Evolving Colors in User Interfaces by Interactive Genetic Algorithm

Cosimo Birtolo*, Paolo Pagano*, and Luigi Troiano†

* *Poste Italiane S.p.A. – TI - SSI*
Centro Ricerca e Sviluppo
80133 Napoli, Italy
{birtoloc,paganop5}@posteitaliane.it

† *University of Sannio*
Department of Engineering
82100 Benevento, Italy
troiano@unisannio.it

***Abstract*—Color selection in designing user interfaces is addressed by an Interactive Genetic Algorithm. The proposed approach is aimed at finding the optimal trade-off between different and sometimes conflicting constraints, without any explicit model of user preferences and abilities. Experimentation investigates the algorithm convergence under several conditions and user behavior.**

***Keywords*-Accessibility; User Interface Design; Interactive Genetic Algorithms; Color Vision; Search Based Software Engineering**

I. Introduction

Colors are probably the most effective means by which concepts, ideas and emotions are evoked. Each person answers to color in a particular way, as cultural and individual traits play a relevant role in how the real world is perceived and experienced by vision. So, if on one side there are colors providing joy and relax, there are other colors producing excitement or melancholy in other cases. The selection of appealing and harmonic color combinations generally attains to artistic abilities of humans. Therefore, choosing an appropriate color palette is a relevant and common issue in designing graphical user interfaces. Besides aesthetics and psychology, adopting colors generally entails requirements regarding for example (i) high luminance contrast between related colors (e.g. foreground and background) in order to improve readability, (ii) preservation of chromatic choices and preferences as expressed by interface designers (e.g. to preserve the meaning of colors), and (iii) color accessibility to a broader audience (e.g color vision impaired users). These requirements can often be conflicting. Solving conflicts means often to experiment several combinations in order to find a suitable trade-off between different chromatic choices. Thus, finding an appropriate color palette can be regarded as a combinatorial optimization problem. Genetic algorithms are able to assist in solving the problem, due to their capability of exploring large combinatorial search spaces, exploiting solutions that reveal a good fitness to a set of requirements.

In previous studies [1] the application of genetic algorithms has been tested in order to explore the palette space, and to automatically identify alternatives to be suggested to the designer. However, humans still play a relevant role in driving the selections process, as their preferences cannot always be made explicit. User interactions is also able to overcome color vision deficiencies, such as protanopia and deuteranopia, and other disorders when no model is available or suitable to some users. For instance, users affected by cataract, making vision cloudy, differ quite a lot from one individual to each other. Building a generic model for this kind of disorders is not feasible. In these cases user's preferences become essential in order to make a well-designed and accessible interface. Involving users in the process becomes valuable also when some color usability aspects have not been taken into account at the initial color selection, or in adapting colors to being used by interface in different contexts from the original they have been chosen for.

Interactive Genetic Algorithms (IGA) can provide a meaningful approach in order to integrate human and artificial intelligence. This paper addresses the problem of evolving a given color palette towards a solution with a sufficient contrast between related colors, even when colors are not correctly perceived by impaired users. At the same time, the algorithm preserves user preferences and the initial hues. Differently by previous work [1], in this paper preferences and disorders are not made explicit by models, but are considered by including users directly in the evolution loop.

Unlike to traditional evolutionary algorithms that do not include the user in the loop, one of the daunting challenges of IGAs entails the effective methods of combating user fatigue in collecting his/her feedback. Indeed, according to [2], user fatigue is a critical element to produce high quality

978-1-4244-5612-3/09/$26.00

results. Evaluating solutions until convergence can lead to tedious and demanding attention periods on the user side. For this reason, we adopted clustering and interpolation techniques as a means for reducing the user fatigue during the algorithm evolution. The user attention is focused on only few but representative individuals, therefore a larger number of solutions can be explored and evolved.

The remainder of this paper is organized as follows: Section 2 briefly overviews the color modeling; Section 3 provides some basics of IGA, with a focus on their role in designing user interfaces; Section 4 describes the algorithm structure and details as used in our work; Section 5 reports experimental results; Section 6 outlines conclusions and future directions.

