparison of the evolutionary system $\mathrm{GNF_{ALL}}$ and the aforementioned CANFIS/NN approaches. $\mathrm{GNF_{ALL}}$ with all three components of the fitness function, employed the results of three approaches: $\mathrm{NN_{norm}}$, $\mathrm{NN_{mod}}$, and CANFIS in generating the initial population. According to the corresponding color difference predicted by $\mathrm{NN_{Lab}}$, only the result of $\mathrm{GNF_{ALL}}$ was good enough to reach a satisfactory level of color difference where human eyes could not tell the difference between presented colors. (Again note that the desired color difference should be smaller than about 1.0.)

To exhibit how indispensable CANFIS/NN results are at the initial seeding stage, we examined $\mathrm{GNF_{ALL}^{void}}$, which had no multi-elites from CANFIS/NN results, but had the same fitness function as $\mathrm{GNF_{ALL}}$. It started the GA search from the randomly initialized colorant proportion vectors. Note in Table IX that the parenthesized values show the best performance with respect to colorant errors regardless of fitness; those were the results when the minimal colorant errors were obtained.

Furthermore, to demonstrate the validity of each of the three components in the fitness function, we tested GNF_C, $\mathrm{GNF_{CP}}$, and $\mathrm{GNF_{CK}}$: GNF_C had $\mathrm{NN_{Lab}}$ as the only component of the fitness function; $\mathrm{GNF_{CP}}$ had both $\mathrm{NN_{Lab}}$ and $\mathrm{NN_{pig}}$ as two components of the fitness function; $\mathrm{GNF_{CK}}$ had the KB as well as $\mathrm{NN_{Lab}}$ as two components. (Note that $\mathrm{NN_{Lab}}$ played an important role as a color simulator; hence, it always had to stay in the fitness function.) Table IX summerizes the results, showing how each component contributed to the prediction, and how they complemented each other.

Fig. 15 shows a sample evolutionary process of $\mathrm{GNF_{ALL}}$. The system selects a chromosome with the highest fitness as the final solution over a preset number of generations (10 000). For instance, in a sample evolution process in Fig. 15, the final outcome is obtained at generation 4468, as shown in Table X.

TABLE VIII
RESULTS OF COMPUTATIONAL PREDICTION IN COLORANT ERROR ($\times 10^{-2}$) AND THE CORRESPONDING COLOR DIFFERENCE PREDICTED BY $\mathrm{NN_{Lab}}$ USING 111 CHECKING DATA. $\mathrm{NN_{norm}}$ IS A SIMPLE BACKPROPAGATION MLP, AND $\mathrm{NN_{mod}}$ IS AN IMPROVED $\mathrm{NN_{norm}}$, AS DISCUSSED IN SECTION III; CANFIS IS A NEURO-FUZZY MODEL DESCRIBED IN SECTION IV; GNF IS A GENETIC NEURO-FUZZY MODEL

	Ideal	NN_{norm}	NN_{mod}	CANFIS	GNF_{ALL}
Colorant error ($\times 10^{-2}$)	0	2.312	1.543	1.139	0.643
Color difference predicted by NN_{Lab}	0.588	6.661	3.165	2.019	0.267

VII. DISCUSSION

In reality, it is very difficult to obtain the perfect fitness function to evaluate a complicated industrial process, especially when its precise mathematical model is not available. To overcome the limitation, we employed NN function approxmators, $\mathrm{NN_{pig}}$ and $\mathrm{NN_{Lab}}$, as components of the fitness function. Although those NN's were not perfect, as shown in Tables VI and VII, they helped direct the GA search to a better region of the search space; consequently, the performance of $\mathrm{GNF_{ALL}}$ was still better than those of other approaches (see Tables VIII and IX).

When $\mathrm{NN_{Lab}}$ solely acted as the fitness function in GNF_C, the system tended to go too far toward minimizing color difference, and therefore the average number of required colorants was larger. In addition, the specified colorants did not match well those designated by $\mathrm{NN_{pig}}$ as indicated in the low percentage of agreement with $\mathrm{NN_{pig}}$ in Table IX.

