"DYE SELECTION AND COST OPTIMIZATION IN TEXTILE INDUSTRY"

PROJECT REPORT
SUBMITTED BY:

Binu V. Thayamkery
Bino Mathew
P. Vangiliappan

Under the Guidance of:

Prof. P. Shanmugam B.E., M.Sc.,(Engg), M.S.(Hawaii), SMIEEE, MISTE;

Submitted in partial fulfilment of the requirements
for the award of the Degree of
Bachelor of Engineering
*in **Computer Science and Engineering** of the*
***Bharathiar University**, Coimbatore.*

Department of Computer Science and Engineering
Kumaraguru College of Technology
Coimbatore - 641 006

Kumaraguru College of Technology

Coimbatore -641 006.

Department of Computer Science and Engineering

Certificate

This is to Certify that the Report entitled

"DYE SELECTION AND COST OPTIMIZATION IN TEXTILE DYEING INDUSTRY"

has been submitted by

Mr. Binu V. Thayamkery, Bino Mathew, P. Vangiliappan

in partial fulfilment of the requirements for the award of Degree of Bachelor of Engineering in the Computer Science and Engineering Branch of the Bharathiar University, Coimbatore - 641 046 during the academic year 1995-'96.

______________________ ______________________

(Guide) (Head of Department)

Certified that the Candidate was Examined by us in the Project Work Viva-Voce Examination held on ____________________ and the University Register Number is ____________________

______________________ ______________________

(External Examiner)

TABLE OF CONTENTS

ACKNOWLEDGMENTS

We are eternally indebted to our project guide and Head of the department of Computer Science and Engineering, **Prof.P.Shanmugam,** M.Sc(Engg), M.S.(Hawaii), SMIEEE, MISTE, for introducing us to textile field .We also thank him for the constant support and helpful suggestions he has given us through out the project.

We would like to express our gratitude to our beloved principal, **Dr.S.Subramanian** for his kind patronage.

We would like to thank **Mr.Hari Somasundaram** M.S.(McGill), for his timely allusions and cooperation, especially in the field of textile dyeing.

Finally, we would like to thank all the members of the faculty of the department of Computer Science and Engineering and our great friends for the strong support and inspiration through out the development of the project..

SYNOPSIS

Textile industry is one of the fast growing industry in India . Textile dyeing being one of its major component catches the importance every time . The quality of the textile depends on its texture as well as its dyed color. In India the textile dyeing is still in a premature stage and is depending on the primitive methods of dyeing . In the past decade computers have emerged as a powerful solution for many of the industrial applications . It proved to be useful in textile design and weaving .

The conventional trial and error mixing of dyes to find the recipe involves a lot of wastage of dyes and labor. This motivated us develop a software which can give the recipe for given input color. The system takes visually matched color as the input and process it to find the cost optimized recipe.

The psychology behind the development of this system is to help the dye houses to get an idea of the recipe in an allowed tolerance limit with the use of the system.

The system also provides facility to store new dyes coming into the market in the database so that it can also be considered while processing the color. The user can maintain a record of all the processed color which he can access it later with out processing the color again.

1.0 INTRODUCTION

1.1 WHAT IS COLOR ?

The Color is something which makes the object more appealing ,attractive and gives the pleasure of observation. So it can be defined as a sensation arising from the stimulation of retina of the eye. Thus it is a psychophysical response ,that is psychological response to a physical stimulus. Color may have different meaning to different people. To chemist it may be a chemical compound, a dye or a pigment, to physicist it is a scattering and absorption of light or reflectance spectra of the object, to physiologist it is a complex process of brain of interpreting the nerve signal. To artist and others ,it is the means to create sensation in the mind of the observer .For example Red and Yellow colors create the sensation of warmness. Green and Blue are associated feelings of coolness. Color harmony in all paints

,curtains and furnitures makes the room atmosphere cheerful and gives comfort.

1.2 PERCEPTION OF COLOR

Perception of color involves a series of events which are interdisciplinary in nature. Perception of color includes source of light , object that is illuminated and eye and brain that perceive the color. A source of light is characterized by the energy radiated at different wavelengths i.e. by its spectral power distribution. The modification of the incident radiations depends on the nature of the colorants in the object and so it is related to chemistry of dyes. The radiations reflected by the object are received by the eye. The radiant energy is absorbed by the photosensitive pigments in the retina. and this gives rise to nerve impulse which is transmitted to brain. The brain interprets the signal depending on the informations stored in the memory and learning process. The interpretation of color by eye and brain is a part of physiology.

1.2.1 SOURCE OF LIGHT

Light is a form of energy and it propagates in the form of electromagnetic waves. Wavelength varies from fraction of nanometer to kilometers, The characteristics of electromagnetic waves change with wavelengths. Only a small part of the electromagnetic spectrum produces the sensation of vision. This part of the spectrum is called as visible radiations and the wave lengths varies from about 380 nm to 750 nm. The hue of visible radiations changes with wavelengths.

1.2.2 OBJECT

The illuminating radiations are modified by physical processes such as transmission, reflection , absorption and scattering. The relative proportions of these processes depend on the characteristics of material. The material appears blue or red as it absorbs all spectral components of radiations except blue or red. The absorbed energy is converted into heat and is lost. In addition to absorption ,the light may also be scattered . In the scattering process light travels in many directions other than incident direction. When sufficient scattering occurs , it is said to be diffuse reflection or transmission. Such objects are translucent e.g. opal glass or plastic. The scattering of the light depends on the difference in refractive indices of the colorant and the surroundings. The appearance of the object is characterized by its spectral reflectance curve . The location of peak in spectral reflectance curve determines the hue of the object color. The object exhibits absorption peak at definite wavelength depending upon its color. The spectral reflectance curves of achromatic objects (i.e.white gray, black do not exhibits any peak.

1.2.3 OBSERVER

The common detector of light and color is eye, nerve system and the brain. The eye focuses the image of the object on retina. The photosensitive detectors are called rods and cones from their shape. The rods detect the light but have no ability to specify the color . The color is detected by the cones. There are three types of cone receptors in the retina. They are sensitive to light in different wavelength range.

Therefore they are known as blue sensitive cones and red sensitive cones or green sensitive cones. The ultimate sensation of color depends on the degree of stimulation of these three color receptors. If all these cones are equally stimulated it gives the sensation of gray to white depending on degree of stimulation . If blue and red receptors are simultaneously excited , the sensation of purple is created. this mechanism of perception of color is based on the additive mixing.

1.3COLOR MIXING LAWS

The series of experiments to produce gamut of colors by adding colored lights were carried out in good old days by Newton. But the fundamental difference in mixing light and pigments/dyes was first recognized by Helmholtz. The former is now known as the additive color mixing and the later as subtractive color mixing. In industrial applications , like textile ,paint , paper and plastic the desired colors are produced by mixing two or more colorants. Therefore it is significant importance to know the resultant color obtained by mixing various colorants in different proportions . The laws governing the color mixture of colorants are known as color mixing laws. In some cases these laws are quite complex but at least gives a qualitative idea of the resultant color.

1.3.1 *ADDITIVE COLOR MIXING*

Additive color mixing occurs when two or more lights are added by focusing them on a white screen. In additive color mixing red , blue and green are used as primaries to produce various colors. the primary colors selected are independent in the sense that mixture of two will not produce third primary color. In case of primaries selected for additive mixing we have,

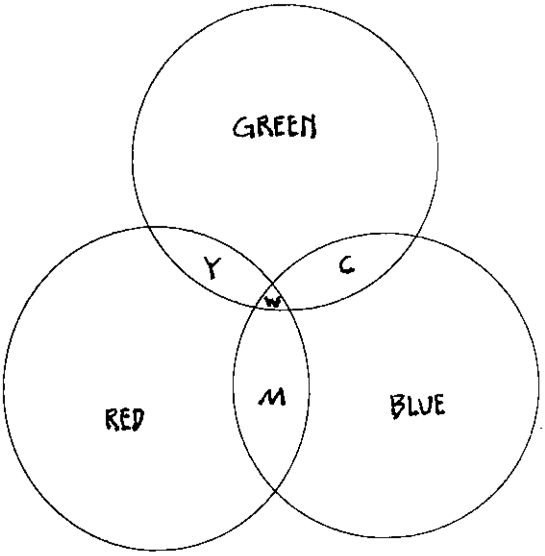
Red + Green ---> Yellow

Green + Blue ---> Cyan (blue-green)

Blue + Red --->Purple (magenta)

Red + Green + Blue ---> White

The colors resulting from the additive color mixing is shown in the figure given below:



Y -Yellow M- Magenta W- White C-Cyan

1.3.2 SIMPLE SUBTRACTIVE MIXING

Subtractive mixing of colors occurs when one or more spectral components are removed from the incident light . The removal of part of incident energy can occur by the process of absorption and scattering. In absorption process, The light energy is converted into heat and/or absorbed by atoms of the material. When the subtraction is made by absorption only it is said to be simple subtractive mixing .

When the light is removed by scattering and absorption . it is said to be a complex subtractive mixing.

Simple Subtractive Mixing:

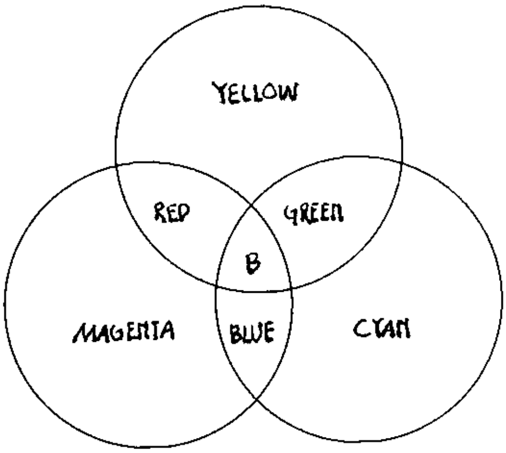
The colors of transparent materials like solutions and glass occur due to simple subtraction of light energy by absorption . The instruments used for color matching by subtractive color mixing employ light filter which has selective absorption . The most commonly used primaries in subtractive mixing are yellow, magenta and cyan. The subtractive primaries are obtained by removing blue green and red light from white respectively

White - Blue ---> Yellow

White - Green ---> Magenta

White - Red ---> Cyan

The primaries of the additive and subtractive color mixing can be represented by the hue circle as shown in the figure given below. The results of the additive and subtractive primary mixing can be best remembered by realizing that the pairs of additive primaries generate the additive primaries. The addition of the subtractive primaries magenta and yellow gives red , the additive primary. The mixture of additive primaries red and green gives subtractive primary yellow.



Complex Subtractive Mixing :

When light energy is removed by absorption and scattering , the most complex type of color mixing occurs. Since complex subtractive color mixing involves the simultaneous absorption and scattering of light , the corresponding laws of mixing should include two parameters accounting for these processes. The colors we see in industries like Textiles , paint etc. are result of complex subtractive color mixing.

1.4 METAMERISM

The spectral reflectance curve is a characteristic of the object. Eye does not have separate receptor for each wavelength and the perception of color is a cumulative effect of nerve signals interpreted by brain received from three color receptors. Under this condition , two objects having different spectral reflectance curve appears to have the identical color under a given source. The colors of these objects may not match when seen under another source. This phenomenon is called as the Metamerism. The metamerism is referred to a pair of objects . Two objects having exactly identical spectral reflectance curves will always appear to have similar color under any source. They are called as non-metameric and the matching is called as invariant match. The metamerism is a very important phenomenon to be considered while matching the color in any industrial product.

2.0 COLOR ORDER SYSTEMS

It is a common practice to describe the color in terms of red , green or greenish yellow etc. this is not sufficient to communicate complete details about the color , e.g. greenish yellow does not carry any sense of ‘ how much green? ‘ ‘how much yellow? ‘ Therefore some methods must be worked out to precisely describe the color such that it can be correlated to with the sensation of color . This will help to discuss about the color across a distance , over a span of time and also in absence of right physical sample . The methods devised to quantitatively describe the color are called as Color Order Systems. Number of color order have been proposed. Each of these system is used for one or other purpose. We classify the color order systems in two groups as follows :

1. Color order systems which are based on collection of samples. Munsell System, Ostwald System, OSA - UCS System and ISCC - NBS System and many other come under this class. Each of the systems are represented with its color atlas. The atlas contains various

color chips, which are arranged in definite configuration and are identified by definite name and numbers.

2. Color order systems which describes the color by mathematical numbers. The descriptions of color in these systems are based on the measurements of spectral reflectance of the sample. Hunter L, a ,b, CIE system and others obtained by its transformation come under this class. These color order systems are used in instrumental color measurements.

2.1 *DESCRIPTION OF COLOR*

The ' desert island ' experiment described by Judd (Judd and Wiszecki 1975,Billmeyer and Saltzman 1981) provides a very good illustration about the arrangement of colors in color space. The experiment is based on natural and logical approach of a person to the gamut of colors available to him. The experiment gives the natural method of describing color by three parameters HUE, LIGHTNESS or VALUE and CHROMA or SATURATION . These attributes form the basis for the color atlas. They are commonly known as psychophysical color parameters. In color atlas collection of several hundred chips are classified and arranged in terms of easily recognizable attribute. The function of color atlas is to assist the color user to describe , select and match the colors. Out of all these systems Munsell Color Order System is most widely used for the description of colors of industrial products.