II. Color Modeling

Color palettes are arrays of colors. Several color models exist, each aimed at describing colors as tuples of numbers (typically three or four values), called color components. RGB and CMYK are well known color models. In RGB, a color is described by three primary components R=red, G=green, B=blue. A color is obtained by additively combining intensities of the primary components. To better address the human perception of colors, in 1976 the French *Commission Internationale de l'Elcairage* (CIE) introduced the CIELab model. In this model, color components are $L*$ that is a measure of color luminance, $a*$ being its position of red/magenta and green, and $b*$ its position between yellow and blue. Uniform variations of components in the CIELab model aim at corresponding to uniform variations in the perception of colors. Therefore, this color model is suitable for measuring the perceptual distance between colors by means of the Euclidian distance ΔE between points in $L* \times a* \times b*$, that is

$$\Delta E = \sqrt{(L*_1 - L*_2)^2 + (a*_1 - a*_2)^2 + (b*_1 - b*_2)^2} \quad (1)$$

The maximum distance ΔE^* is between green and blue values. Distance ΔE provides a measure of both hue and density changes. In perceiving colors, a key metrics is the contrast between related colors. The W3C's WCAG [3] defines the *contrast ratio* as

$$C = \frac{\max(L_1, L_2) + 0.05}{\min(L_1, L_2) + 0.05} \quad (2)$$

where L is the *relative luminance*. Relative luminance is defined as the relative perceived brightness of any point, normalized to 0 for black and 1 for maximum white. We notice that relative luminance L as defined by W3C's WCAG differs from luminance $L*$ defined in CIELab. Contrast ratios can range from 1 to 21 (commonly written 1:1 to 21:1). According to W3C to reach level AAA of accessibility:"*text (and images of text) must have a contrast ratio of at least 7:1, except if the text is pure decoration. Larger-scale text or images of text can have a contrast ratio of 5:1*". This means that solution utility passing the contrast ratio threshold T_c is $C_u = 1$, decreasing below T_c. Therefore we adopted: RGB, as a standard means for describing the palette colors; CIELab, for measuring the distance and contrast between colors; CIEXYZ[1], in order to project RGB values into CIELab space.

III. Interactive Genetic Algorithms in Designing User Interfaces

Designers usually adopt guidelines to organize the layout and the features of user interface. Recent developments use meta-heuristic and evolutionary techniques to organize structural elements of interface [4]–[6] or non structural features such as colors [1]. Ichikawa et al. [7] describe the re-working of Web pages for color-deficient viewers. The authors investigated a genetic algorithm able to improve the image contrast as perceived by simulation of impaired users, but still preserving the image chromaticity. To reach their goal, they first decompose the page into a hierarchy of colored regions. These spatial relations determine important pairs of colors to be modified. Then they minimize the fitness function using a genetic algorithm. In a previous work [1] we confirmed Ichikawa et al. results proving the advantages in using a genetic algorithm to optimize a color palette with a good trade-off between aesthetics and accessibility requirements.

In this work we introduce the user into the evolution process, as described by Takagi in Interactive Evolutionary Computation [8]. In particular we develop an IGA in order to include human preferences and knowledge into a Genetic Algorithm (GA). Indeed IGAs provide a mapping from a users psychological space to a GAs parameter space and thereby combine the power of human subjective evaluation with evolutionary computation [8].

In literature, recent works use IGAs for user interface design, for instance Quiroz et al. [5], [9] encode user interfaces as individuals and run through a number of generations in order to support the exploration of user interface option space. Oliver et al. [6] focus their attention on the appearance and layout of Web sites. The user drives the evolution of style and layout of a Web pages through an IGA by picking solutions he/she prefers. Amit Banerjee et al. [10] describe a computational model for creative design based on collaborative interactive genetic algorithms, and present an implementation for evolving creative floorplans and widget layout/colors for individual UI panels.

Jia-Bin Huang et al. [11] propose a fast algorithm that automatically enhances the color contrast for CVD viewers through a histogram transformation approach. They also

[1]CIE introduced the XYZ model in 1931. Despite its age, it is still widely used in practice, especially as a reference for converting colors from one model into another. Similarly to RGB, XYZ adopts a system of additive primary components, namely X, Y, and Z. Each of these components represents the power perceived when RGB primaries are emitted.

provide a control parameter for users to specify the degree of enhancement. Their algorithm can run at real-time, and simulation results have demonstrated the effectiveness and efficiency of the algorithm.