Intriguingly enough, $\mathrm{GNF_{CP}}$ and $\mathrm{GNF_{CK}}$ have comparable $\mathrm{NN_{pig}}$-agreements, and $\mathrm{GNF_{ALL}}$ were still better than those two in the $\mathrm{NN_{pig}}$-agreement. This indicates that the KB surely helped the system evolve to recognize important aspects in colorant selections, compensating for the lack of accuracy of $\mathrm{NN_{pig}}$ (Table VI). Here, it is emphasized that they functioned synergistically.

$\mathrm{GNF_{ALL}^{void}}$ had no multi-elites but had all three components of the fitness function, starting from the randomly initialized colorant concentration vectors. Its poor performance in Table IX places emphasis on the existence of the multi-elites (i.e., mutant copies from the CANFIS/NN approach results); without them, we cannot draw any advantage from the search direction presented in Fig. 14. In other words, seedings from CANFIS or other NN approaches are indispensable in enabling "manufacturing intelligence" to function efficiently. This $\mathrm{GNF_{ALL}^{void}}$ case corresponds to a situation where professional colorists have no access to their own file of previous color recipes, leading to very time-consuming efforts to generate near-optimal recipes.

As shown in the parenthesized values in Table IX, the system did not put the highest fitness on the 'best' chromosome in terms of colorant errors (see Table X also). In this simulation, we did not use the elitist selection method since the fitness function could not calculate colorant errors, which may suggest that even if a better child chromosome in terms of colorant error appears, the elitist strategy may jeopardize its chance of advancement to the next generation. In fact, when the elitist selection strategy was used, the system had a tendency to go too far toward minimizing color difference [10], [13].

TABLE IX
PERFORMANCE EVALUATION OF "MANUFACTURING INTELLIGENCE" SYSTEMS FOR 111 CHECKING DATA. $\mathrm{GNF_{ALL}}$ SHOWS THE MAXIMAL ABILITY OF THE MANUFACTURING INTELLIGENCE IN THE PREDICTION TASK. PARENTHESIZED VALUES SHOW POTENTIAL CAPABILITIES WITH RESPECT TO COLORANT ERRORS; THEY WERE OBTAINED WHEN COLORANT ERRORS WERE MINIMIZED. A COLUMN, "AVG. # OF GENERATION" SHOWS WHEN MAXIMAL FITNESS IS REACHED. A COLUMN, "ERROR" DENOTES THE AVERAGE COLORANT ERROR. A COLUMN, "AGREEMENT WITH $\mathrm{NN_{pig}}$" IMPLIES HOW MUCH THE PREDICTED CHOICE OF COLORANTS OPTIMIZED BY THE SYSTEM MATCHES THE COLORANT CHOICE SPECIFIED BY $\mathrm{NN_{pig}}$, AND THE LAST COLUMN, "COLOR DIF. BY $\mathrm{NN_{Lab}}$" SIGNIFIES COLOR DIFFERENCE PREDICTED BY THE COLOR SIMULATOR, $\mathrm{NN_{Lab}}$

	Fitness Function			Avg. # of generation	Error $\times 10^{-2}$	Avg. # of colorants	Agreement with NN_{pig}	Color dif. by NN_{Lab}
	NN_{Lab}	NN_{pig}	KB					
GNFALL	○	○	○	5058.7 (4759.5)	0.643 (0.213)	3.90 (3.88)	79.28% (79.28%)	0.267 (0.809)
GNFvoid	○	○	○	4134.4 (3082.8)	72.209 (36.165)	3.94 (4.89)	50.45% (21.62%)	48.800 (34.637)
GNFCP	○	○	×	4915.4 (4358.8)	1.190 (0.206)	4.02 (4.02)	78.38% (74.77%)	0.121 (0.659)
GNFCK	○	×	○	4559.3 (4655.5)	1.695 (0.215)	3.88 (3.89)	74.77% (77.48%)	0.202 (0.656)
GNFC	○	×	×	4604.3 (4742.4)	2.802 (0.191)	5.35 (4.36)	28.83% (55.86%)	0.060 (0.567)