2.2 *MUNSELL COLOR SYSTEM*

A color order system describing colors psychophysical dimension hue, saturation and lightness was proposed by artist Albert S. Munsell (1929,1963,1969) . The system is based on the collection of samples. The Munsell samples are prepared to represent equal interval of visual perception between adjacent samples. Munsell hue qualify colors by name . Munsell hue consists of five principle hue red, yellow , green, blue, purple and five intermediate hues as yellow - red , green-yellow, blue-green, purple-blue and red-purple. Each of ten hues is divided into ten steps or any fractional quantities as 2.5,5,7.5,10.0 .Typical examples of representation are 5R, 5YR, 5Y, 5GY etc.In the Munsell hue circle the center is a gray pole, varying in value. The likeness dimension is called as Munsell value . this Munsell value varies black at the bottom with ‘zero’ value to white at the top with value ‘ten’. From gray pole , the colors increase in saturation as the radius of the hue circle increases, so that the most saturated colors occur at the periphery. This saturation scheme is called as Munsell Chroma. Munsell Chroma has its value 0 for neutral and increases in steps of two. It increases up to 14 for non-fluorescent red and yellow paints. In Munsell system, the color is qualified by Munsell Hue ,Munsell value and Munsell Chroma.

2.3 *COLOR ATLAS SYSTEMS*

Munsell color systems is widely accepted to specify the color of an object . In addition to the Munsell system , few more systems , based on the collection of samples are also proposed in the literature .

2.3.1 *OSTWALD SYSTEM*

The first system based on modern understanding of spectrophotometric properties of dyes and pigments was proposed by Ostwald (Ostwald, 1931,1969 , Jacobson 1948). The colors in Ostwald system are produced by mixing the colors of high Chroma with black and white colorants. The color atlas based on Ostwald system is not much used in practice. The color chips in the atlas are prepared on transparent cellulose acetate base. Thus one side of the chip is glossy and other is matte , and this can be considered as one of the advantage of this system.

2.3.2 *THE OSA-UCS SYSTEM*

The task of developing the color catalogue exhibiting uniform visual spacing was undertaken by Optical Society Of America. With an intense effort for 30 years , a Uniform Color Scale Committee prepared 558 color chips spaced according to redness-greenness , yellowness-blueness and lightness. The system provides the chips illustrating high Chroma colors.

2.3.3 *THE NATURAL COLOR SYSTEM*

The 'Natural Color System' term was used to describe a color system based on opponent colors. (Hard 1968,1970). He postulated that the surface color could be described with six elementary attributes whiteness,blackness,redness,greenness,yellowness and blueness. This system , partly based on steps of equal visual perception has been developed in Sweden. This system is published by Swedish Color Center in the form of color atlas containing 1300 color samples.

2.3.4 *CHROMA COSMOS 5000*

The major drawback of the above described systems is that they contain comparatively small numbers of colors. The Japanese Chroma Cosmos extended the color chips from 1500 to 5000. These chips are arranged in pages of constant Chroma contrary to Munsell book which contains the pages of constant hues.

2.3.5 *THE ICI COLOR ATLAS*

It is said that the color atlas provides only very small fraction of colors that textile dye house is producing . The sewing thread firm of J&P Coats of Paisley produces not less than 10,000 color standards. To cater the needs of industries Imperial Chemical Industries (ICI) has produced a color atlas containing over 27,000 different colors . In this case 1379 brightest colors are printed onto 12 cards. Each of these cards can then be overlaid by one of the twenty neutral gray filters . Each combination is identified by simple letter/number code.

2.3.6 *CIE SYSTEM*

The perception of color includes source , object and the observer . In 1931 CIE (Commission International de l'Eclairage or International Commission Of illumination) introduced a system to quantify the colors in terms of mathematical numbers. The system is of great use in instrumental color measurement and computer color matching technique. To evolve a system, it is necessary to describe and quantify standard sources and observer.

[AYMW1]

Colour Assessment in Textiles

3.0 COLOR ASSESSMENT IN TEXTILES

Color of textile product is an important parameter to be presented to the customers . Therefore , the coloration of textiles , its assessment and matching are important phases in production of textile materials. As per prevailing practice in many industries , the recipe is determined by Color technologists by conventional method and its validity is tested by dyeing and visually examining the samples. The recipe so decided should be economical and least metameric. For this purpose color technologists should have detailed knowledge about the characteristics of wide range of dyes, for example knowledge of fastness , compatibility of dyes , cost etc. Further shading is required at production department as every batch is likely to be subjected to some fluctuations in production parameters . This is a difficult task. Colorists should be able to correctly sense what is required to eliminate to achieve the exact match. Large number of dyes and substrates with widely differing properties are coming in the market and it is hardly possible for new generation color technologists to solely rely on eye and treasure of memory collected through experience. Visual expertize should be supplemented by computer aided instruments . But even today considerable proportion of textile units are following age old technique of visual color assessment and empirical colorant formulation . The technique of instrumental color measurement has provided a powerful tool in the hands of color technologists for color assessment and matching in textile industries.

3.1 *VARIABLES IN VISUAL COLOR ASSESSMENTS*

In Textile Industries the word color matching implies the duplication of color in a dye house using appropriate color recipe . Therefore

Color of the sample = Color of the standard

In visual color matching,

Color sensation produced by a sample = Color sensation produced by a standard.

In color co ordinates the color is described by the hue , saturation and brightness.

Thus , Color A= (H,S,B)

The resultant visual sensation may be related to many variables in observers surrounding and response of his eye and the brain. These three variables in visual assessment are related to

$$(H,S,B)=f(E,R,r,g,b,M,S,A,O,T,u,v,w)$$

Here product of E and R represents spectral composition of light striking the eye. r,g,b are particular observer spectral response of color sensitive pigment in the eye,M is memory , S is the nature of surrounding , A is the state of adaptation of the observer , O is the nature of surrounding object , T is the observer attitude and u,v,w are other unspecified variables. Thus the result of visual color assessment are significantly affected by several parameters which are difficult to control. In practice the textile color matchers manipulate dyes and their concentrations to match the color of the samples with the standard. In this case,

$$(H,S,B)_{\text{sample}} = (H,S,B)_{\text{standard}}$$

In visual color assessment eye , brain and many other functional variables are involved. But the visual color assessment can be correlated with instrumental color measurement by controlling or by putting some restrictions on functional variables.

3.2 *VISUAL COLOR ASSESSMENT*

The visual examination of industrial products can be made by keeping the production sample and customer standard side by side under standardized lighting conditions. The human eye and the brain have incredible sensitivity to detect even a small color difference . Although eye is marvelously sensitive and discriminating sensor it cannot make the measurement of magnitude of variations in color of the sample as compared to the standard. The limitations of the human eye can be compensated by using more than one standard which may be differing in known amount of nature and magnitude in color as compared to the target standard. Further physical conditions in viewing booth in industry must be maintained so as to have precise and reproducible visual assessment of the product. To achieve this objective it is essential to know the limitations and the impact of various prameters involved in visual assessment .

The source , illuminating - viewing conditions and surroundings are parameters which will have significant on the result of visual assessment. A 200 foot candle of light is recommended for discrimination of light and dark shades. A source for visual examination may be natural day light or simulated day light. Natural day light received from northern sky is preferred for color assessment. Unfortunately spectral quality of daylight varies with the time of day, weather , direction of view , time of the year readily adjust to variations in spectral nature of light and that we are often unaware about its significant impact on visual evaluation particularly in color matching. Therefore the recent trend in industrial testing is to use artificial light source. The extended light source is recommended rather than point source to dilute the effect of geometric attribute like gloss and texture of samples.

The color perceived by the observer is directly related to the geometric conditions of the observation .i.e., the direction of illumination and viewing . For correct assessment of color , the sample must be illuminated about the perpendicular and viewed at 45 Degrees. The reciprocal of this arrangement can also give the same results.

The assessment of color will be made by the observer . The problem related to variability in observer is the most difficult to come to commonly accepted solutions . Light sources and the geometry of observation , the standardization in observer is yet not possible . It may be noted that there is a large spread of observer characteristics within the range identified as normal observer. Therefore the variation in judgment from a team of observer having normal color vision is not a new experience in color industries. Therefore there should be a common agreement between buyer and seller about the team of observers. The observer difference must also be recognized and taken into account while accepting or rejecting the end product.

4.0 ALGORITHMIC APPROACH

4.1 *PALETTE ALGORITHM*

We have implemented a palette which will support all the combination of the color i.e. 16 million colors. However our system requirement allows, us only to have one color only at a time to be displayed on the screen. The Windows Palette which is also linked with the palette helps the user to select color in a very easy way by adjusting the different parameters of the color such as R,G,B components, hue, saturation and brightness.

In generating color each pixel is represented in 24 bit representation. The basic components of the color, i.e. Red,Green and Blue will take a byte each. This will enable us to have 256 different combinations in each R,G or B. So ultimately combining all these into a 24 bit representation, all the possible 16 million colors can be generated.

Algorithm:

Define color as 24 bit.

```
Color = R

Color = G << 8 //shifting 8 bits

Color =  B <<16 // shifting 16 bits.

Color = R+G+B


// color extraction.

Define R as long

Define G as  long

Define B as long

R = color AND  255 // red extracts

G = (color AND 65280) 1256 // blue extracts

B = (color  AND  16711680)/65536 // green extracts
```

4.2 *MATHEMATICAL MODEL FORMULATION.*

The system requirements forces us to formulate our problem into a mathematical model which will give emphasis on the combinational recipe preparation in an optimized way.

Each dye can be stored as a color by having its basic color components. Our task is to find out the recipe to generate the input color. The color of a particular dye is represented as a vector (R,G,B)'. Then the problem is modelled into a linear programming problem.

To get the combinational mix in a cost optimized way,

minimize cost = $c1\ k1 + c2\ k2 + c3\ k3 + \ldots + c_n\ k_n$

subject to the constraints

$r1\ k1 + r2\ k2 + r3\ k3 + \ldots + r_n\ k_n = R$

$g1\ k1 + g2\ k2 + g3\ k3 + \ldots + g_n\ k_n = G$

$b1\ k1 + b2\ k2 + b3\ k3 + \ldots + b_n\ k_n = B$

$r_n, g_n, b_n > = 0; \quad 0 < = n < = N$

where $c1, c2, c3 \ldots c_n$ represents the color of the dyes

$k1, k2, k3 \ldots k_n$ represents the quantity.

$r1, r2, r3 \ldots r_n$. represents red component

$g1, g2, g3 \ldots g_n$ represents green component

$b1, b2, b3 \ldots b_n$ represents blue component

The color which are represented as a vector i.e. (R,G,B)' we have formulated the linear programming problems keeping in mind that the color can be vector additive.

4.3 *SIMPLEX ALGORITHM*

The starting point of the simplex algorithm is always a set of equations, which includes the objective function along with the equality constraints of the problem in canonical form. Thus the objective of the -simplex algorithm is to find the vector $X > = 0$ which minimizes the function $f(X)$ and satisfies the equations:

$1.x_1 + 0.x_2 + \dots 0.x_m + a1''_{m+1}\; x_{m+1} + \dots + a_1''{}_{n}x_n = b_1''$

$0.x_1 + 1.x_2 + \dots 0.x_m + a_2''{}_{,m+1}\; x_{m+1} + \dots + a_2''{}_{n}x_n = b_2''$

.

.

$0.x_1 + 0.x_2 + \dots 1.x_m + a_m''{}_{,m+1}\; x_{m+1} + \dots + a_m''{}_{n}x_n = b''_m$

$0.x_1 + 0.x_2 + \dots 0.x_{m} - f + c_{m'+1}\; x_{m+1} + \dots + \cdot c_{m'n}x_n = -f'_0$

where $a'_{ij}, c'j, b'_i$ and f'_0 are constants. Notice that $(-f)$ is treated as a basic variable in the canonical form of above equations. The basic solution which can be readily deduced from above equations is

$x_i = b_i', \; i = 1,2 \dots, m$

$f = f_0'$

$x_i = 0$; I = m+1, m+2, ... n

If this basic solution is also feasible, the values of xi, I = 1,2,.. n are non negative and hence

$b_i'' >= 0$, I = 1,2, ..., m

We can write the objective function as

$f = f_0'' + \Sigma\, c_i''\, x_i + \Sigma\, c_j'' x_j$

$= f_0''$

If at least one cj' is negative, the value of f can be reduced by making the corresponding xj > 0. In other words, the non basic variable xj, for which the cost coefficient cj' is negative, is to be made a basic variable in order to reduce the value of the objective function. At the same time, due to the pivotal operation, one of the current basic variables will become non basic and hence the values of the new basic variables are to be adjusted in order to bring the value of f less than f0'. If there are more than one cj' < 0, the index s of the non basic variable xs which is to be made basic is chosen such that

c_s'' = minimum $c_j < 0$

Although this may not lead to the greatest possible decrease in f (since it may not be possible to increase xs very far), this is intuitively at least a good rule for choosing the variable to become basic. It is the one generally used in practice because it is simple and it usually leads to fewer iterations i.e., if more than one cj'' have the same minimum value, we select one of them as cs' arbitrarily.

Having decided on the variable xs to become basic, we increase it from zero holding all other non basic variables zero, and observe the effect on the current basic variables. By the equations these are related as

$x_1=b_1''- a_{1s}''x_s$, $b_1''>=0$

$x_2=b_2''-a_{2s}''x_s$, $b_2''>=0$

.

