Designing a Website using a Genetic Algorithm

Lukas Günthermann[1] and John Kingston[2]

[1] formerly of School of Computing, Engineering and Mathematics, University of Brighton BN2 4GJ

[2] Centre for Secure, Intelligent and Usable Systems, University of Brighton BN2 4GJ
`j.k.kingston@brighton.ac.uk`

Abstract. This paper describes the use of a genetic algorithm to design a website, according to principles of clarity, symmetry, golden ratio and image size. The website's logo is used to calculate a matching colour scheme. Results indicate that local maxima can be a problem but that with the right weighting of the fitness function, a pleasing design can be achieved.

Such a program could be used when designing large numbers of websites; when a website has to be re-designed regularly to match changing content; or to provide a starting point for human website designers or users of interactive genetic algorithms to improve.

Keywords: Genetic algorithms; Aesthetics; Design; Websites; Configuration

1 Introduction

Aesthetics is the subsection of philosophy examining the nature of beauty and taste. Lavie and Tractinsky [1] divides it into classical aesthetics and expressive aesthetics. Classical aesthetics describes characteristics such as clarity, orderliness, and symmetry; the traits of expressive aesthetics are being original, fascinating, refined, or creative.

The perception of aesthetics in the design of a website influences how likely visitors are to trust it and even how good they perceive the usability of the website to be [2]. Since the majority of online users give a website only a few moments of attention [3] and do not browse deeply to subpages [4] it is necessary to gain the viewer's attention and trust within this short time span.

The goal of the work described in this paper is to produce a website design using genetic algorithms that fits the principles of classical aesthetics and whose colours blend with the logo provided. The focus is placed on the logo because of research that suggests that companies can embed traits of credibility in the design of a logo and that these logos can trigger positive credibility assessments of the sponsor of the website [5]. If this is correct then a good logo fulfils both the need to gain attention and the need to gain trust. However, attention can produce positive or negative results; for logos to be visually appealing, they have to blend in with the design of the website.

This approach could be applied in situations, in which it is necessary to create or maintain many pages with similar requirements. It could be used in web services which offer a modular construction system for customers without experience in web design.

They would select the content and the settings, so that the algorithm can create a basic solution, which could then be improved by human designers if required.

Other researchers have applied genetic algorithms to the optimisation of website design (e.g. [6], [7], [8]). However, all these approaches rely on user feedback to select preferred designs while the genetic algorithm, is running. The work described in this paper uses pre-programmed design principles instead which allows the algorithm to run through many more iterations in a reasonable time.

2   Genetic Algorithms

The theory of evolution posits that reproduction within a species occasionally produces an individual with one or more mutations; that some of these mutations are beneficial to survival, causing the mutation to spread throughout the population according to the principle of survival of the fittest; and that an accumulation of such mutations eventually produces a new species. Genetic algorithms mimic the first two of these three steps within a computer system in an attempt to produce optimised or near-optimal results. Genetic algorithms are usually applied to optimisation problems which are too complex to be solved arithmetically in a reasonable amount of time due the high number of possible combinations (e.g. the well-known Travelling Salesman problem).

The core of a genetic algorithm is the population of chromosomes: units of information which usually consists of a bit string. Each locus in the chromosome has two possible alleles (0 and 1); this allows every chromosome in the population to represent one possible set of parameters. The goal is to develop a genotype (set of parameters) which transforms the source code into the fittest phenotype (in this case, the most appealing website).

According to Mitchell [9], there are three operators in the most basic version of a GA used to evolve the population and create stronger solutions:

Selection: The higher the estimated value of a chromosome is, the more likely it is that the chromosome is selected for reproduction. However, if there is a dominating chromosome in the population, the entire population will slowly segue into similar characteristics, reducing diversity and adaptability; so it is important not to focus only on the most successful members of the population, but to keep variety high and eliminate members which are similar to the dominating member.

Crossover: This operator takes two selected chromosomes as parents and combines their genes into offspring with the same genotypic structure. There are several methods to recombine the two parent chromosomes. The most frequently used one is **single point crossover** [10] which selects a random locus within a chromosome and then takes the subsequence before the locus of the first chromosome and merges it with the subsequence after the locus of the second chromosome to form a new chromosome for the population. The same happens with the two remaining subsequences. Other alternatives are **N-point crossover** (using multiple randomly selected break points to recombine the two strings; this may impair the performance by splitting up interdependent gene blocks, but it does allow the head and tail section of a chromosome to stay together; and **uniform crossover** where the source parent for each gene of the offspring is chosen by chance, which covers a larger part of the search space [11].