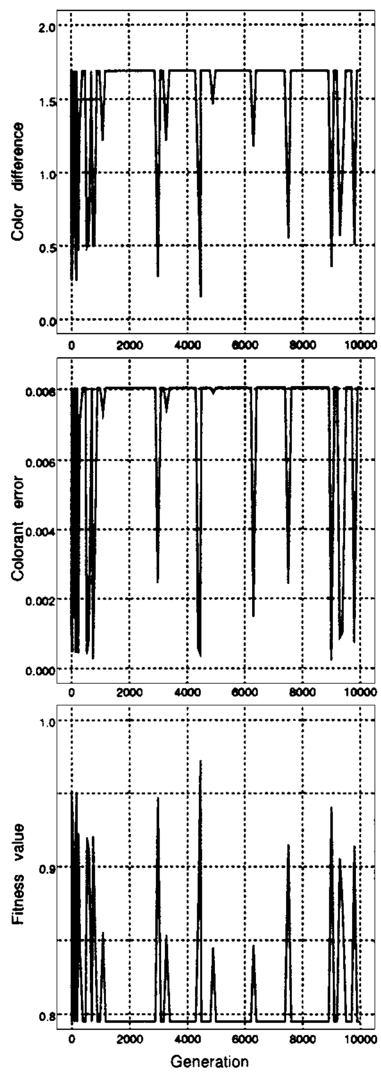

Fig. 15. Sample evolutionary process without the elitist selection strategy (see Table X).

The parenthesized colorant errors in Table IX show almost the same error level (0.2×10^{-2}) except for that of $\mathrm{GNF_{ALL}^{void}}$, although the colorant errors of the final solutions of the system are quite different. Actually, only 71 patterns among the 111 checking patterns were improved in terms of colorant error. This may be partly because the CANFIS models worked well in prediction, so their results may be hard to improve upon, but partly also because the system may happen to find another colorant composition solution. In other words, the presented "manufacturing intelligence" can potentially handle it if the color simulator, $\mathrm{NN_{Lab}}$, learns much of the mapping from colorant compositions to perceptual attributes of color (L^*, a^*, and b^*).

TABLE X
SAMPLE OF EVOLVED COLOR RECIPES THAT WERE SORTED ACCORDING TO THE FITNESS VALUES. THE RECIPE WITH THE HIGHEST FITNESS AT GENERATION 4468 WAS SELECTED AS THE FINAL OUTCOME. NOTE THAT THE COLOR DIFFERENCE WAS PREDICTED BY $\mathrm{NN_{Lab}}$ (SEE FIG. 15)

Generation number	Colorant error	Color difference by NN_{Lab}	Fitness value
4468	0.000416	0.147402	0.972258
173	0.002407	0.260854	0.950668
16	0.002416	0.269533	0.948915
3000	0.002440	0.282920	0.947709
9000	0.000211	0.349964	0.940885
27	0.000481	0.368336	0.938377
157	0.000444	0.455951	0.924514
251	0.000431	0.468585	0.922556
751	0.000272	0.500607	0.920708
532	0.000408	0.491637	0.92007
7500	0.002426	0.546673	0.915025
9800	0.000724	0.510859	0.914321

Fig. 16 shows an interesting fact that the real perceived color difference did not exactly correspond to the magnitude of colorant errors. Such complicated relationships between colorant errors and actual color differences may imply that the mapping from surface spectral reflectance to a list of colorants may not be a one-to-one correspondence. (As stated in Table I, we may need to take care of the (P4) and (P5) problems; different colorant compositions may produce the same or almost the same color to human perception.)