Note that for an IGA-based user centric application, we could not rely on large populations sizes and hundreds of generations (like a canonical GA) because it is unrealistic to request a user to make hundreds or thousands of choices. If pressed for too much feedback, users are likely to lose interest and get tired [2]. In a recent work of Shackelford [12] the issues of delivering interactive algorithm to real world users is addressed, e.g. considering automated tester to avoid some user interactions.

IV. Algorithm Structure and Fitness Function

Users differ for age, education, skills, gender, and vision disorders. Therefore data cannot be used directly for evolving the interface, if we are interested to adapt the interface to specific needs and preferences.

According to the work of Quiroz [5], in order to minimize user interaction preserving the user effort and also to keep an objective component in evaluating solutions, we do not demand the user to provide a feedback at each generation, but only every F generations, as shown in Fig.1. An initial feedback is collected just after the algorithm starts in order to drive the evaluation of individuals since the beginning. Even requiring the user feedback every F generations, the large number of individuals does not make feasible to fully evaluate a population. To overcome interaction limitations, we evaluate individuals indirectly comparing them to few samples given to the user for feedback. Samples are chosen as representative of the whole population, thus clustering provides a natural means for this task. Clustering is aimed at partitioning the population into a fixed number k of classes each of those being represented by an average color palette. The average individual associated to each class can be obtained by different criteria. Examples are the mean, the most likely and the nearest to the mean. Each individual is then evaluated according to the score assigned to each class. In particular we adopted an Expectation-Maximization (EM) algorithm [13]. In EM clustering, the algorithm iteratively refines an initial cluster model to fit the data and determines the probability that a data point belongs to a cluster. The algorithm terminates when the probabilistic model well fits the data. The function used to measure how the model fits the data is the log-likelihood. In more details, the EM algorithm consists of repeating two steps until convergence:

- E-Step: estimates the Expected value of unknown variables, given the current parameter estimation
- M-Step: re-estimates the distribution parameters in order to Maximize the data likelihood, given the expected estimation of unknown variables.

The result of EM clustering is probabilistic: unlike K-Means clustering and similar, where each data point can belong only to one cluster at a time, in EM algorithm a data point always belongs to multiple clusters with different probability.

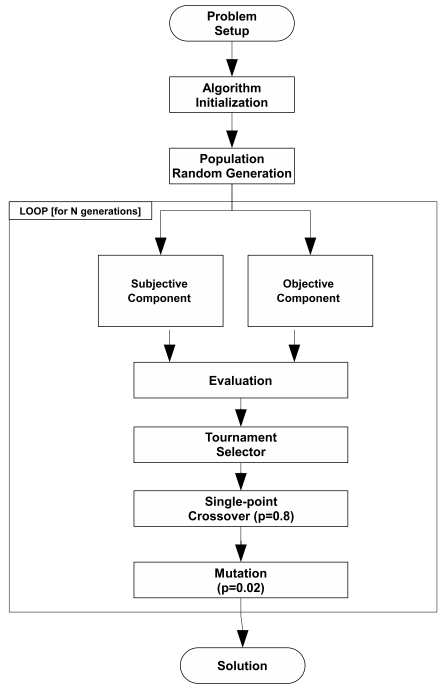

Figure 1. The Interactive Genetic Algorithm

In Fig.1 the structure of the proposed genetic algorithm is illustrated. In particular individual evaluation is made by two factors: objective component and subjective component.

Objective component, whose equation is provided in Eq.4, keep into account some objective metrics about the color palette such as luminance contrast among related colors (e.g. foreground and background) and initial chromatic choices planned by interface designers which are already considered in the genetic algorithm proposed in a previous work [1].

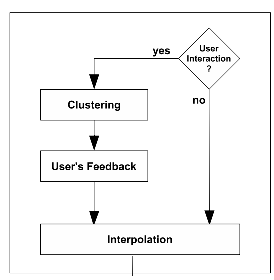

Figure 2. Subjective Component

Subjective component introduces the user's preferences according to theoretical discussion argued by Takagi [8] and to conclusion outlined by Quiroz in a recent example of application [9]. As shown in Fig.2, when the user interaction is invoked every F generations, clustering identifies k average individuals representative of the entire population. After the user feedback is collected for scoring those individuals (see Eq.7). The evaluation of the current population is given by interpolating these scores as the user expected value attributed on the basis of cluster average individuals.