Mutation: To increase the diversity in the population, each bit has a small chance of being flipped. The probability should not be too high, otherwise the algorithm will turn into a random search [12].

Once the reproduction process has finished, the value of chromosomes is estimated by a fitness function, and the process is repeated based on the new generation. The new generation is usually of a similar size to the previous one, but sometimes only the best members of the previous generation are carried forward – this is known as elitist selection [13]. This prevents the performance of the best solution in the population from decreasing [14] but may also prevent the algorithm from overcoming a local maximum.

3   Fitness of a design

There is no objective way to describe beauty or good design in general, but we can evaluate a design according to classical aesthetic principles (clarity, orderliness and symmetry). Deciding how those features should be weighted requires drawing on fields such as psychology, cognitive science and neuroscience.

3.1   Gestaltung

According to [15] the first impression of a design influences all of the following perceptions. A user first views the whole Gestalt (German for "essence or shape of an entity's complete form") of the design and then starts noticing details.

The following Gestalt principles were suggested in [16]:

• Objects which are similar to each other in shape, size, colour, or texture will be perceived as part of a pattern.
• Lines or curves will lead the eye as a path and point attention towards breaking point like the end of the path or a crossing with another one.
• Viewers will subconsciously fill blanks and perceive structures as a whole, even if they are not closed.
• Simple and minimalistic designs reduce distraction.
• Similar elements can create the perception of combined objects, if they are close enough to each other. The eye tends to separate objects into background and foreground and shaping both in contrast to each other.
• A composition should provide order and balance, otherwise the viewer will feel unease and will not be able to decipher the content.

A fitness function could provide positive scores to simple and minimalist design; to order and balance; to paths that point attention to key elements (e.g. the logo); and perhaps to similarity of objects.

3.2   Page Layout

Three aspects of page layout are considered here: symmetry; the golden ratio; and images.

Symmetry. In [17] it is suggested that male users consider vertical symmetrical designed web pages more beautiful and appealing than asymmetrical ones. Surprisingly, the assessments by female participants seem not to be influenced by the factor whether a web page is symmetrical designed or not. Altaboli and Yin [18] confirm that the more similar the numbers and sizes of objects in two neighbouring areas are, the more appealing the visual aesthetics are perceived.

Golden Ratio. An asymmetrical design can be appealing as well, especially if it conforms to the *golden ratio*. The golden ratio is achieved when the ratio between two quantities is the same as the ratio between the bigger quantity and the sum of both quantities. The ratio is commonly believed to make a design more appealing to the human eye. Therefore, it is used, inter alia, in architecture, art, and music; the ancient Greek sculptor Phidias may have known of this ratio in 447 BC, when he created the sculptures for the Parthenon [19]. The golden ratio can also be found in nature, which might indicate that it offers some advantage in natural selection.

Images. Images are a critical part of a website as they help visitors to connect and feel comfortable [20]. They are also much more important than text because few visitors read all the text on any website.

Millennials favour websites with a large main picture; the users observed in [21] rated the visual appealing of such pages "significantly higher" compared to websites without such a main picture. Tullis and Tullis [22] showed that visual appeal ratings of e-commerce websites improved as the size of the largest image increased.

So a fitness function for page layout could evaluate symmetry; use of the golden ratio; and the size of the largest image.

3.3 Colour

When designing a website, it is important that the colour scheme matches the content, because it will determine in which way this content is perceived. The interpretation of colour depends on the social and cultural background of the viewer, some colours also create specific emotions: "Within the psychology of colors, warm colors show excitement, optimism, and creativity; cool colors symbolize peace, calmness, and harmony" [23].

The foundation of colour theory is the colour wheel as shown in **Error! Reference source not found.**. Complementary colours are colours which are on opposite sides of the wheel (e.g. yellow – violet). They create a high contrast when used in design. Analogous colours create a harmonious design by combining a colour and its neighbouring colours (e.g. red - red orange and red - red violet).
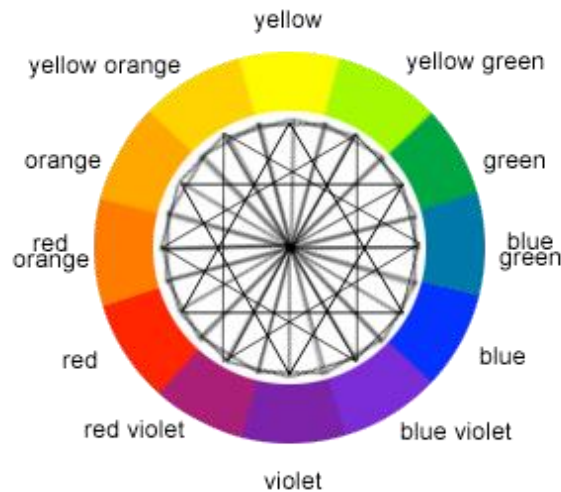