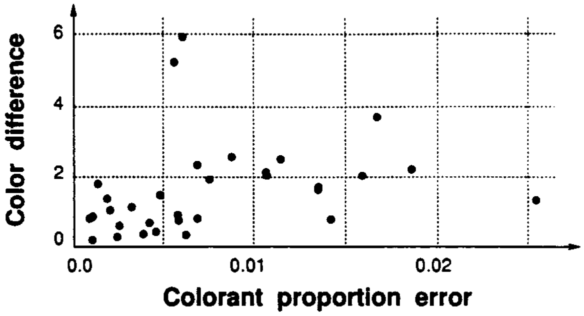

Fig. 16. Complicated relationships of 32 color samples between actual color difference and colorant proportion errors; these 32 sample color paints were actually manufactured and their color differences were actually measured.

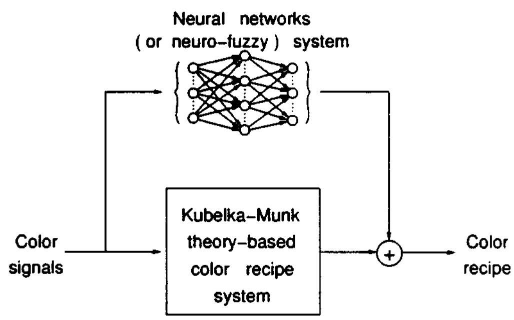

Fig. 17. Kubelka-Munk theory-based NN system; neural networks (or neuro-fuzzy) models compensate for the conventional Kubelka-Munk theory-based system.

Also, the smallest colorant error calculated by (1) may not be the best solution for human color perception. We may need weighted colorant error calculations in place of (1); in a bright color, for instance, black colorant error would be considered more important than white colorant error, and therefore, larger weight may be assigned to the black colorant error.

This section concludes with one notice of accuracy of "color difference" formula defined in (2); the adopted CIE 1976 (L^*, a^*, b^*)-space may not be perfect. In color science, it is still important to characterize the nature of human color perception.

VIII. Concluding Remarks and Future Directions

In Section IV, we have demonstrated the strength of a knowledge-embedded neuro-fuzzy model, CANFIS. By constructing MF's in the color attribute space, this neuro-fuzzy approach allows us to express and realize meaningful representations of colorists' knowledge. This concept was further incorporated into "manufacturing intelligence," highlighted in Section V, a unique blend of principal components of soft computing where a GA with a KB plays a leading role in pursuit of predictions, linking an FS and NNs; they function complementarily as an evolutionary system. The resultant "manufacturing intelligence" system has a mechanism for checking predicted perceptual color difference in conjunction with an embedded color simulator $\mathrm{NN}_{\mathrm{Lab}}$ by simulating the manufacturing cycle of color paint. Therefore, the system realized a higher degree of prediction precision, improving the results of other individual approaches, although its disadvantage is that it was fairly time-consuming to construct the entire architecture (shown in Fig. 10) using soft-computing function approximators.

Our immediate future work includes the following:

- employ CANFIS/NN's to compensate for the conventional Kubelka-Munk-theory-based system, as illustrated in Fig. 17;
- improve CANFIS and NN performances by using advanced nonlinear least squares techniques, i.e., a direct dogleg trust-region algorithm [14], [15] for a small-scale problem, or an iterative Krylov-dogleg algorithm [16], [17] for a large-scale problem;
- develop systematic and faster implementations of the computational intelligence for further improvements.

Acknowledgment

The authors are deeply grateful to anonymous reviewers for detailed comments on our early manuscript. Also, we would like to thank T. Hirayama (Kansai Paint Co., Ltd.), K. Nishio (Sony Corp.), and Dr. M. A. Lee for their valuable suggestions.