.

.

$x_m=b_m''-a_{ms}''x_s,b_m''>=0$

$f=f_0''+c_s''x_s,c_s''<0$

Since $c_s'' < 0$ the above equation suggests that the value of xs should be made as large as possible in order to reduce the value of f as much as possible. However, in the process of increasing the value of xs, some of the variables x_i (I = 1,2,..,m) in above equations may become negative. It can be seen that if all the coefficients $a_{is}' < =$, $I = 1,2,\ldots,m$; then xz can be made infinitely large without making any $x_i < 0$, $I = 1,2,..,m$. In such case, the minimum value of f is minus infinity and the linear programming problem is said to have an unbounded solution.

On the other hand, if at least one aiz' is positive, the maximum value that xs can take without making xi negative is (b_i''/a_{is}''). If there are more than one $a_{iz}' > 0$ the largest value xs* that xs can take is given by the minimum of the ratios (b_1''/a_{is}'') for which $a_{is}'' > 0$. Thus

$$x_s^* = b_r''/a_{rs}'' = \underset{a_{is}'' > 0}{\text{minimum}} \, (b_i''/a_{is}')$$

The choice of r in the case of a tie, assuming that all bk'> 0, is arbitrary. If any b1' for which ais' > 0 is zero in the equations then xs cannot be increased by any amount. Such a solution is called a degenerate solution.

In the case of a non-degenerate basic feasible solution a new basic feasible solution can be constructed with a lower value of the objective function as follows. By substituting the value of x_s^* in the above equations we obtain

$x_s=x_s*$

$x_i=b_i''-a_{is}''x_s* >= 0$

where I= 1,2,....m and i<>r

$x_r=0$

$x_j=0$, j=m+1,m+2,....,n and j <>s.

$f=f_0''+ c_s''x_s* <= f_0''$

which can readily be seen to be a feasible solution different from the previous one. Since $a_{rs}'> 0$ in the equation a single pivot operation on the element ars' in the system of the above equation will lead to a new canonical form from which the basic feasible solution of equation can easily be deduced. Also, equation shows that this basic feasible solution corresponds to a lower objective function value compared to that of equation. This basic feasible solution can again be tested for optimality be seeing whether all ci'> 0 in the new canonical form. If the solution is not optimal, the whole procedure of moving to another basic feasible solution from the present one has to be repeated. In the simplex algorithm, this procedure is repeated in a iterative manner until the algorithm finds either (i) a class of feasible solutions for which f -->-α or (ii) an optimal basic feasible solution with all ci' > = 0, I

=1,2,.. n. Since there are only a finite number of ways to choose a set of m basic variables out of n variables, the iterative process of the simplex algorithm will termination in a finite number of cycles.

5.0 PRACTICAL IMPLEMENTATION

5.1 LANGUAGE USED

GUI(graphics user interface) is the buzz word of today and Microsoft Windows is the choice for the majority of PC users. There is an explosion of Microsoft Windows users and programmers, and more users will dramatically increase the demand for more applications and more programmers. Now a days it is a must for every PC programmer to include windows development skills in their programming repertoire.

Until recently, a developer creating a Microsoft Windows application could expect to learn 600 or more functions at a low level and write applications in either assembly language or C. It was, at best, a difficult time consuming task; at worst, it was a scenario restricted to professional or masochistic programmers.

During 1991, Microsoft introduced Visual Basic, its Windows application development alternative to C. Visual Basic has most of the poser of Turbo Pascal for Windows (the first simplified compiler for Windows), but in addition it simplifies the Windows development process.

Visual Basic's visual and event oriented approach are the keys to simplifying windows development. In a nutshell, one can develop applications graphically. Programming the interface in Visual Basic is much closer to user level than traditional programming level. Windows are created and code is added to each of them.

Visual Basic is a complete programming environment within the Windows environment and has several advantages:

It abstracts many of the 600 Windows application programmer interface (API) functions to a high level, and therefore enables a programmer to use existing Windows functionality (buttons, dialog windows, menus and so on) without using the Windows software

development toolkit (SDK). Programmers can still call low level Windows API functions through a dynamic link library (DLL).

It lets a programmer write, compile, run and debug applications within the Windows environment, without exiting to DOS. No other popular Windows compiler (including Borland C++ or Turbo Pascal for Windows) does this well. The turbo Pascal debugger, for example, despite running as a Windows application, awkwardly switches from graphics to text screen during the debugging process.

It incorporates resource development on site - that is within the Visual Basic environment. One does not need to run a resource editor or compiler to add menus, dialogs, controls and so on.

Visual Basic is genetal purpose and fast (as fast as either Turbo Pascal for Windows or C++), it's practical for commercial applications.

It has interpretive syntax checker, compiling Visual Basic applications is much more faster than both C and C++ compilers.

Dynamic date exchange (DDE) and object linking and embedding (OLE) are extensively supported.

5.2 SYSTEM IMPLEMENTATION

5.2.1 COLOR GENERATION

The system incules a pallette which can support all the 16 Million colors. To generate a particular color each pixel is represented as 24 Bit. By manipulating this 24 bit the resultant color can be generated. The system pallette is also linked with the Windows Pallette which can support colors extensively. For every input color, the color is matched visually to find out the RGB components and it is processed to give the combinational mix of dyes.

5.2.2 DATABASE ORGANIZATION

We have used the Visual Basic data linking property to create and manipulate the database. Access 1.0 which comes along with the Data Manager utilility of the Visual Basic sereves us to create the database. The database have five tables which represents the five classes of dyes that we have taken into consideration in the system . For each table the different fields are DYE NAME, RED,GREEN & BLUE Value and COST/Kg for each of the standared dyes present. For each of the standard dyes present their values are stored in the database to assist in the calculation of the recipe.

5.2.3 RECIPE PROCESSING AND OPTIMIZATION

The input color is matched visually with the help of pallette and the RGB components are found out . These values are passed to the processing routine which will in turn formulate the Linear Programming Equations,. The sSimplex Algorithm is used to solvrethe Linear Programming Problem for finding an optimized recipe. The system is implemented in such a way that the user is given the freedom to choose the combination which suites his needs. this is done by a series of calculations which recognises the LP equations to give the combiations one by one.

5.2.4 COMPUTER PROGRAMS

The computer programs could be divided into three main sets of routines.

@ The first set of routines will allow the user to match the color visually and then to process that color to obtain the recipe. Here we have used a set of routines , one for the Pallette generation and Color extraction and a second set of routines for the processing of colors to generate the recipe.The process part will take in RGB values from the pallette as the input to formulate equations from the pallette to calculate the combination of dyes. The results are then displayed in a form where the user is given the freedom to select the next combination

.

@ The second set of routine will enable the user to store the processed color data . This processed color data is stored as a file . If the user knows that the input color is already processed in a previous session, he can load that file to view the pre processed color and the combination.

@ The third set of routines is used to manipulate the database. We have created a dynamic database which grows with the addition of new dyes as and when it comes up. The stand utilities like saving the records ,deleting the records are employed here.

5.2.5 PROGRAM FILE SPECIFICATION

The different files available in the package are :

@ COL_DYE.EXE : This file is the main project file which includes all the form file and the process modules.

@ COMP.MDB : This is the ACCESS database file which ids used to store the information about the standard dyes present.

@ *.FRM Files : These are the Visual Basic form files that contain informationabout the user interface.

@ *.BAS Files : These files contain Visual Basic source code which gives the process information.

@ *.VBX Files : These are Visual Basic Dynamic Link Libraries(DLL) which are loaded directly to the RAM when Microsoft Windows is loaded. For the proper usage of the system these files should be present in the system directory.

@ *.COL Files : These files contains the information about the pre processed colors .

@ *.HLP Files : These are help files which enables user to get the help information online about the package.

5.3 *FEATURES OF THE SYSTEM*

We have facillitated a fullfledged system that can be used by the user for an effective color matching . The process routine effectively finds out all the combinations in which a color can be created from the standard dyes present.

Extensive context sensitive help has been proveided with the hypertext facility(the technique to jump from one topic to other).

Full blown realistic three dimensional menus with a great feel

We have tried to fit in a almost nil keyboard usage interface. This makes the system accessable to most lame computer users.

Statistical information such as graphs have been portraied for the intererst of the user.

5.4 PROBLEMS ENCOUNTERED

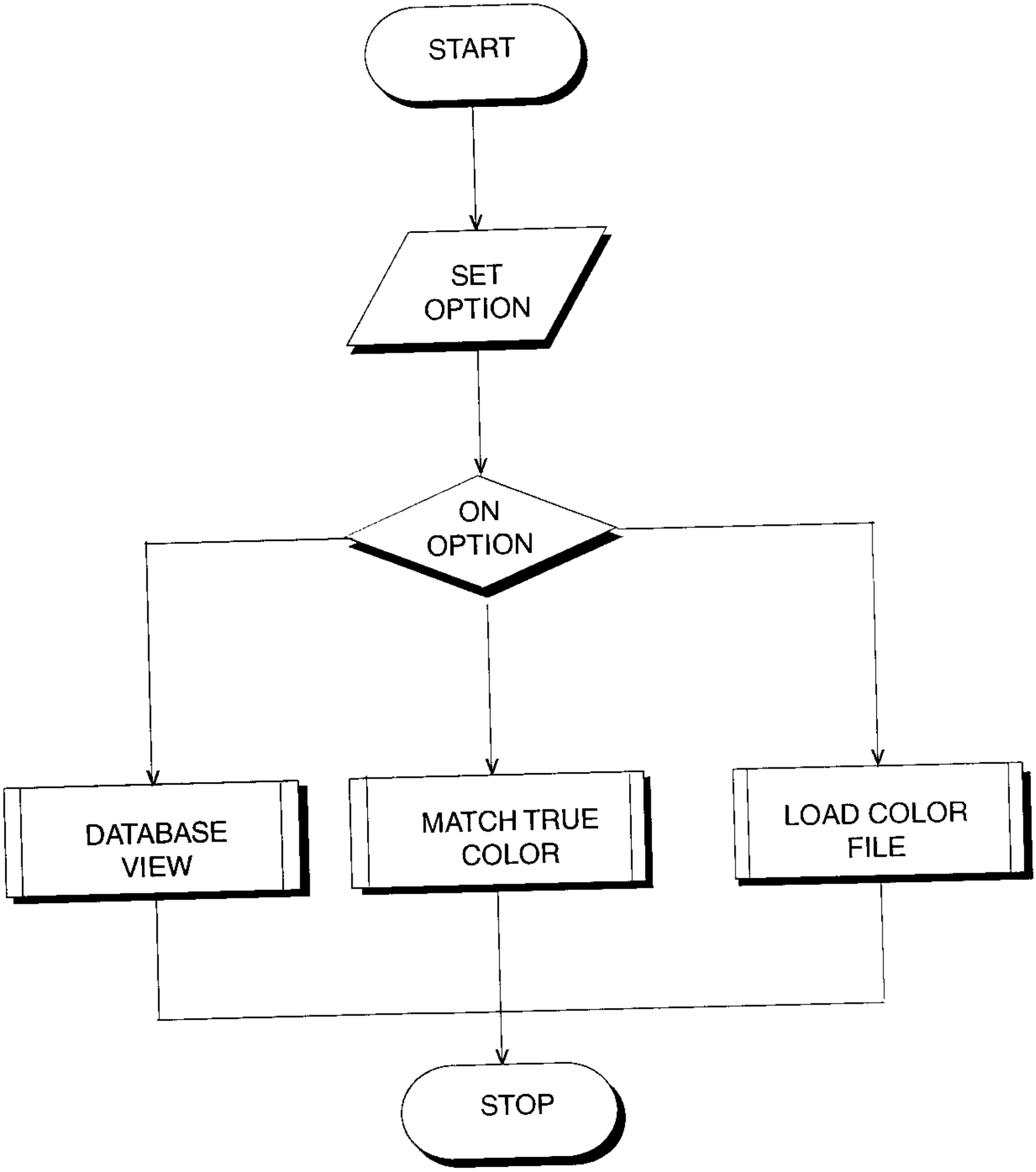
We have tried to make the system bullet proof at real time.Other than the errors that crept in due to human limitations,hopefully all of which we have debugged.We encountered minor problems which we have emphasized below:

@ To display solid colors we need to have a 2MB video RAM or a graphic accelerator which can support 16 million colors in a 640 by 480 resolution.