Fig. 1: The Colour Wheel

However, since this project aims to create a universal solution, it is not supposed to be optimised for a certain style or scheme. The nature of the colour theme is determined by the colour of the logo, which is given as an input variable. The fitness function will therefore evaluate the use of complementary colours to create a contrast between the main element and the background, and of analogous colours that harmonise with the logo.

4   Implementation

A solution was developed using Microsoft Visual Design Studio 2017 and compiled in C++14. The genetic algorithm was drawn from the GALGO 2.0 template library for constrained optimisation.

The objective function has a vector of parameters as input. This vector represents the decoded chromosome. The return value of this function is the calculated fitness of the individual. The GALGO algorithm uses this function to determine which chromosomes should be selected for crossover.

The algorithm's settings were:

• The first generation is created with random chromosome values based on lower and upper bounds of colours (0 to 256), column widths (0 to 800), logo and images sizes, and padding.

- Stochastic universal sampling selection was used as a selection method and uniform crossover as a crossover method, following [24]:
- It was decided that flipping bits in a space of 256 options was not useful, so uniform mutation was used.
- A mutation rate of 0.03 turned out to convey the best results.
- The elite population size to 1, so that only the best member of each generation is carried on to the next one, thus preventing the performance of the solution from decreasing.
- Experimentation with different population sizes and number of generations determined that a good set-up involved 200 chromosomes mutating over 100,000 generations.
- Different coloured versions of the logo were introduced to enable multiple test runs.

4.1   Graphic handling

The goal is for the header to have a matching colour with the logo. A simple method would be using the average colour value of the border pixels but this was infeasible because then a logo with blue and yellow pixels at the margin would result in a green header.

Therefore, the logo is processed with the CImg package before the genetic algorithm is run. The algorithm counts which border pixels occur the most. All RGB colours c_1 and c_2 within a tolerance range of 20 in the difference Dc are counted as one.

$$D_C = |c_1.red - c_2.red| + |c_1.green - c_2.green| + |c_1.blue - c_2.blue|$$

The library is also used to create the background image of the two columns in the main part of the website which is coloured according to the colour of the two columns and the background colour.

4.2   Web design

The program converts the values of the three RGB channels to single colours and calculates the size of website elements based on the corresponding genes and the general width of the page, which is set in the input variables (1024px was used).

The output is in the form of CSS code (Fig. 2) and HTML code (Fig. 3). The head and body of the HTML document are represented by a vector of nodes (Fig. 4). Each node contains a name which represents the start tag and a Boolean value which determines whether the result *string* includes a closing tag or not. Optional components are attributes which will be included in the start tag; content which will be added between the start and end tag; and other nodes which enable the creation of a nested structure.

```
css += "#header {\n";
css += " text-align: center;\n";
css += " width: " + std::to_string(this->pageWidth - 2u * this->_padding) + "px;\n";
css += " padding: " + std::to_string(this->_padding) + "px;\n";
css += " height: "+ std::to_string(this->_heightLogo) + "px;\n";
css += " margin: " + std::to_string(this->_padding) + "px 0px " + std::to_string((int)(0.5 * this->_padding)) + "px 0px;\n";
css += " background: rgb(" + this->_colourTitle.toString() + ");\n";
css += "}\n";
css += "#header img {\n";
css += " float: left;\n";
css += " width: " + std::to_string(this->_widthLogo) + "px;\n";
css += " height: " + std::to_string(this->_heightLogo) + "px;\n";
css += "}\n";
css += "#header h1 {\n";
css += " position: absolute;\n";
css += " text-align: center;\n";
css += " line-height: " + std::to_string(this->_heightLogo) + "px;\n";
css += " left: 0;\n";
css += " right: 0;\n";
css += "}\n";
```

Fig. 2: CSS settings of the 'header' element

The fitness function mostly gives out penalties for aspects which deviate from design norms. The only exceptions are the differences in shade between the colour of the text and the background colours of header, footer, and the two columns. The darker the text is compared to the background, the higher the reward.