References

[1] H. Bersini and G. Seront, "In search of a good evolution-optimization crossover," *Parallel Problem Solving from Nature 2*, vol. 2, pp. 479–488, 1992.
[2] J. M. Bishop, M. J. Bushnell, and S. Westland, "Application of neural networks to computer recipe prediction," *Color Res. Applicat.*, vol. 16, no. 1, pp. 3–9, 1991.
[3] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*. Reading, MA: Addison-Wesley, 1989.
[4] R. W. G. Hunt, *Measuring Color*, 2nd ed. London, U.K.: Ellis Horwood, 1991.
[5] J.-S. R. Jang, "ANFIS: Adaptive-network-based fuzzy inference systems," *IEEE Trans. Syst., Man, Cybern.*, vol. 23, pp. 665–685, May 1993.
[6] T. Kido, H. Kitano, and M. Nakanishi, "A hybrid search for genetic algorithms: Combining genetic algorithms, tabu search, and simulated annealing," in *Proc. 5th Int. Conf. Genetic Algorithms*, July 1993, p. 640.
[7] D. L. MacAdam, *Color Measurement: Theme and Variations*. Berlin, Germany: Springer-Verlag, 1981.
[8] N. Mansour and G. C. Fox, "A hybrid genetic algorithm for task allocation in multicomputers," in *Proc. 4th Int. Conf. Genetic Algorithms*, July 1991, pp. 466–473.
[9] E. Mizutani, J.-S. R. Jang, K. Nishio, H. Takagi, and D. M. Auslander, "Coactive neural networks with adjustable fuzzy membership functions and their applications," in *Proc. Int. Conf. Fuzzy Logic Neural Networks*, Iizuka, Japan, Aug. 1994, pp. 581–582.
[10] E. Mizutani, H. Takagi, and D. A. Auslander, "A cooperative system of neural networks and genetic algorithm with fuzzy population generator for computer color recipe prediction," in *Proc. Int. Conf. Fuzzy Logic Neural Networks*, Iizuka, Japan, Aug. 1994, pp. 83–84.
[11] E. Mizutani, J.-S. R. Jang, K. Nishio, H. Takagi, and D. M. Auslander, "Coactive neuro-fuzzy modelings for color recipe prediction," in *Proc. IEEE Int. Conf. Neural Networks*, Nov. 1995, pp. 2252–2257.
[12] E. Mizutani and J.-S. R. Jang, "Coactive neural fuzzy modelings," in *Proc. IEEE Int. Conf. Neural Networks*, Nov. 1995, pp. 760–765.
[13] E. Mizutani, H. Takagi, and D. M. Auslander, "A cooperative system based on soft computing methods to realize higher precision of computer color recipe prediction," in *Proc. Applications Science Artificial Neural Networks*, Apr. 1995, pp. 303–314.
[14] E. Mizutani, "Powell's dogleg trust-region steps with the quasi-Newton Hessian for neural nonlinear least-squares learning," in *Proc. IEEE Int. Conf. Neural Networks*, Washington, DC, July 1999.
[15] ——, "Computing Powell's dogleg steps for solving adaptive networks nonlinear least-squares problems," *Proc. 8th Int. Fuzzy Systems Assoc. World Congr.*, vol. 2, pp. 959–963, Aug. 1999.

[16] E. Mizutani and J. W. Demmel, "On iterative Krylov-dogleg trust-region steps for solving neural networks nonlinear least squares problems," in *Proc. Advances Neural Information Processing Systems (NIPS 2000)*, Dec. 2000, to be published.

[17] ——, "On generalized dogleg trust-region steps using the Krylov subspace for solving neural networks nonlinear least squares problems," Dept. Comput. Sci., Univ. Calif., Berkeley, Tech. Rep., 2000.

[18] J. A. Nelder and R. Mead, "A simplex method for function minimization," *Comput. J.*, vol. 7, pp. 308–313, 1965.

[19] E. Polak, *Computational Methods in Optimization*. New York: Academic, 1971.

[20] J. Spehl, M. Wolker, and J. Pelzl, "Application of backpropagation nets for color recipe prediction as a nonlinear approximation problem," in *Proc. Int. Conf. Neural Networks*, June 1994, pp. 3336–3341.

[21] M. Srinivas and L. M. Patnaik, "Genetic algorithms: A survey," in *IEEE Comput.*, June 1994, pp. 17–26.