Our work definitively differ from the literature for exploiting a different way to collect user feedback and investigated in the experimental results. In other words, the proposed algorithm searches for a trade-off among contrast ratio/accessibility requirements, user aesthetics/preferences and the original palette.

A. Chromosome

The RGB model for representing colors having been chosen, the palette chromosome coding is straightforward, as depicted in Fig.3. In particular, the chromosome is a bit string, reserving 24 bits (8 bits per component) to represent each color in the RGB space.

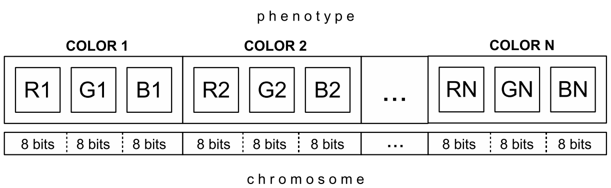

Figure 3. Chromosome structure and phenotype mapping.

B. Fitness Function

Our aim is to find a palette that improves the luminance contrast between related colors, while still preserving the original chromatic setting and awarding palettes which best fit the user's preferences.

For these reasons, the fitness function is made by two components: (i) a preference factor depending on some subjective user evaluations of the interface colors in order to capture the user preferences along the evolution process; and (ii) a colorimetric factor as a function of some metrics aimed at evaluating the *contrast ratio* [1], [3] between related colors and the distance from the original color selection. As we aim at trading off the two components, we decide for a weighted geometric mean. The fitness function of individual x is defined as

$$fitness(x) = f_c(x)^{1-\omega} \cdot f_s(x)^{\omega} \tag{3}$$

where ω is the weight assigned to the subjective component f_s, and $1-\omega$ is the weight of the objective component f_c. Both components are scores belonging to the unit interval $[0,1]$, thus geometric mean enforces them to be higher than arithmetic weighted mean, as in general it provides lower values.

The objective component is defined as geometric mean

$$f_c(x) = \left(\prod_{i=1}^{n} (1 - d_i(x)) \prod_{j=1}^{m} c_j(x) \right)^{\frac{1}{n+m}} \tag{4}$$

where d_i is the distance of color i from the corresponding original one, c_j the contrast ratio of the j-th pair of related colors. We assume to have n colors and m pairs of related colors.

$$d_i(x) = \frac{\Delta E_i(x)}{\Delta E^*} \tag{5}$$

$$c_j(x) = \frac{20 + \min(C_j(x) - T, 0)}{20} \tag{6}$$

where ΔE_i is the distance of color i from original one as defined in Eq.1, C_j is the contrast ratio as defined in Eq.2 and T is the contrast threshold (i.e. 7 or 5) as recommended by W3C's guidelines.

On the other side, the subjective component aims at providing an estimation of score the user would assign to color palette of the target interface. Therefore, it is defined as expectation value of scores attributed by the user to cluster average individuals.

$$f_s(x) = E[S(x)] = \sum_{i=1}^{k} Pr(x \in H_i) \cdot S_i \tag{7}$$

where S is the set of scores collected by user feedback, H_i is the i-th cluster class, and S_i is the related score.

The maximum fitness value is $fitness(x) = 1$, but this is ideal as only reachable when $c_j(x) = 1$, $d_i(x) = 0$ for all $i = 1..n, j = 1..m$. In addition, this individual has to be evaluated by the user as the best possible ($f_s(x) = 1$), leading to the conclusion that the original palette is already optimal, thus not requiring any variation. In general, this value is below 1 as far as there is a need to modify colors ($d_i(x) > 0$ for some i), or some contrast ratios are not satisfactory ($c_j(x) < 1$ for some j), or the solution does not satisfy the user ($f_s(x) < 1$).

V. Experimental Results

Testing the performance of IGAs on quantitative basis is not an easy task. Considering humans in the loop entails a limitation in the number of trials. In addition inconsistencies of user behavior can affect the algorithm convergence and results. Therefore we followed a different testing strategy, making some simplifying assumptions and performing simulations by agents. We defined different agents in order to test the algorithm emulating different two different user behaviors, specifically:

- $AgentCBY$ prefers palettes containing at least one color similar to blue, and one to yellow in a consistent way;
- $AgentNBY$ rewards palettes not containing both blue and yellow, thus entailing a behavior opposite to $AgentCBY$;

An agent evaluates each i-th (cluster) average individual assigning a score S_i to it when the user feedback is invoked.