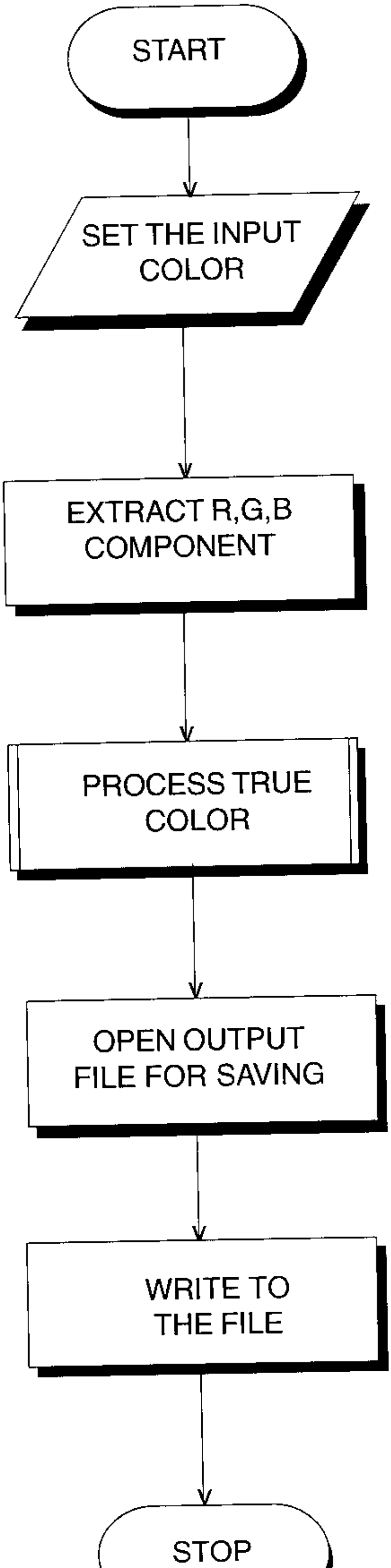
@ The system modelling as a Linear Programming problem restricted us to produce only three dye combination for a particular color.

Flow charts

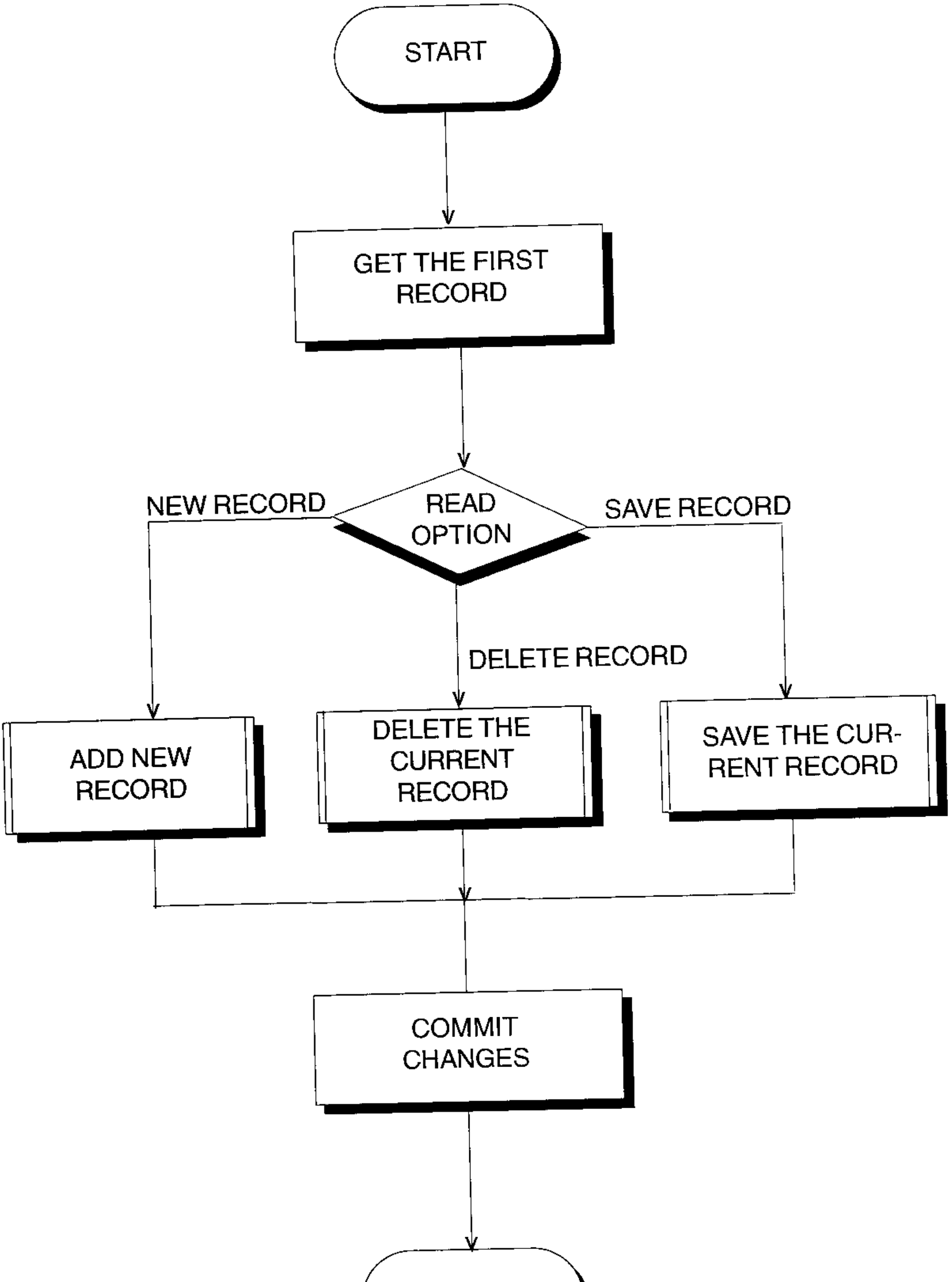
MAIN BRANCHING MODEL



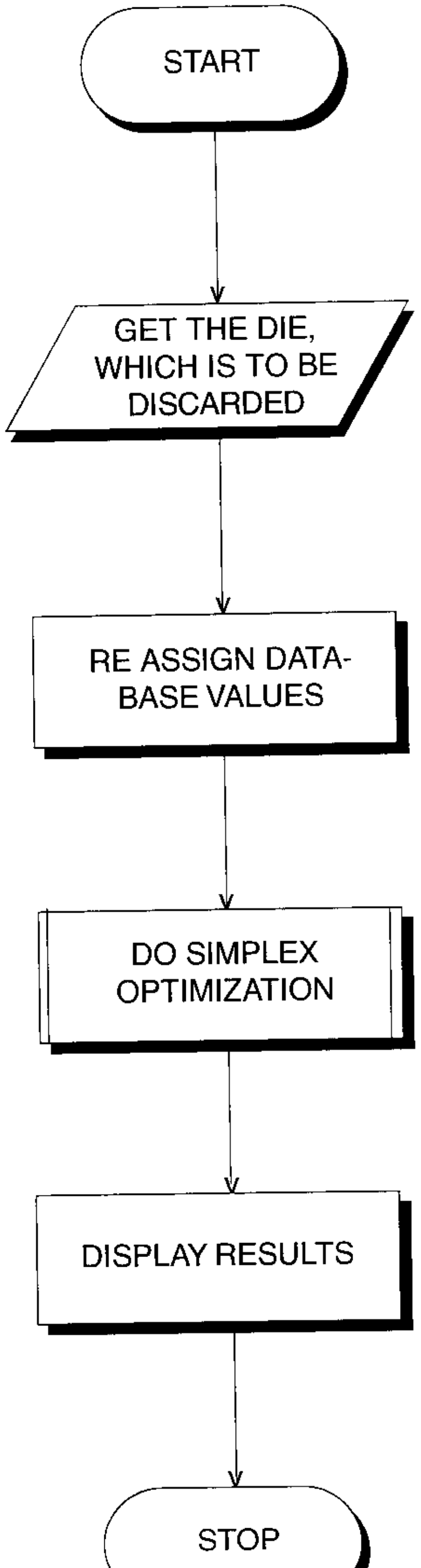
COLOR MATCHING



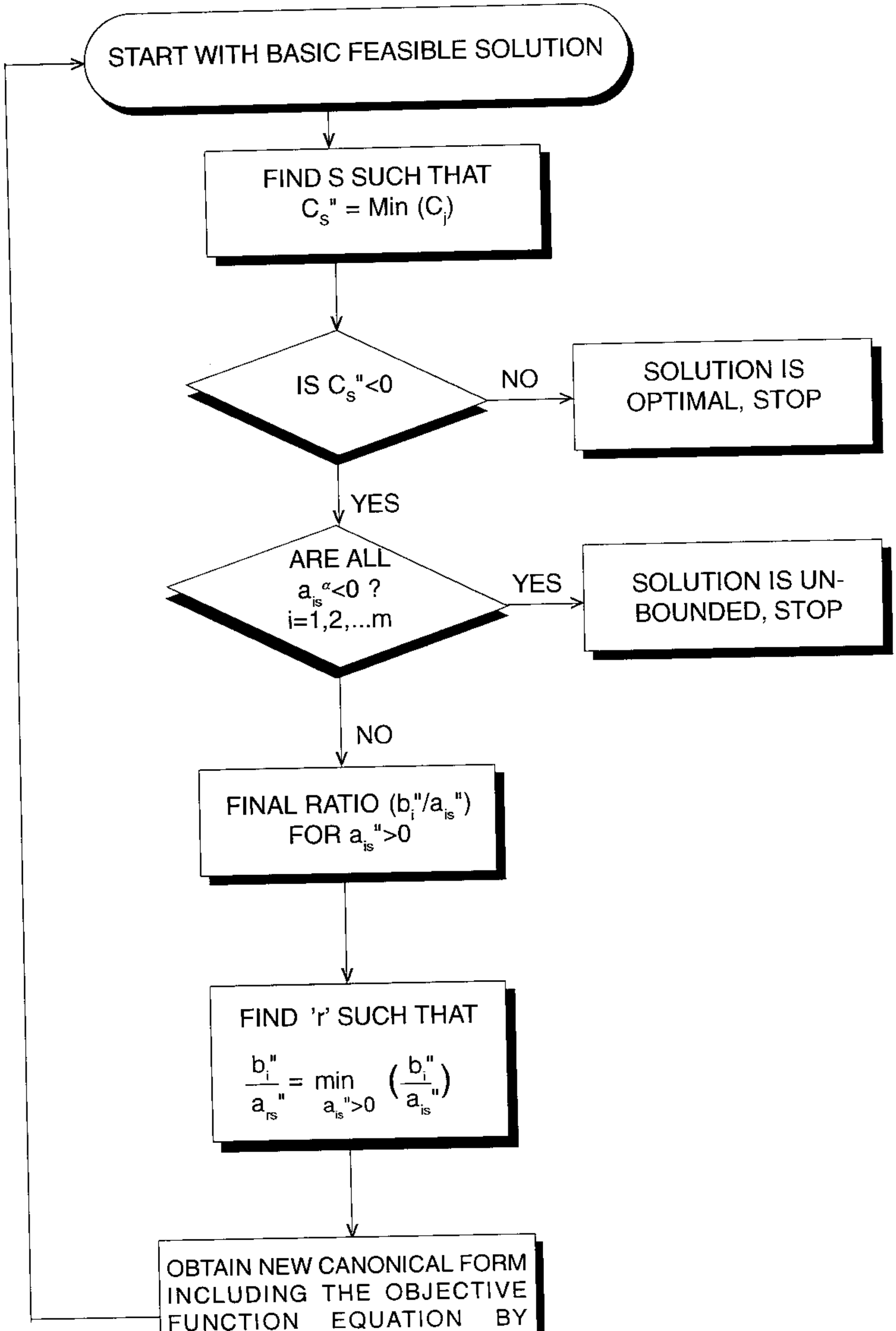
DATABASE MANIPULATION



REOPTIMIZATION MODULE


START
GET THE DIE, WHICH IS TO BE DISCARDED
RE ASSIGN DATA-BASE VALUES
DO SIMPLEX OPTIMIZATION
DISPLAY RESULTS
STOP

SIMPLEX ALGORITHM


START WITH BASIC FEASIBLE SOLUTION
FIND S SUCH THAT $C_s'' = \text{Min}\ (C_j)$
IS $C_s''<0$
NO
SOLUTION IS OPTIMAL, STOP
YES
ARE ALL $a_{is}^{\alpha}<0$? $i=1,2,\ldots m$
YES
SOLUTION IS UN- BOUNDED, STOP
NO
FINAL RATIO (b_i''/a_{is}'') FOR $a_{is}''>0$
FIND 'r' SUCH THAT $\frac{b_i''}{a_{rs}''} = \min\limits_{a_{is}''>0} \left(\frac{b_i''}{a_{is}''}\right)$
OBTAIN NEW CANONICAL FORM INCLUDING THE OBJECTIVE FUNCTION EQUATION BY

7.0 CONCLUSION AND FUTURE SCOPE

7.1 CONCLUSION

We were able to achieve the objective of this project , namely to obtain the cost optimized recipe for the textile dyeing. We found that the recipe generated by the computer falls within the tolerance level one can expect from a visual matching system.

We also came to conclusion that the Windows graphic user interface is excellent and is highly suitable for industrial applications like textile color matching. Processing colors was a lot easier under Windows than in the conventional DOS environment. Windows also facilitated us to make the software easily understandable by any lame user , no prior knowledge of computing is necessary.

7.2 SCOPE FOR FURTHER EXPANSION

The limitations of our system which was explained earlier can be overcome by fine tuning the system a little more. The mathematical model we developed is suitable for visual color matching only . The system can be modeled in such a way that a photospectrometer gives the input to the system as the spectral curves which can be manipulated for a more precise color matching.

APPENDIX -A

SYSTEM CONFIGURATION

Hardware

An IBM compatible computer with at least a 386DX processor, at least 4MB RAM, 2MB Video RAM.
A hard disc with at least 10 MB free space.
An SVGA monitor, which can support 16 M colors at 640 X 480 resolution.

The above configuration is a minimal.. Higher configuration should be considered when speed is a criterion.