[22] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its application to modeling and control," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-15, pp. 116–132, 1985.

[23] W. D. Wright, *The Measurement of Color*, 4th ed. London, U.K.: Hilger, 1969.

[24] G. Wyszecki and W. S. Stiles, *Color Science: Concepts and Methods, Quantitative Data and Formulae*, 2nd ed. New York: Wiley, 1982.

Eiji Mizutani (S'00) was born in Evanston, IL, in 1965. He received the B.S. degree in electrical engineering from Ritsumeikan University, Japan, in 1989, and the M.S. degrees in mechanical engineering in 1994 and in industrial engineering and operations research in 1999, both from the University of California, Berkeley.

He was an Artificial Intelligence Research Engineer with Kansai Paint Co., Ltd., Osaka, Japan, from 1989 to 1996, and has been a Consultant with Sony Electronics, Inc., San Jose, CA, since 1998. His current research interests are in the fields of operations research and numerical linear algebra. He is a coauthor (with J.-S. R. Jang and C.-T. Sun) of *Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence* (Englewood Cliffs, NJ: Prentice-Hall, 1997).

Hideyuki Takagi (M'91) was born in Japan in 1956. He received the B.S. and M.S. degrees from Kyushu Institute of Design, Japan, in 1979 and 1981, and the Dr.E. degree in 1991 from Toyohashi University of Technology, Japan, in 1991.

He was with Central Research Labs, Matsushita Electric Industrial Co., Ltd., from 1981 to 1995. He was a Visiting Researcher with the Computer Science Division, University of California, Berkeley, from 1991 to 1993. He has been with Kyushu Institute of Design, Fukuoka, Japan, as an Associate Professor since April 1995. His research interests include NN, FL, GA, and other soft computing technologies; he is especially interested in fusing these techniques and interactive evolutionary computation and human capability such as interactive evolutionary computation.

He is a member of the Institute of the Electronics, Information, and Communication Engineers, the Acoustic Society of Japan, Japan Society of Artificial Intelligence, and the Japan Society for Fuzzy Theory and Systems.

David M. Auslander received the B.S.M.E. degree from Cooper Union, New York, NY, and the S.M. and Sc.D. degrees from the Massachusetts Institute of Technology, Cambridge, all in mechanical engineering.

He is Professor of mechanical engineering and Associate Dean for Student Affairs and Research at the University of California, Berkeley. He has interests in dynamic systems and control. His research and teaching interests include mechatronics and real time software, bioengineering, and mechanical control. Current projects in these areas are design methodology for real-time control software for mechanical systems, control of regenerative life-support systems, satellite control and engineering curriculum development. He consults in industrial servo control systems and other control and computer applications. He is Cofounder and Senior Technical Consultant for Berkeley Process Control, Inc., Berkeley, a company specializing in industrial machine control.

Dr. Auslander was twice awarded the Levy Medal from the Franklin Institute, Education Awards from the Dynamic Systems and Control Division of ASME and the American Automatic Control Council. He is a Fellow of ASME. He has a longstanding association with the Dynamic Systems and Control Division of ASME.

Jyh-Shing Roger Jang (S'89–M'93) was born in Taiwan, R.O.C., in 1962. He received the B.S. degree in electrical engineering from National Taiwan University, Taipei, in 1984 and the Ph.D. degree in Electrical Engineering and Computer Science from the University of California, Berkeley, in 1992.

He was with The MathWorks, Inc., Natick, MA, from 1993 to 1995, and coauthored the Fuzzy Logic Toolbox. Since 1995, he has been with the Department of Computer Science, National Tsing Hua University, Hsinchu, Taiwan. He has published two books: *Neuro-Fuzzy and Soft Computing* (Englewood Cliffs, NJ: Prentice-Hall, 1997) and *MATLAB Programming and Applications* (China: CWeb, 2000, in Chinese). His research interests include melody/music recognition, biometric identification, web technologies, neural networks, and fuzzy logic.