$AgentCBY$ provides a score to the palette containing blue and yellow according to distance defined by Eq.8.

$$S_i = (1 - d_{blue}) \cdot (1 - d_{yellow}) \tag{8}$$

where d_{blue} and d_{yellow} are defined as

$$\begin{aligned} d_{blue} &= \min\{d_{b_1}, d_{b_2}, ..., d_{b_n}\} \\ d_{yellow} &= \min\{d_{y_1}, d_{y_2}, ..., d_{y_n}\} \end{aligned} \tag{9}$$

and d_{b_j} and d_{y_j} are respectively the distance of color j from the intended blue and yellow (see Eq.5). Therefore $AgentCBY$ assigns higher score to palettes having a lower distance from yellow and blue.

As distance scores can slightly vary, we adopted an emphasizing function $g : [0,1] \rightarrow [0,1]$ aimed at increasing the score differences. Thus, we actually considered

$$S_i = g(1 - d_{blue}) \cdot g(1 - d_{yellow}) \tag{10}$$

where

$$g(x) = \begin{cases} \frac{(2x)^\alpha}{2} & x \leq 0.5 \\ 1 - \frac{(2(1-x))^\alpha}{2} & x > 0.5 \end{cases} \tag{11}$$

and plotted in Fig.4. The parameter α controls the emphasizing degree of g, higher by increasing values of α. We notice this function mostly emphasizes score differences around 0.5 as we are not very interested to relative differences when the score is high or low.

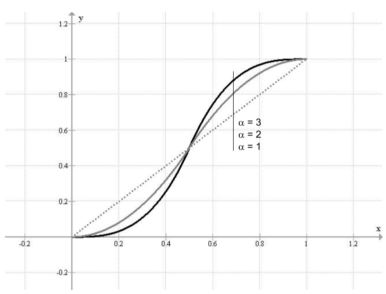

Figure 4. Emphasizing function $g(x)$.

$AgentNBY$ and $AgentCBY$ use the same mathematics, although they apply to two different preference models.

In our experimentation, the agent feedback is collected on 5 average individuals that are palette color representatives, obtained by EM clusters. The algorithm uses 1000 generations on populations made of 500 individuals. Charts refer to the best individual fitness series closest to the median and best individual mean of 10 different runs. The feedback is collected every F generations (with F equal to 10, 20, 50, 100, 200). In addition we tested the algorithm in two limit conditions: (i) when the algorithm requires the user feedback at each iteration (i.e. $F = 1$) and (ii) when the feedback is never invoked but including the preference explicitly in the fitness function thus considering a canonical non-interactive genetic algorithm [14]. In particular Fig.5 [2] outline the behavior of $AgentCBY$, while Fig.6 refer to $AgentNBY$. In both cases we notice the user feedback provides a step to fitness when it occurs, as clustering is able to focus on better individuals over the generations. The feedback rate affects the algorithm convergence, as the more frequently the feedback is provided, the more accurately the algorithm is driven by the user. The best result is obtained by including the user preferences explicitly in the fitness value of each individual (case NOF), as evaluation is more fine-grained. However, with less frequent feedback the algorithm was still able to converge towards a suboptimal solution.

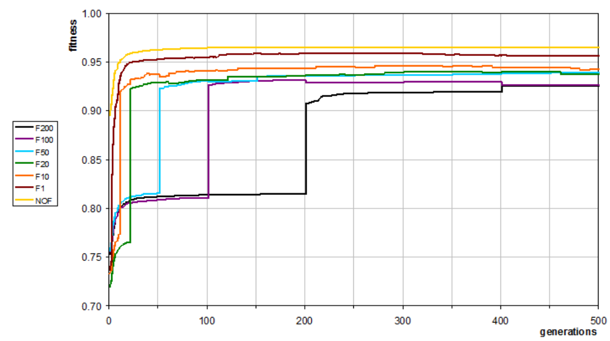

Figure 5. $AgentCBY$: Fitness' trend of mean Best Individual of 10 different runs at varying F .