Software

- The Microsoft Windows operating environment.
  DOS(Ver 6.0 or higher)

APPENDIX B

INFORMATION ABOUT THE WINDOWS HELP FILE

The Windows help file gives high quality context sensitive help for the software users. It provides the facility of hyperlinks and hypertexts. Follow the below given steps to create a Help file for your application.

- Type your help text in any Wordprocessors or even in Windows Notepad and save the text in *.RTF(Rich Text) format.
    - Make your help project file (*.HPJ) specifying the RTF file in it.

      For E.g.: [options]

      ```
      Compress=False
      [Files]
      sam.rtf
      ```
- can be the contents of a *.HPJ file.
- Compile your project file using the Windows Help Compiler HC31.EXE which comes along with Visual C++, Visual Basic and Borland C++.
    - This will give your *.HLP file
    - Now write code in your program to call the help file.

APPENDIX - C

****

SOURCE CODE LISTING OF COL_DYE.MAK

****

```
Dim pq As Variant
Sub Command3D1_Click ( )
'About form displays the details of the project
'and the team
Unload ab_frm
If previous_form = "opt_frm" Then
opt_frm.Show
opt_frm.SetFocus
Else
start_frm.Show
start_frm.SetFocus
End If
End Sub

Sub Command3D1_GotFocus ( )
label6.Caption = "Click on Back to return to parent Window"
End Sub

Sub Command3D1_Click ( )
' Adding a new record to the database
Data1.Recordset.MoveLast
text1.SetFocus
Data1.Recordset.AddNew
Data1.Recordset.Update
End Sub

Sub Command3D2_Click ( )
' Deleting the record from the database
If Data1.Database Is Nothing Then
MsgBox "Open failed(Database Invalid)"
End If
If Data1.Recordset Is Nothing Then
MsgBox "Open failed(Recordset invalid)"
End If
msg = "Delete this Record ?"
ch = MsgBox(msg, 1, "Deleting Record")
If ch = 1 Then
```

```
Data1.Recordset.Delete
Data1.Recordset.MoveNext
End If
End Sub

Sub Command3D3_Click ( )
' saving the data into the database
If Data1.Database Is Nothing Then
MsgBox "Open failed(Database Invalid)"
End If
If Data1.Recordset Is Nothing Then
MsgBox "Open failed(Recordset invalid)"
End If
msg = "Save the current changes ?"
MsgBox msg, 48, "Saving.."
Data1.Recordset.Update
End Sub

Sub Command3D5_Click ( )
Unload Me
opt_frm.Show
End Sub

Sub Data1_Error (DataErr As Integer, Response As Integer)
MsgBox "data open failed!"
End Sub

Sub Data1_Reposition ( )
'dye color
If Data1.Database Is Nothing Then
MsgBox "Open failed! May be end of record."
End If
'If Data1.Recordset.EOF Then
'MsgBox "error"
'End If
If IsNull(Data1.Recordset) Then
MsgBox "Open failed! May be end of record."
Else
picture1.BackColor = RGB(Val(text2.Text), Val(text3.Text),
Val(text4.Text))
End If
End Sub
Sub Data1_Validate (Action As Integer, Save As Integer)
'dye color
If Data1.Database Is Nothing Then
```

```
MsgBox "Open failed! May be end of record."
End If
'If Data1.Recordset.EOF Then
'MsgBox "error"
'End If
If IsNull(Data1.Recordset) Then
MsgBox "Open failed! May be end of record."
Else
 picture1.BackColor = RGB(Val(text2.Text), Val(text3.Text),
Val(text4.Text))
End If
End Sub


Sub Form_Load ( )
dbase_frm.Data1.Refresh
picture1.BackColor = RGB(Data1.Recordset("red"),
Data1.Recordset("green"), Data1.Recordset("blue"))
End Sub
```

****

Load color file routine

****

```
Dim filename  As String
Sub Command3D1_Click ( )
'This event procedure is used to open a particular
 'color file if it is processed previously
Dim FileData, Msg, NL   ' Declare variables.
TESTFILE = text1.Text
 On Error GoTo Fill

 NL = Chr(10)    ' Define newline.
 'MakeDataFile    ' Create sample data file.
 Open TESTFILE For Input As #1 ' Open to read file.
   Do While Not EOF(1)
   Input #1, FileData  ' Read line of data.
   Msg = Msg & FileData & NL   ' Construct message from data.
   Loop
   Close #1    ' Close file.
   MsgBox Msg  ' Display message.
   'Kill "TESTFILE" ' Remove file from disk.
   Exit Sub
```

```
Fill:
Action = FileErrors(Err)
If Action = 0 Then
Resume
Else
Exit Sub
End If
End Sub

Sub Command3D1_GotFocus ( )
file1.Path = dir1.Path
End Sub

Sub Command3D2_Click ( )
'This event procedure is used to delete a particular
'color file from the disk
   Dim Ansr, DelFile, Msg  ' Declare variables.
   On Error GoTo Errhandler   ' Set up error handler.
   DelFile = UCase(InputBox("Enter the file you want to delete specifying
the path"))
   If Len(DelFile) Then    ' Check for entry.
   Ansr = MsgBox("Sure you want to delete " & DelFile & "?", 4)
   If Ansr = 6 Then    ' User chose "Yes."
        Msg = "Deleting " & DelFile & " from your disk."
        Kill DelFile    ' Delete file from disk.
        text1.Text = " "
        file1.Refresh
        file1.Path = dir1.Path
   Else
        Msg = DelFile & " was not deleted."
   End If
   Else
   Msg = "You didn't enter a file name."
   End If
   MsgBox Msg  ' Display message.

  Errhandler:
   If Err = 53 Then    ' Error 53 is "File not Found".
        Msg = "Sorry, the file you named could not be found."
   Else
        Msg = "Sorry, unable to delete file."
   End If
   Resume Next
 End Sub
```

```
Sub Command3D3_Click ( )
'This will cancel the current operation saving the state
Unload Me
opt_frm.Show
End Sub

Sub Dir1_Change ( )
' This event procedure is used to select the directory
file1.Pattern = "*.col"
file1.Path = dir1.Path
'text1.Text = file1.file
End Sub


Sub Drive1_Change ( )
'This event procedure is used to change
'the drives in the system to select the color files
On Error GoTo drivehandler
dir1.Path = drive1
Exit Sub

drivehandler:
 MsgBox Error$, 48, "Disk Error"
 Resume Next
End Sub


Sub File1_Click ( )
'This event procedure is used to select the color file
On Error GoTo errorhandler
command3d1.Enabled = True
command3d2.Enabled = True
text1.Text = file1
Exit Sub
errorhandler:
MsgBox "ERROR"
Resume Next
End Sub


Sub Form_Load ( )
ChDir CurDir$
file1.Pattern = "*.col"
command3d1.Enabled = False
command3d2.Enabled = False
```

```
End Sub


Sub Text1_Change ( )
' This text box displays the selected file
' which will have the details about the
' color input and also the processed result
End Sub

Sub abt_Click ( )
'Calling th ABOUT FORM
previous_frm = "opt_frm"
ab_frm.Show
End Sub
Sub Command3D1_Click ( )
'Cancelling the Event
Unload Me
start_frm.Show
End Sub


Sub Form_Load ( )
ok_opt.Enabled = False
ok_opt.Visible = True
End Sub

Sub hlp_Click ( )
'Enables the help file
Keydown HELPKEYCODE, 0, 1
End Sub

Sub ok_opt_Click ( )
'This procedure is used to select the options
'The different options are
'* You can select an existing colr file
'* You can go for matching the color
If option3d1.Value = True Then
col_frm.Show
Else
 If option3d3.Value = True Then
  class_frm.Show
  Else
    'Unload Me
    class_frm.Show
  End If
End If
End Sub
Sub Option3D1_Click (Value As Integer)
'Option Selection
 ok_opt.Visible = True
 ok_opt.Enabled = True
End Sub


Sub Option3D2_Click (Value As Integer)
 ' Option selection
 ok_opt.Visible = True
ok_opt.Enabled = True
End Sub


Sub Option3D3_Click (Value As Integer)
 ok_opt.Visible = True
ok_opt.Enabled = True
End Sub
```

****

Saving the color file

****

```
Sub Com2_Click ( )
' This procedure will save the color file
' after processing it to find the optimized reciepe
' of the dyes

'Dim Msg
'Dim Title
'Dim DefVal
'Dim FileData
Dim filename As String
CUR_DIR = "c:\vb\project"
'Dim NL
'NL = Chr(10)
  Dim FileData, msg, NL   ' Declare variables.
'ChDir CurDir$
'fnumber = FreeFile
msg = "Please enter the file name"
Title = "Reading Filename"
```

```
filename = InputBox$(msg, Title, DefVal)
'Open filename & "col" For Output As fnumber
On Error GoTo Fil
'Print #fnumber, "vsroll1.Value", "Vscroll2.value", "Vscroll3.Value"
'Close #fnumber
'Open filename & "col" For Input As #fnumber ' Open to read file.
'   Do While Not EOF(1)
'      Input #fnumber, FileData  ' Read line of data.
'      Msg = Msg & FileData & NL   ' Construct message from data.
'   Loop
'   Close #fnumber    ' Close file.
'
   Msg1 = "The RGB Components"
   NL = Chr(10)    ' Define newline.
   Open "c:\vb\project\" & filename For Output As #1    ' Open to write file.
   Print #1, VScroll1.Value, Vscroll2.Value, Vscroll3.Value
   Close #1    ' Close file.
   Open CUR_DIR & filename For Input As #1 ' Open to read file.
   Do While Not EOF(1)
        Line Input #1, FileData ' Read a line of data.
        msg = Msg1 & NL1 & FileData & NL   ' Construct message.
   Loop
   Close #1    ' Close file.
   MsgBox msg  ' Display message.
   'Kill "sam.col" ' Remove file from disk.

  Exit Sub
  Fil:
  Action = FileErrors(Err)
  If Action = 0 Then
  Resume
  Else
  Exit Sub
  End If
  End Sub



  Sub com3_Click ( )
  ' Cancelling the event
  Unload Me
  opt_frm.Show
  End Sub
```