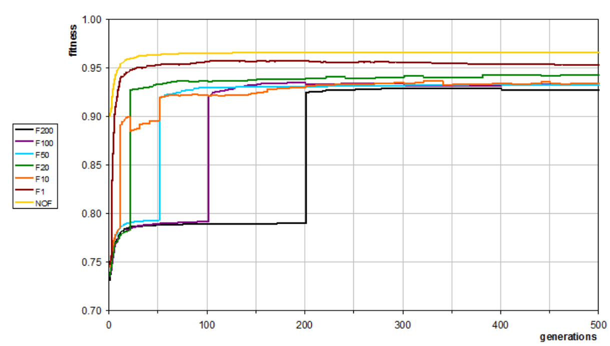

Figure 6. $AgentNBY$: Fitness' trend of mean Best Individual of 10 different run at varying F.

In order to analyze the different behavior of the algorithm at different feedback rates $r = 1/F$, we performed a Kruskal-Wallis test on the fitness samples obtained at the end of evolution (i.e. at generation 1000). The test p-value (1.204e-06 when NOF is considered, 4.566e-03 when NOF is not in the set) rejects the null hypothesis that fitness results belong to the same population at confidence level of 99%. Investigating this results more deeply, we also

[2]For the sake of simplicity we plotted only the first 500 generations

considered a pairwise (two-sided) Wilcoxon rank-sum test, reporting results in Table I. We gain that, although we cannot assume the fitness output is not affected by the interaction rate in general, similar outputs can be obtained at some different rates. For instance, when the output got at rates $F10$,$F20$,$F50$, the fitness samples are not statistically different (p-value ranging between 1.088e-01 and 6.020e-01), as well as those at rates $F100$ and $F200$ (p-value 4.853e-01). But even considering the whole group $F10$-$F200$ there is no statistical difference, although this conclusion is weaker (minimal p-value 2.163e-02). Obviously the result at $F1$ is slightly different but infeasible, and output tend to be slower at smaller interaction rates.

	NOF	$F1$	$F10$	$F20$	$F50$	$F100$	$F200$
NOF	-	5.413e-06	5.413e-06	5.413e-06	5.413e-06	5.413e-06	5.413e-06
$F1$	5.413e-06	-	1.237e-01	9.272e-03	1.943e-03	7.523e-04	2.436e-04
$F10$	5.413e-06	1.237e-01	-	2.179e-01	1.088e-01	3.763e-02	2.163e-02
$F20$	5.413e-06	9.272e-03	2.179e-01	-	6.020e-01	2.644e-01	9.516e-02
$F50$	5.413e-06	1.943e-03	1.088e-01	6.020e-01	-	8.275e-02	1.237e-01
$F100$	5.413e-06	7.523e-04	3.763e-02	2.644e-01	8.275e-02	-	4.853e-01
$F200$	5.413e-06	2.436e-04	2.163e-02	9.516e-02	1.237e-01	4.853e-01	-

Table I
$AgentCBY$: P-VALUES.

The same tests applied to $AgentNBY$ provided similar results as both agents consistently keep their set of preferences, although in opposite manner.

Still in this case, Kruskal-Wallis test rejected the null hypothesis that there is no statistical difference in the resulting fitness, with p-value being 7.339e-06 when 7 different feedback rates are considered (NOF,$F1$,$F10$,$F20$,$F50$,$F100$,$F200$) and 3.233e-02 when NOF is not take into the account. Furthermore pairwise Wilcoxon rank-sum test leads to conclusions similar to those outlined for $AgentCBY$ as reported in Table II.

	NOF	$F1$	$F10$	$F20$	$F50$	$F100$	$F200$
NOF	-	1.624e-04	5.413e-06	5.413e-06	5.413e-06	5.413e-06	5.413e-06
$F1$	1.624e-04	-	1.773e-02	3.151e-02	3.421e-03	2.598e-03	7.523e-04
$F10$	5.413e-06	1.773e-02	-	7.106e-01	6.020e-01	3.697e-01	3.697e-01
$F20$	5.413e-06	3.151e-02	7.106e-01	-	1.965e-01	1.965e-01	1.237e-01
$F50$	5.413e-06	3.421e-03	6.020e-01	1.965e-01	-	4.853e-01	3.421e-01
$F100$	5.413e-06	2.598e-03	3.697e-01	1.965e-01	4.853e-01	-	3.421e-01
$F200$	5.413e-06	7.523e-04	3.697e-01	1.237e-01	3.421e-01	3.421e-01	-

Table II
$AgentNBY$:P-VALUES.

We also experimented in Fig.7 the algorithm by varying the weight of objective and subjective fitness components (i.e. $\omega = \{0.2, 0.5, 0.8\}$).

A. Example of Application

As an example of application we can consider an initial palette of n colors c_n and the proposed algorithm provides a final accessible palette as a solution. Starting with a set of 6 colors we optimize the colors according a particular model

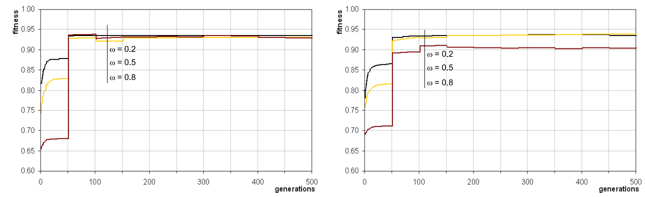

Figure 7. Best individual fitness series closest to the median (on the left) and best individual mean (on the right) of 10 different runs at varying ω.

of relation among colors. In particular the the first color (c_1) is related to the second (c_2) and third (c_3) color, whilst the fourth (c_4) to the fifth (c_5) and the sixth (c_6).

The genetic algorithm is setup with following parameters[3]: generation = 1000, population = 500, feedback rate $r = 1/50$, tournament = 1, crossover probability = 0.8, mutation probability = 0.02, elitism = 5.

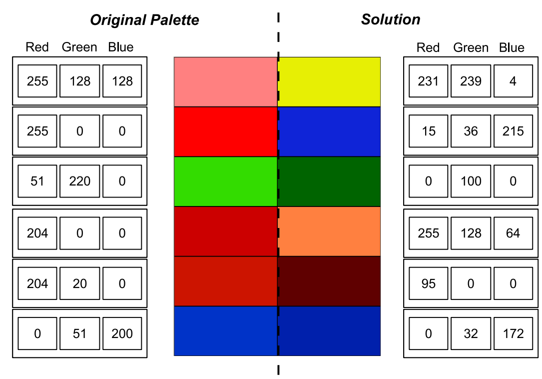

Figure 8. Qualitative Analysis with $\omega = 0.2$: Color Palette.

In Fig.8 we present the outputs of a run of algorithm with ω levels equal to 0.2. In the figure the RGB components and the relative colors are so expressed: on the left, the RGB components of the original palette made by 6 colors c_i (with $i = 1...6$), while on the right the solution found g_i (with $i = 1...6$). We observe how (c_1, c_2, c_3), with an initial contrast ratio is $1.6 : 1$ between c_1 and c_2 and $1.3 : 1$ between c_1 and c_3, evolves into (g_1, g_2, g_3) entailing contrast ratio of $7.6 : 1$ and $5.9 : 1$ respectively. Furthermore c_4, which has a contrast ratio of $1.01 : 1$ with c_5 and $1.5 : 1$ with c_6, evolves into g_4, g_5 and g_6 with a contrast ratio of $5.7 : 1$ and $4.8 : 1$ respectively.

In Fig.8 and in Fig.9 a lower value of ω (see Eq.3) emphasizes the subjective component (Eq.7). Indeed, the output illustrated in Fig.9 outlines how user's preferences, simulated by $AgentCBY$, are properly satisfied as the color

[3]Structural parameters have been chosen by a simple qualitative analysis, according to common values adopted for them and to previous experimentation [1], while *interaction parameters* (such as subjective the component weight ω and the feedback rate r) are in-depth analyzed in the experimentation.

Figure 9. Qualitative Analysis with $\omega = 0.2$: Interface.

c_1 is transformed in g_1 and color c_2 in g_2, becoming similar to the intended yellow and blue.

Although the initial palette is unreadable for most of the users, thanks to the proposed algorithm the palette not only satisfies accessibility requirements, but also finds a right trade-off among W3C's requirements, chromatic choices as originally made by the interface designer and user preferences.