****

Routine for removing dye and re optimizing

****

```
Sub Command1_Click ( )
Dim prc As Single
 g_var = 1
flag = 1
prc = 0
Load dbase_frm
dbase_frm.Data1.Refresh
Do While dbase_frm.Data1.Recordset.EOF = False
dbase_frm.Data1.Recordset.MoveNext
flag = flag + 1
Loop
flag = flag - 1
dbase_frm.Data1.Enabled = False
'here starts

pal_frm.mousepointer = 11
  Load dbase_frm
  dbase_frm.Data1.Refresh
  ReDim a(6, (flag + 7)) As Single
  ReDim dupe(6, (flag + 7)) As Single
  ReDim c2(6, (flag + 7)) As Single
  For i = 1 To 6
  For j = 1 To flag + 6
  c2(i, j) = 0
  dupe(i, j) = 0
  Next j
  Next i

  For i = 1 To 6
  For j = 1 To flag + 6
  a(i, j) = 0
  Next j
  Next i

  For i = 2 To flag + 1
  a(1, i) = dbase_frm.Data1.Recordset("cost")
  dbase_frm.Data1.Recordset.MoveNext
  dupe(1, i) = a(1, i)
  Next i
  For i = 2 To 4
```

```
a(1, flag + i) = 2500
dupe(1, flag + i) = 2500



Next i
dbase_frm.Data1.Refresh
For j = 2 To flag + 1
a(2, j) = dbase_frm.Data1.Recordset("red")
dupe(2, j) = a(2, j)
a(3, j) = dbase_frm.Data1.Recordset("green")
dupe(3, j) = a(3, j)
a(4, j) = dbase_frm.Data1.Recordset("blue")
dupe(4, j) = a(4, j)
dbase_frm.Data1.Recordset.MoveNext
Next j
For i = 2 To 4
For j = flag + 2 To flag + 4
If (j = (i + flag)) Then
a(i, j) = 1
dupe(i, j) = 1
Else
   a(i, j) = 0
   dupe(i, j) = 0
End If
Next j
Next i
a(2, flag + 5) = pal_frm.Text1.Text
dupe(2, flag + 5) = pal_frm.Text1.Text
  If a(2, flag + 5) = 0 Then
            a(2, flag + 5) = 0
            dupe(2, flag + 5) = 0
  End If
a(3, flag + 5) = pal_frm.Text2.Text
dupe(3, flag + 5) = pal_frm.Text2.Text
  If a(3, flag + 5) = 0 Then
            a(3, flag + 5) = 0
            dupe(3, flag + 5) = 0
  End If

a(4, flag + 5) = pal_frm.Text3.Text
dupe(4, flag + 5) = pal_frm.Text3.Text
  If a(4, flag + 5) = 0 Then
            a(4, flag + 5) = 0
            dupe(4, flag + 5) = 0
  End If
```

```
For i = 2 To 4
a(i, 1) = 2500
dupe(i, 1) = a(i, 1)
Next i
equa
dbase_frm.Data1.Enabled = True

pal_frm.mousepointer = 0
res_frm.Refresh
res_frm.Show

 If fn_var(1) <= flag Then
 dbase_frm.Data1.Refresh

 For j = 0 To fn_var(1) - 1
 If (fn_var(1) - 1) = 0 Then
dbase_frm.Data1.Recordset.MoveNext
 Exit For
 Else
 dbase_frm.Data1.Recordset.MoveNext
 End If
 Next j
dbase_frm.Data1.Recordset.MovePrevious
 res_frm.Frame3D2.Caption = dbase_frm.Data1.Recordset("dye_name")
 res_frm.Picture1.BackColor =
RGB(Int(dbase_frm.Data1.Recordset("red")),
Int(dbase_frm.Data1.Recordset("green")),
Int(dbase_frm.Data1.Recordset("blue")))
 res_frm.Text1.Text = a(2, flag + 5) & "   " & "Kg"
 prc = prc + a(2, flag + 5)

 Else
 res_frm.Frame3D2.Enabled = False
 res_frm.Frame3D2.Visible = False
 res_frm.Picture1.Visible = False
 res_frm.Text1.Visible = False
 res_frm.Frame3D5.Visible = False
 a(5, flag + 5) = a(5, flag + 5) - a(2, 1) * a(2, flag + 5)
 End If

 If fn_var(2) <= flag Then
 dbase_frm.Data1.Refresh
```

```
For j = 0 To fn_var(2) - 1
If (fn_var(2) - 1) = 0 Then
dbase_frm.Data1.Recordset.MoveNext
Exit For
Else
dbase_frm.Data1.Recordset.MoveNext
End If
Next j
dbase_frm.Data1.Recordset.MovePrevious
res_frm.Frame3D3.Caption = dbase_frm.Data1.Recordset("dye_name")
res_frm.Picture2.BackColor =
RGB(Int(dbase_frm.Data1.Recordset("red")),
Int(dbase_frm.Data1.Recordset("green")),
Int(dbase_frm.Data1.Recordset("blue")))
res_frm.Text2.Text = a(3, flag + 5) & "  " & "Kg"
prc = prc + a(3, flag + 5)

Else
res_frm.Frame3D3.Enabled = False
res_frm.Frame3D3.Visible = False
res_frm.Picture2.Visible = False
res_frm.Text2.Visible = False
res_frm.Frame3D6.Visible = False
a(5, flag + 5) = a(5, flag + 5) - a(3, 1) * a(3, flag + 5)

End If

If fn_var(3) <= flag Then
dbase_frm.Data1.Refresh

For j = 0 To fn_var(3) - 1
If (fn_var(3) - 1) = 0 Then
dbase_frm.Data1.Recordset.MoveNext
Exit For
Else
dbase_frm.Data1.Recordset.MoveNext
End If
Next j
dbase_frm.Data1.Recordset.MovePrevious
res_frm.Frame3D4.Caption = dbase_frm.Data1.Recordset("dye_name")
res_frm.Picture3.BackColor =
RGB(Int(dbase_frm.Data1.Recordset("red")),
Int(dbase_frm.Data1.Recordset("green")),
Int(dbase_frm.Data1.Recordset("blue")))
res_frm.Text3.Text = a(4, flag + 5) & "  " & "Kg"
```

```
prc = prc + a(4, flag + 5)
Else
res_frm.Frame3D4.Enabled = False
res_frm.Frame3D4.Visible = False
res_frm.Picture3.Visible = False
res_frm.Text3.Visible = False
res_frm.Frame3D7.Visible = False
a(5, flag + 5) = a(5, flag + 5) - a(4, 1) * a(4, flag + 5)
End If
If prc = 0 Then
ms1 = MsgBox("This Color may be existing as a pure Dye Try the View
Database Option", 36, "Check!")
If ms1 = 6 Then
pal_frm.mousepointer = 0
pal_frm.SetFocus
Exit Sub
End If
End If
res_frm.Text4.Text = "Optimized Cost is  " & "Rs " & a(5, flag + 5) / prc &
" per  Kg"


End Sub

Sub Form_Load ()
g_var = 0
pan.Caption = ""
command1.Enabled = True
End Sub


Sub res_pic_Change ()
'Resultant color display
pan.ForeColor = RGB(VScroll1.Value, Vscroll2.Value, Vscroll3.Value)
pan.Caption = "Resultant Color"
End Sub
***************************************************************************************
```

Color extraction from Windows Palette

```
***************************************************************************************

Sub res_pic_Click ()
Dim col As Long
Dim r As Long
Dim g As Long
Dim b As Long
```

```
col = res_pic.BackColor
r = col And 255
g = (col And 65280) / 256
b = (col And 16711680) / 65536
VScroll1.Value = r
Vscroll2.Value = g
Vscroll3.Value = b
Text1.Text = r
Text2.Text = g
Text3.Text = b
Pic1.BackColor = RGB(r, 0, 0)
Pic2.BackColor = RGB(0, g, 0)
Pic3.BackColor = RGB(0, 0, b)
End Sub

Sub res_pic_GotFocus ()
 pan.ForeColor = RGB(VScroll1.Value, Vscroll2.Value, Vscroll3.Value)
 pan.Caption = "Resultant Color"
End Sub


Sub res_pic_MouseDown (Button As Integer, Shift As Integer, X As
Single, Y As Single)
 pan.ForeColor = RGB(VScroll1.Value, Vscroll2.Value, Vscroll3.Value)
 pan.Caption = "Resultant Color"
End Sub
```

****

Color generation routine

****

```
Sub VScroll1_Change ()
' To change the color value
 Dim red As Long
Dim green As Long
Dim blue As Long
hyper_text.Text = "Extracting Red Component"
red = Val(VScroll1.Value)
green = Val(Vscroll2.Value)
blue = Val(Vscroll3.Value)
Text1.Text = Val(VScroll1.Value)
Text2.Text = Val(Vscroll2.Value)
Text3.Text = Val(Vscroll3.Value)
Pic1.BackColor = RGB(red, 0, 0)
Pic2.BackColor = RGB(0, green, 0)
Pic3.BackColor = RGB(0, 0, blue)
```

```
res_pic.BackColor = RGB(red, green, blue)
Text1.Text = VScroll1.Value
End Sub


Sub VScroll1_GotFocus ()
 pan.ForeColor = &HFF&
pan.Caption = "To change the RED values"
End Sub


Sub VScroll1_Scroll ()
 Dim red As Long
Dim green As Long
Dim blue As Long
hyper_text.Text = "Extracting Red Component"
red = Val(VScroll1.Value)
green = Val(Vscroll2.Value)
blue = Val(Vscroll3.Value)
Text1.Text = Val(VScroll1.Value)
Text2.Text = Val(Vscroll2.Value)
Text3.Text = Val(Vscroll3.Value)
Pic1.BackColor = RGB(red, 0, 0)
Pic2.BackColor = RGB(0, green, 0)
Pic3.BackColor = RGB(0, 0, blue)
res_pic.BackColor = RGB(red, green, blue)
End Sub



Sub VScroll2_Change ()
' To change the color value
Dim red As Long
Dim green As Long
Dim blue As Long
hyper_text.Text = "Extracting Green Component"
red = Val(VScroll1.Value)
green = Val(Vscroll2.Value)
blue = Val(Vscroll3.Value)
Text1.Text = Val(VScroll1.Value)
Text2.Text = Val(Vscroll2.Value)
Text3.Text = Val(Vscroll3.Value)
Pic1.BackColor = RGB(red, 0, 0)
Pic2.BackColor = RGB(0, green, 0)
Pic3.BackColor = RGB(0, 0, blue)
```

```
res_pic.BackColor = RGB(red, green, blue)
Text2.Text = Vscroll2.Value
End Sub


Sub VScroll2_GotFocus ()
pan.ForeColor = &HC000&
pan.Caption = "To change the GREEN values"
End Sub

Sub VScroll2_Scroll ()
Dim red As Long
Dim green As Long
Dim blue As Long
hyper_text.Text = "Extracting Green Component"
red = Val(VScroll1.Value)
green = Val(Vscroll2.Value)
blue = Val(Vscroll3.Value)
Text1.Text = Val(VScroll1.Value)
Text2.Text = Val(Vscroll2.Value)
Text3.Text = Val(Vscroll3.Value)
Pic1.BackColor = RGB(red, 0, 0)
Pic2.BackColor = RGB(0, green, 0)
Pic3.BackColor = RGB(0, 0, blue)
res_pic.BackColor = RGB(red, green, blue)
End Sub



Sub VScroll3_Change ()
'To change the color value
Dim red As Long
Dim green As Long
Dim blue As Long
hyper_text.Text = "Extracting Blue Component"
red = Val(VScroll1.Value)
green = Val(Vscroll2.Value)
blue = Val(Vscroll3.Value)
Text1.Text = Val(VScroll1.Value)
Text2.Text = Val(Vscroll2.Value)
Text3.Text = Val(Vscroll3.Value)
Pic1.BackColor = RGB(red, 0, 0)
Pic2.BackColor = RGB(0, green, 0)
Pic3.BackColor = RGB(0, 0, blue)
res_pic.BackColor = RGB(red, green, blue)
```

```
Text3.Text = Vscroll3.Value
End Sub



Sub VScroll3_GotFocus ()
pan.ForeColor = &HFF0000
pan.Caption = "To change the BLUE values"
End Sub

Sub VScroll3_Scroll ()
Dim red As Long
Dim green As Long
Dim blue As Long
hyper_text.Text = "Extracting Blue Component"
red = Val(VScroll1.Value)
green = Val(Vscroll2.Value)
blue = Val(Vscroll3.Value)
Text1.Text = Val(VScroll1.Value)
Text2.Text = Val(Vscroll2.Value)
Text3.Text = Val(Vscroll3.Value)
Pic1.BackColor = RGB(red, 0, 0)
Pic2.BackColor = RGB(0, green, 0)
Pic3.BackColor = RGB(0, 0, blue)
res_pic.BackColor = RGB(red, green, blue)
End Sub


Sub win_pal_Click ()
'This event procedure will extract the color components
' from the windows pallette and fixes it to the main pal
' form

************************************************************************************
                              Color Extraction Routine
************************************************************************************

Dim r As Long
Dim g As Long
Dim b As Long
Dim col As Long
CMDialog1.Filter = &H1&
CMDialog1.Action = 3
col = CMDialog1.Color
r = col And 255
g = (col And 65280) / 256
```

```
b = (col And 16711680) / 65536
VScroll1.Value = r
Vscroll2.Value = g
Vscroll3.Value = b
res_pic.BackColor = CMDialog1.Color
pan.ForeColor = &H800080
 pan.Caption = "Windows Pallette"
End Sub


Sub win_pal_DragOver (Source As Control, X As Single, Y As Single,
State As Integer)
pan.Caption = "Resultant Color"
End Sub


Sub win_pal_MouseDown (Button As Integer, Shift As Integer, X As
Single, Y As Single)
pan.Caption = "Windows Pallette"
End Sub
```

```
*************************************************************************************
```

Routine for opening a pre processed Color file

```
*************************************************************************************
```

```
Sub Command3D2_Click ()
   'This event procedure is used to open a particular
   'color file if it is processed previously
   Dim FileData, Msg, NL   ' Declare variables.
   ms = "Do you want save this combination ?"
   mas = MsgBox(ms, 36, "Saving")

   If mas <> 6 Then
   Unload Me
   pal_frm.SetFocus

   Else
   fil = InputBox("Please enter filename(*.col):")

   On Error GoTo Fi

   NL = Chr(10)   ' Define newline.
   'MakeDataFile    ' Create sample data file.

   Open fil For Output As #1 ' Open to read file.
```

```
  Print #1, Frame3D2.Caption, NL, Frame3D3.Caption, NL,
Frame3D4.Caption, NL, Text1.Text, NL, Text2.Text, NL, Text3.Text,
Text4.Text
  Close #1
  Open fil For Input As #1 ' Open to read file.
  Do While Not EOF(1)
  Input #1, FileData  ' Read line of data.
  Msg = Msg & FileData & NL   ' Construct message from data.
  Loop
  Close #1    ' Close file.
  MsgBox Msg  ' Display message.

  'Msg = NL & FileData & NL   ' Construct message from data.
  'MsgBox Msg

  Close #1    ' Close file.
  End If
  pal_frm.SetFocus
  Exit Sub

Fi:
Action = FileErrors(Err)
If Action = 0 Then
Resume
Else
Exit Sub
End If
End Sub
*******************************************************************************
```