VI. Conclusion

We presented an approach that explores the color palette space, searching for a solution that provides a good trade-off between aesthetics and accessibility requirements. This search is performed by using an IGA that combines both computable metrics and human subjective feedback to drive the choice of an appropriate set of colors in a Graphical User Interface. Experimental results proved this approach to be feasible and advantageous, with respect to non-interactive genetic algorithms. This is due to the IGA ability of capturing fitness attributes related to human perception that cannot be acquired by mathematical modeling. In order to make analysis robust and independent on the human perception, we used agents to simulate an expected behavior. Analysis showed that all IGAs were able to reach a consistent result, independently on the interaction rate. This suggests that lesser frequent interactions are desirable on more frequent ones, as each interaction is expensive due to the user fatigue in evaluating a large number of solutions. In this paper we grouped individuals by EM clustering, and asking the user to evaluate a small number of solutions obtained. Future directions are aimed at experimenting the algorithm with users with a particular color vision deficiencies and with different skills. Moreover we want to analyze the color symbolism related to the focus of user interface.

References

[1] L. Troiano, C. Birtolo, and M. Miranda, "Adapting palettes to color vision deficiencies by genetic algorithm," in *GECCO '08: Proceedings of the 10th annual conference on Genetic and evolutionary computation*. New York, NY, USA: ACM, 2008, pp. 1065–1072.

[2] X. Llorà, K. Sastry, D. E. Goldberg, A. Gupta, and L. Lakshmi, "Combating user fatigue in igas: partial ordering, support vector machines, and synthetic fitness," in *Proc. GECCO '05*. New York, NY, USA: ACM, 2005, pp. 1363–1370.

[3] "Web Content Accessibility Guidelines (WCAG) 2.0," W3C, Tech. Rep. 11, December 2008.

[4] L. Troiano, C. Birtolo, R. Armenise, and G. Cirillo, "Web form page in mobile device: Optimization of layout with a simple genetic algorithm," in *Proc. of 11th Int. Conf. on Enterprise Information Systems*. ICEIS, 2009.

[5] J. Quiroz, S. Louis, A. Shankar, and S. Dascalu, "Interactive genetic algorithms for user interface design," in *Evolutionary Computation, 2007. CEC 2007. IEEE Congress on*, Sept. 2007, pp. 1366–1373.

[6] A. Oliver, N. Monmarch, and G. Venturini, "Interactive design of web sites with a genetic algorithm," in *Proc. of the IADIS Int. Conf. WWW/Internet*, November 2002, pp. 355–362.

[7] M. Ichikawa, K. Tanaka, S. Kondo, K. Hiroshima, K. Ichikawa, S. Tanabe, and K. Fukami, "Web-page color modification for barrier-free color vision with genetic algorithm." in *GECCO*, ser. LNCS, Erick Cant-Paz et al., Ed., vol. 2724. Springer, 2003, pp. 2134–2146.

[8] H. Takagi, "Interactive evolutionary computation: fusion of the capabilities of ec optimization and human evaluation," *Proc. of the IEEE*, vol. 89, no. 9, pp. 1275–1296, Sep 2001.

[9] J. Quiroz, S. Louis, A. Banerjee, and S. Dascalu, "Towards creative design using collaborative interactive genetic algorithms," in *Evolutionary Computation, 2009. CEC '09. IEEE Congress on*, May 2009, pp. 1849–1856.

[10] A. Banerjee, J. C. Quiroz, and S. J. Louis, "A model of creative design using collaborative interactive genetic algorithms." in *Proceedings of the Third International Conference on Design Computing and Cognition. DCC08*. Springer Netherlands, 2008, pp. 397–416.

[11] J.-B. Huang, S.-Y. Wu, and C.-S. Chen, "Enhancing Color Representation for the Color Vision Impaired," in *Workshop on Computer Vision Applications for the Visually Impaired*. Marseille France: J. Coughlan and R. Manduchi, 2008.

[12] M. R. N. Shackelford, "Implementation issues for an interactive evolutionary computation system," in *Proc. of GECCO '07*. New York, NY, USA: ACM, 2007, pp. 2933–2936.

[13] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 39, no. 1, pp. 1–38, 1977.

[14] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison Wesley, 1989.