Reoptimizing with the first Dye

```
*******************************************************************************

Sub Picture1_Click ()
Dim du_name As String
Dim var As Integer
res_frm.mousepointer = 11
var = 1
g_var = g_var + 1
du_name = Frame3D2.Caption
dbase_frm.Data1.Refresh
Do Until dbase_frm.Data1.Recordset("dye_name") = du_name
dbase_frm.Data1.Recordset.MoveNext
var = var + 1
Loop
If g_var > flag - 3 Then
```

```
MsgBox "No more changes possible!"
Exit Sub
End If
For i = 1 To 4
dupe(i, var + 1) = 0
Next i
 res_frm.Picture1.Cls
 res_frm.Picture2.Cls
 res_frm.Picture3.Cls
 Frame3D2.Visible = False
 Frame3D3.Visible = False
 Frame3D4.Visible = False
 Frame3D5.Visible = False
 Frame3D6.Visible = False
 Frame3D7.Visible = False
 Text1.Visible = False
 Text2.Visible = False
 Text3.Visible = False
  Text4.Visible = False
 For i = 1 To 5
 For j = 1 To flag + 5
 a(i, j) = dupe(i, j)
 Next j
 Next i
 equa
  If fn_var(1) <= flag Then
  dbase_frm.Data1.Refresh

  For j = 0 To fn_var(1) - 1
  If (fn_var(1) - 1) = 0 Then
  dbase_frm.Data1.Recordset.MoveNext
  Exit For
  Else
  dbase_frm.Data1.Recordset.MoveNext
  End If
  Next j
 dbase_frm.Data1.Recordset.MovePrevious
  Frame3D2.Visible = True
  Frame3D5.Visible = True
  Text1.Visible = True
   Text4.Visible = True

  res_frm.Frame3D2.Caption = dbase_frm.Data1.Recordset("dye_name")
  res_frm.Picture1.BackColor =
 RGB(Int(dbase_frm.Data1.Recordset("red")),
Int(dbase_frm.Data1.Recordset("green")),
Int(dbase_frm.Data1.Recordset("blue")))
 res_frm.Text1.Text = RTrim(a(2, flag + 5)) & "   " & "Kg"
 prc = prc + a(2, flag + 5)

 Else
 res_frm.Frame3D2.Enabled = False
 res_frm.Frame3D2.Visible = False
 res_frm.Picture1.Visible = False
 res_frm.Text1.Visible = False
 res_frm.Frame3D5.Visible = False
 a(5, flag + 5) = a(5, flag + 5) - a(2, 1) * a(2, flag + 5)
 End If

 If fn_var(2) <= flag Then
 dbase_frm.Data1.Refresh

 For j = 0 To fn_var(2) - 1
 If (fn_var(2) - 1) = 0 Then
 dbase_frm.Data1.Recordset.MoveNext
 Exit For
 Else
 dbase_frm.Data1.Recordset.MoveNext
 End If
 Next j
 dbase_frm.Data1.Recordset.MovePrevious
 Frame3D3.Visible = True
 Frame3D6.Visible = True
 Text2.Visible = True
  Text4.Visible = True

 res_frm.Frame3D3.Caption = dbase_frm.Data1.Recordset("dye_name")
 res_frm.Picture2.BackColor =
RGB(Int(dbase_frm.Data1.Recordset("red")),
Int(dbase_frm.Data1.Recordset("green")),
Int(dbase_frm.Data1.Recordset("blue")))
 res_frm.Text2.Text = RTrim(a(3, flag + 5)) & "  " & "Kg"
 prc = prc + a(3, flag + 5)

 Else
 res_frm.Frame3D3.Enabled = False
 res_frm.Frame3D3.Visible = False
 res_frm.Picture2.Visible = False
 res_frm.Text2.Visible = False
 res_frm.Frame3D6.Visible = False
```

```
a(5, flag + 5) = a(5, flag + 5) - a(3, 1) * a(3, flag + 5)

End If

If fn_var(3) <= flag Then
dbase_frm.Data1.Refresh

For j = 0 To fn_var(3) - 1
If (fn_var(3) - 1) = 0 Then
dbase_frm.Data1.Recordset.MoveNext
Exit For
Else
dbase_frm.Data1.Recordset.MoveNext
End If
Next j
dbase_frm.Data1.Recordset.MovePrevious
Frame3D4.Visible = True
Frame3D7.Visible = True
Text3.Visible = True
 Text4.Visible = True
res_frm.Frame3D4.Caption = dbase_frm.Data1.Recordset("dye_name")
res_frm.Picture3.BackColor =
RGB(Int(dbase_frm.Data1.Recordset("red")),
Int(dbase_frm.Data1.Recordset("green")),
Int(dbase_frm.Data1.Recordset("blue")))
res_frm.Text3.Text = RTrim(a(4, flag + 5)) & "   " & "Kg"
prc = prc + a(4, flag + 5)
Else
res_frm.Frame3D4.Enabled = False
res_frm.Frame3D4.Visible = False
res_frm.Picture3.Visible = False
res_frm.Text3.Visible = False
res_frm.Frame3D7.Visible = False
a(5, flag + 5) = a(5, flag + 5) - a(4, 1) * a(4, flag + 5)
End If

res_frm.Text4.Text = "Optimized Cost is  " & "Rs " & a(5, flag + 5) / prc &
" per  Kg"
res_frm.mousepointer = 0
End Sub
```

```
*******************************************************************************
```

Reoptimizing with the second Dye

```
*******************************************************************************

Sub Picture2_Click ()
```

```
Dim du_name As String
Dim var As Integer
var = 1
res_frm.mousepointer = 11
g_var = g_var + 1
check_flag = 0
du_name = Frame3D3.Caption
dbase_frm.Data1.Refresh
Do Until dbase_frm.Data1.Recordset("dye_name") = du_name
dbase_frm.Data1.Recordset.MoveNext
var = var + 1
Loop
If g_var > flag - 3 Then
MsgBox "No more changes possible!"
Exit Sub
End If
For i = 1 To 4
dupe(i, var + 1) = 0
Next i
check_flag = 1
For i = 1 To 5
For j = 1 To flag + 5
a(i, j) = dupe(i, j)
Next j
Next i
 res_frm.Picture1.Cls
 res_frm.Picture2.Cls
 res_frm.Picture3.Cls
 Frame3D2.Visible = False
 Frame3D3.Visible = False
 Frame3D4.Visible = False
 Frame3D5.Visible = False
 Frame3D6.Visible = False
 Frame3D7.Visible = False
 Text1.Visible = False
 Text2.Visible = False
 Text3.Visible = False
  Text4.Visible = False
 equa
  If fn_var(1) <= flag Then
  dbase_frm.Data1.Refresh

  For j = 0 To fn_var(1) - 1
  If fn_var(1) - 1 = 0 Then
  dbase_frm.Data1.Recordset.MoveNext
```

```
Exit For
 Else
dbase_frm.Data1.Recordset.MoveNext
End If
Next j
dbase_frm.Data1.Recordset.MovePrevious
Frame3D2.Visible = True
Frame3D5.Visible = True
Text1.Visible = True
 Text4.Visible = True

res_frm.Frame3D2.Caption = dbase_frm.Data1.Recordset("dye_name")
res_frm.Picture1.BackColor =
RGB(Int(dbase_frm.Data1.Recordset("red")),
Int(dbase_frm.Data1.Recordset("green")),
Int(dbase_frm.Data1.Recordset("blue")))
 res_frm.Text1.Text = a(2, flag + 5) & "   " & "Kg"
 prc = prc + a(2, flag + 5)

 Else
 res_frm.Frame3D2.Enabled = False
 res_frm.Frame3D2.Visible = False
 res_frm.Picture1.Visible = False
 res_frm.Text1.Visible = False
 res_frm.Frame3D5.Visible = False
 a(5, flag + 5) = a(5, flag + 5) - a(2, 1) * a(2, flag + 5)
 End If

 If fn_var(2) <= flag Then
 dbase_frm.Data1.Refresh

 For j = 0 To fn_var(2) - 1
 If fn_var(2) - 1 = 0 Then
 dbase_frm.Data1.Recordset.MoveNext
 Exit For
 Else
 dbase_frm.Data1.Recordset.MoveNext
 End If
 Next j
 dbase_frm.Data1.Recordset.MovePrevious
 Frame3D3.Visible = True
 Frame3D6.Visible = True
 Text2.Visible = True
  Text4.Visible = True
```

```
res_frm.Frame3D3.Caption = dbase_frm.Data1.Recordset("dye_name")
res_frm.Picture2.BackColor =
RGB(Int(dbase_frm.Data1.Recordset("red")),
Int(dbase_frm.Data1.Recordset("green")),
Int(dbase_frm.Data1.Recordset("blue")))
res_frm.Text2.Text = a(3, flag + 5) & "  " & "Kg"
prc = prc + a(3, flag + 5)

Else
res_frm.Frame3D3.Enabled = False
res_frm.Frame3D3.Visible = False
res_frm.Picture2.Visible = False
res_frm.Text2.Visible = False
res_frm.Frame3D6.Visible = False
a(5, flag + 5) = a(5, flag + 5) - a(3, 1) * a(3, flag + 5)

End If

If fn_var(3) <= flag Then
dbase_frm.Data1.Refresh

For j = 0 To fn_var(3) - 1
If fn_var(3) - 1 = 0 Then
dbase_frm.Data1.Recordset.MoveNext
Exit For
Else
dbase_frm.Data1.Recordset.MoveNext
End If
Next j
dbase_frm.Data1.Recordset.MovePrevious
Frame3D4.Visible = True
Frame3D7.Visible = True
Text3.Visible = True
 Text4.Visible = True

res_frm.Frame3D4.Caption = dbase_frm.Data1.Recordset("dye_name")
res_frm.Picture3.BackColor =
RGB(Int(dbase_frm.Data1.Recordset("red")),
Int(dbase_frm.Data1.Recordset("green")),
Int(dbase_frm.Data1.Recordset("blue")))
res_frm.Text3.Text = a(4, flag + 5) & "   " & "Kg"
prc = prc + a(4, flag + 5)
Else
res_frm.Frame3D4.Enabled = False
res_frm.Frame3D4.Visible = False
```

```
 res_frm.Picture3.Visible = False
 res_frm.Text3.Visible = False
 res_frm.Frame3D7.Visible = False
a(5, flag + 5) = a(5, flag + 5) - a(4, 1) * a(4, flag + 5)
 End If

 res_frm.Text4.Text = "Optimized Cost is  " & "Rs " & a(5, flag + 5) / prc &
" per  Kg"
 res_frm.mousepointer = 0
End Sub
```

****

Reoptimizing with the third Dye

****

```
Sub Picture3_Click ()
Dim du_name As String
Dim var As Integer
res_frm.mousepointer = 11
var = 1
g_var = g_var + 1
check_flag = 0
du_name = Frame3D4.Caption
dbase_frm.Data1.Refresh
Do Until dbase_frm.Data1.Recordset("dye_name") = du_name
dbase_frm.Data1.Recordset.MoveNext
var = var + 1
Loop
If g_var > flag - 3 Then
MsgBox "No more changes possible!"
Exit Sub
End If

For i = 1 To 4
dupe(i, var + 1) = 0
Next i
check_flag = 1
For i = 1 To 5
For j = 1 To flag + 5
a(i, j) = dupe(i, j)
Next j
Next i
 res_frm.Picture1.Cls
 res_frm.Picture2.Cls
 res_frm.Picture3.Cls
 Frame3D2.Visible = False
```

```
Frame3D3.Visible = False
Frame3D4.Visible = False
Frame3D5.Visible = False
Frame3D6.Visible = False
Frame3D7.Visible = False
Text1.Visible = False
Text2.Visible = False
Text3.Visible = False
 Text4.Visible = False
equa
If fn_var(1) <= flag Then
dbase_frm.Data1.Refresh

For j = 0 To fn_var(1) - 1
If fn_var(1) - 1 = 0 Then
dbase_frm.Data1.Recordset.MoveNext
Exit For
Else
dbase_frm.Data1.Recordset.MoveNext
End If
Next j
dbase_frm.Data1.Recordset.MovePrevious
Frame3D2.Visible = True
Frame3D5.Visible = True
Text1.Visible = True
 Text4.Visible = True

res_frm.Frame3D2.Caption = dbase_frm.Data1.Recordset("dye_name")
res_frm.Picture1.BackColor =
RGB(Int(dbase_frm.Data1.Recordset("red")),
Int(dbase_frm.Data1.Recordset("green")),
Int(dbase_frm.Data1.Recordset("blue")))
res_frm.Text1.Text = a(2, flag + 5) & "   " & "Kg"
prc = prc + a(2, flag + 5)

Else
res_frm.Frame3D2.Enabled = False
res_frm.Frame3D2.Visible = False
res_frm.Picture1.Visible = False
res_frm.Text1.Visible = False
res_frm.Frame3D5.Visible = False
a(5, flag + 5) = a(5, flag + 5) - a(2, 1) * a(2, flag + 5)
End If

If fn_var(2) <= flag Then
```

```
dbase_frm.Data1.Refresh

For j = 0 To fn_var(2) - 1
If fn_var(2) - 1 = 0 Then
dbase_frm.Data1.Recordset.MoveNext
Exit For
Else

dbase_frm.Data1.Recordset.MoveNext
End If
Next j
dbase_frm.Data1.Recordset.MovePrevious
Frame3D3.Visible = True
Frame3D6.Visible = True
Text2.Visible = True
 Text4.Visible = True

 res_frm.Frame3D3.Caption = dbase_frm.Data1.Recordset("dye_name")
 res_frm.Picture2.BackColor =
RGB(Int(dbase_frm.Data1.Recordset("red")),
Int(dbase_frm.Data1.Recordset("green")),
Int(dbase_frm.Data1.Recordset("blue")))
 res_frm.Text2.Text = a(3, flag + 5) & "  " & "Kg"
 prc = prc + a(3, flag + 5)

 Else
 res_frm.Frame3D3.Enabled = False
 res_frm.Frame3D3.Visible = False
 res_frm.Picture2.Visible = False
 res_frm.Text2.Visible = False
 res_frm.Frame3D6.Visible = False
 a(5, flag + 5) = a(5, flag + 5) - a(3, 1) * a(3, flag + 5)

 End If

 If fn_var(3) <= flag Then
 dbase_frm.Data1.Refresh

 For j = 0 To fn_var(3) - 1
 If fn_var(3) - 1 = 0 Then
 dbase_frm.Data1.Recordset.MoveNext
 Exit For
 Else
 dbase_frm.Data1.Recordset.MoveNext
 End If
```

```
Next j
dbase_frm.Data1.Recordset.MovePrevious
Frame3D4.Visible = True
Frame3D7.Visible = True
Text3.Visible = True
Text4.Visible = True

res_frm.Frame3D4.Caption = dbase_frm.Data1.Recordset("dye_name")
res_frm.Picture3.BackColor =
RGB(Int(dbase_frm.Data1.Recordset("red")),
Int(dbase_frm.Data1.Recordset("green")),
Int(dbase_frm.Data1.Recordset("blue")))
res_frm.Text3.Text = a(4, flag + 5) & "   " & "Kg"
prc = prc + a(4, flag + 5)
Else
res_frm.Frame3D4.Enabled = False
res_frm.Frame3D4.Visible = False
res_frm.Picture3.Visible = False
res_frm.Text3.Visible = False
res_frm.Frame3D7.Visible = False
a(5, flag + 5) = a(5, flag + 5) - a(4, 1) * a(4, flag + 5)
End If

res_frm.Text4.Text = "Optimized Cost is  " & "Rs " & a(5, flag + 5) / prc &
" per  Kg"
res_frm.mousepointer = 0
End Sub
```

****

Start up Routines

****

```
Sub Form_Load ()
st_load_frm.Show
st_load_frm.MousePointer = 11
text1.MousePointer = 11
gauge1.MousePointer = 11

gauge1.Max = 1000
For i% = 1 To 1000
 gauge1.Value = i%
 Next i%
st_load_frm.MousePointer = 0
```

```
text1.MousePointer = 0
gauge1.MousePointer = 0

gauge1.Max = 1000

Unload Me
start_frm.Show
End Sub


Sub Gauge1_Change ()
st_load_frm.MousePointer = 11
text1.MousePointer = 11
gauge1.MousePointer = 11

text1.Text = Int(gauge1.Value / 10) + "%"
End Sub

Dim Y As Integer
Dim X As Integer
Dim delay As Integer
Dim toggle As Integer
Dim flap As Integer

Sub about_Click ()
previous_frm = "start_frm"
Unload Me
ab_frm.Show
End Sub


Sub Command3D1_Click ()
Unload Me
opt_frm.Show
End Sub


Sub Command3D2_Click ()
previous_frm = "start_frm"
Unload Me
ab_frm.Show
End Sub
```

****

Saving the status and quitting routine

****

```
Sub Command3D3_Click ()
Const MB_OK = 0, MB_OKCANCEL = 1    ' Define buttons.
   Const MB_YESNOCANCEL = 3, MB_YESNO = 4
   Const MB_ICONSTOP = 16, MB_ICONQUESTION = 32    ' Define
Icons.
   Const MB_ICONEXCLAMATION = 48, MB_ICONINFORMATION = 64
   Const MB_DEFBUTTON2 = 256, IDYES = 6, IDNO = 7  ' Define other.
   Dim DgDef, Msg, Response, Title ' Declare variables.
   Title = "Quitting Color Match"
   ' Put together a sample message box with all the proper components.
   Msg = "This will end your Color Match Session."
   Msg = Msg & " Do you want to continue?"
   DgDef = MB_YESNO + MB_ICONSTOP + MB_DEFBUTTON2  '
Describe dialog.

   Response = MsgBox(Msg, DgDef, Title)    ' Get user response.
   If Response = IDYES Then    ' Evaluate response
        Msg = "You chose Yes.'" and take appropriate
         MsgBox Msg  ' Display action taken.
   Else    ' action.
        Msg = "You chose No or pressed Enter."
        End
   End If
End Sub




Sub Command3D4_Click ()
Keydown HELPKEYCODE, 0, 1
End Sub

Sub exit_Click ()
Const MB_OK = 0, MB_OKCANCEL = 1    ' Define buttons.
   Const MB_YESNOCANCEL = 3, MB_YESNO = 4
   Const MB_ICONSTOP = 16, MB_ICONQUESTION = 32    ' Define
Icons.
   Const MB_ICONEXCLAMATION = 48, MB_ICONINFORMATION = 64
   Const MB_DEFBUTTON2 = 256, IDYES = 6, IDNO = 7  ' Define other.
   Dim DgDef, Msg, Response, Title ' Declare variables.
   Title = "Quitting Color Match"
   ' Put together a sample message box with all the proper components.
   Msg = "This will end your Color Match Session."
   Msg = Msg & " Do you want to continue?"
   DgDef = MB_YESNO + MB_ICONSTOP + MB_DEFBUTTON2  '
Describe dialog.
   Response = MsgBox(Msg, DgDef, Title)    ' Get user response.
   If Response = IDYES Then    ' Evaluate response
        Msg = "You chose Yes."' and take appropriate
         MsgBox Msg  ' Display action taken.
   Else    ' action.
        Msg = "You chose No or pressed Enter."
        End
   End If
 End Sub



 Sub Form_Load ()
 Timer2.Interval = 1000
 c_date.Caption = Date
 picture1.Picture = picClip1.GraphicCell(2)
 Y = 2
 toggle = 1
 End Sub



 Sub go_Click ()
 Unload Me
 opt_frm.Show
 End Sub



 Sub help_Click ()
 Keydown HELPKEYCODE, 0, 1
 End Sub


 Sub runtop ()
 Y = Y + 1
 If Y = 18 Then Y = 0
 picture1.Picture = picClip1.GraphicCell(Y)
 start_frm.Icon = picture3(Y).Picture
 End Sub
```

```
Sub Timer1_Timer ()
If toggle = 1 Then runtop
End Sub


Sub Timer2_Timer ()
c_time.Caption = Time
End Sub
********************************************************************************
```

Module and declarations to check for the file errors

```
********************************************************************************

Global Const Err_DeviceUnavailable = 68
Global Const Err_DiskNotReady = 71, Err_FileAlreadyExists = 58
Global Const Err_TooManyFiles = 67, Err_RenameAcrossDisks = 74
Global Const Err_Path_FileAccessError = 75, Err_DeviceIO = 57
Global Const Err_DiskFull = 61, Err_BadFileName = 64
Global Const Err_BadFileNameOrNumber = 52, Err_FileNotFound = 53
Global Const Err_PathDoesNotExist = 76, Err_BadFileMode = 54
Global Const Err_FileAlreadyOpen = 55, Err_InputPastEndOfFile = 62
Global Const MB_EXCLAIM = 48, MB_STOP = 16
Function FileErrors (errVal As Integer) As Integer
' Return Value  Meaning     Return Value    Meaning
' 0         Resume      2           Unrecoverable error
' 1         Resume Next 3             Unrecognized error
Dim MsgType As Integer
Dim Response As Integer
Dim Action As Integer
Dim Msg As String
MsgType = MB_EXCLAIM
Select Case errVal
   Case Err_DeviceUnavailable  ' Error #68
      Msg = "That device appears to be unavailable."
      MsgType = MB_EXCLAIM + 5
   Case Err_DiskNotReady       ' Error #71
      Msg = "The disk is not ready."
   Case Err_DeviceIO
      Msg = "The disk is full."
   Case Err_BadFileName, Err_BadFileNameOrNumber   ' Errors #64 &
52
      Msg = "That file name is illegal."
   Case Err_PathDoesNotExist                      ' Error #76
      Msg = "That path doesn't exist."
```

```
 Case Err_BadFileMode                          ' Error #54
    Msg = "Can't open your file for that type of access."
 Case Err_FileAlreadyOpen                      ' Error #55
    Msg = "That file is already open."
 Case Err_InputPastEndOfFile                    ' Error #62
 Msg = "This file has a nonstandard end-of-file marker,"
 Msg = Msg + "or an attempt was made to read beyond "
 Msg = Msg + "the end-of-file marker."
 Case Else
    FileErrors = 3
    Exit Function
 End Select
 Response = MsgBox(Msg, MsgType, "File Error")
 Select Case Response
    Case 4          ' Retry button.
       FileErrors = 0
    Case 5          ' Ignore button.
       FileErrors = 1
    Case 1, 2, 3    ' Ok and Cancel buttons.
       FileErrors = 2
    Case Else
       FileErrors = 3
  End Select
End Function
*******************************************************************************
```

Routine to call the Help file

```
*******************************************************************************


Global previous_frm As String
Global PATH_NAME As Variant'Varient'= "C:\vbasic\"
Declare Sub WinHelp Lib "user" (ByVal hWnd%, ByVal HelpFile$, ByVal
HelpCommand%, ByVal HelpData As Any)
Global Const HELPKEYCODE = 112  ' F1
Global Const HELP_CONTEXT = 1, HELP_QUIT = 2, HELP_INDEX = 3
Global Const HELP_HELPONHELP = 4, HELP_SETINDEX = 5
Global Const HELP_KEY = &H101, HELP_MULTIKEY = &H201
Global Const HelpFileName = "c:\vb\hc\tex.hlp"
Const winColorPalette = 21004  ' Define constants.
Sub Keydown (KeyCode As Integer, Shift As Integer, sess As Integer)
 If (sess = 0) Then
  If KeyCode = HELPKEYCODE Then
     If Len(HelpFileName$) Then
       WinHelp start_frm.hWnd, PATH_NAME & HelpFileName$,
 HELP_INDEX, 0&
     Else
```

```
        MsgBox "Online help is not available inside the Visual Basic
environment.", 64, "VB Bug!"
      End If
      Exit Sub
    End If
  Else
  If KeyCode = HELPKEYCODE Then
      If Len(HelpFileName$) Then
        WinHelp start_frm.hWnd, PATH_NAME & HelpFileName$,
HELP_INDEX, 0&
      Else
        MsgBox "Online help is not available inside the Visual Basic
environment.", 64, "VB Bug!"
      End If
      Exit Sub
    End If
  End If
End Sub
**********************************************************************************
```

Simplex algorithm routine

```
**********************************************************************************
Global coef_array() As Single
Global fn_var(4) As Single
Global flag As Integer
Global c2() As Single
Global a() As Single
Global dupe() As Single
Global r1, c1 As Integer
Global c As Single
Global g_var  As Integer
Global check_flag As Integer
Const PATH_NAME = "c:\vbasic\project"

Sub equa ()

Dim x1, nw, ch As Integer
For i = 2 To 4
 fn_var(i - 1) = flag + i - 1
 Next i
x1 = 1
nw = 1
func1    'subroutine
Do While x1 = 1
   ch = 0
   For i = 2 To flag + 4
   If (a(6, i) <= 0) Then
    ch = ch + 1
   End If
  Next i
   If (ch = (flag + 3)) Then
   Exit Do
   End If
 func2  ' calling subroutine
 For i = 2 To flag + 6
  c2(r1, i) = a(r1, i)
 Next i
 For i = 1 To 6
  For j = 1 To flag + 6
   a(i, j) = c2(i, j)
   Next j
  Next i
  nw = nw + 1
If nw > 15 Then
MsgBox "Sorry! Cannot process data ,Try again"
   Exit Do
   End If
   func1
Loop
End Sub
Sub func1 ()
Dim k, i, y As Integer
 k = 2
 c1 = 2
 Do While k < flag + 6
  c = 0
    For i = 2 To 4
     c = c + a(i, 1) * a(i, c1)
    Next i
    a(5, c1) = c
    a(6, c1) = c - a(1, c1)
    k = k + 1
    c1 = c1 + 1
   Loop
    c1 = 2
    c = a(6, 2)
      For i = 3 To flag + 4
       If (a(6, i) > c) Then
        c = a(6, i)
        c1 = i
       End If
  Next i
   For i = 2 To 4
  If (a(i, c1) = 0) Then
    a(i, flag + 6) = 5000
  Else
    a(i, flag + 6) = a(i, flag + 5) / a(i, c1)
  End If
  Next i

  r1 = 2
  c = 1000
   For i = 2 To 4
   If (a(i, flag + 6) < 0) Then
                y = y + 1
    Else
    If (c > a(i, flag + 6)) Then
     c = a(i, flag + 6)
     r1 = i
    End If
  End If
   Next i
End Sub


Sub func2 ()
Dim i, j As Integer
 c = a(r1, c1)
 a(r1, 1) = a(1, c1)
 fn_var(r1 - 1) = CSng(c1 - 1)
  For i = 1 To 6
  For j = 1 To flag + 7
   c2(i, j) = a(i, j)
  Next j
  Next i
  For i = 2 To 4
  For j = 2 To flag + 5
   If i <> r1 Then
   c2(i, j) = a(i, j) - (a(r1, j) * a(i, c1)) / c
   End If
  Next j
  Next i
  For i = 2 To flag + 5
   a(r1, i) = a(r1, i) / c
  Next i
 End Sub
```

BIBLIOGRAPHY

@ Instrumental Colour Measurements and Computer Aided Colour Matching For Textiles by H.S. Shah and R.S.Gandhi, Mahajan Books.

@ Optimization theory and applications by S.S. Rao,Wiley Eastern.

@ Numerical Algorithms by E.V.Krishnamurthy and S.K.Sen, East West Press.

@ Microsoft Visual Basic -Programmers Guide by Microsoft Corporation.

@ Visual Basic Power Programming by Naveen Shammas.

@ Secrets of Visual Basic for Windows Master by Gary Entsminger, SAMS publishing