The colour of the left column is used as a point of reference and from now on referred to as main colour. Penalty points are applied for the difference between the colour of the background and the complementary colour of the main colour, for the difference between the colour of the header and the calculated colour of the border of the logo, and for the differences between the colours of header/footer and the colour of the right column and the analogous colours of the main colour.

4.3   Fitness Function

The difference D_c between two colours c_1 and c_2 is calculated using the following formula:

$$D_c = (c_1.red - c_2.red)^2 + (c_1.green - c_2.green)^2 + (c_1.blue - c_2.blue)^2$$

The complementary colour c_{comp} is calculated by converting the original RGB colour into the HSL colour c_0, adding 180° to the hue value, and converting it back into the complementary RGB colour. The analogous colours c_{an1} and c_{an2} are calculated in a similar way, but instead of adding 180°, in this case 30° are added to create the first, and 30° are subtracted to create the second analogous RGB colour. Since both of header/footer colour and the colour of the right column could be either one of the analogous colours, as long as they are not the same, both cases are calculated and the lesser penalty is applied.

```
// ------------------------------------------------- //
// The class Node represents one line of html code //
// ------------------------------------------------- //
class Node
{
private:
    std::string _name;          // name
    bool _close;                // true = add a closing tag
    std::vector<Attribute> _att;// attributes
    std::vector<Node> _nodes;   // containing nodes
    std::string _content;       // the content between the tags
public:
    Node() {}
    Node(std::string v_name, bool v_close);
    Node& setContent(std::string v_content);
    Node& addAttribute(Attribute v_att);
    Node& addAttribute(std::string v_name, std::string v_value);
    Node& Node::addNode(Node v_node);
    std::string getString(unsigned int tab);
};
```

Fig. 3: HTML settings of the 'header'

$$c_{comp}.hue = c_0.hue + \frac{180°}{360°}$$

$$c_{an1}.hue = c_0.hue + \frac{30°}{360°}$$

$$c_{an2}.hue = c_0.hue - \frac{30°}{360°}$$

The following aspects are punished as well: height of the header too small/big compared to the height of the main body, main picture too small/big compared to the left column, additional picture too small/big compared to the right column, right column too small/big compared to the left column, and padding/margins too small/big compared to the mass of the content.

```
doc.add2body(Node("div", true)
    .addAttribute("id", "wrapper")
    /* add header */
    .addNode((Node("div", true))
        .addAttribute("id", "header")
        .addNode(Node("img", false)
            .addAttribute("src", "\"logo.bmp\"")
            .addAttribute("alt", "\"logo\"")
        )

        .addNode((Node("h1", true))
            .setContent("Genetic Algorithms")
        )
    )
)
```

Fig. 4: Class 'Node'

Furthermore, a penalty for lacking symmetry is applied by adding the mass of pictures (width * height) to the mass of the text (with a lower weighting) and comparing to the mass of the other side (left –> right) of the website. The result is the difference

D_m. The masses m_1 and m_2 are not supposed to be equal, instead the golden ratio φ is applied.

$$D_m = (m_1 - \varphi * m_2)^2$$

5   Results

The algorithm was tested with different input logos and a varying number of generations. The layout was fixed to include two panes and a header that included the logo, but the sizes of the panes, header and logo could be adjusted by the algorithm, The population had a constant size of 200 members throughout the test runs of the program.

As shown in Table 1, the runtime increases almost linearly with the amount of generations.

Table 1. Duration of the process depending on the number of generations

N generations (1000s)	0.1	1	5	20	50	100
Runtime (mins:sec)	0:12	2:0	12 :41	51 :30	127:29	250:36

The results of the first generations in each run are very poorly designed solutions without an appropriate structure or colour harmony (see e.g. Fig. 5). However, after only one hundred generations the first results can be seen (Fig. 6). Soon, a reasonable ratio between the different segments of the page evolves.

Then improvements usually stagnate, as it takes a very long time for all the different colour values to arrange in a pleasing way. Fig. 7 shows the solutions after 1,000 (top

Fig. 5: First generation

left), 2,000 (top right), 6,000 (bottom left), and 20,000 generations (bottom right). Apparently, the reward for adjusting the colour of the header depending on the colour of the logo is barely enough to compensate the loss for shifting away from the analogous colour setting.

After configuring the weightings to reward the similarity between logo and header colour even more, the algorithm came up with an acceptable solution after just 2,000 generations (**Error! Reference source not found.**Fig. 8). However, this version was still likely to get stuck in local maxima; a different run of the same program produced a very small logo (about 5mm wide). Because increasing the logo size would harm the achieved symmetry, this problem was still not solved after 100,000 generations.
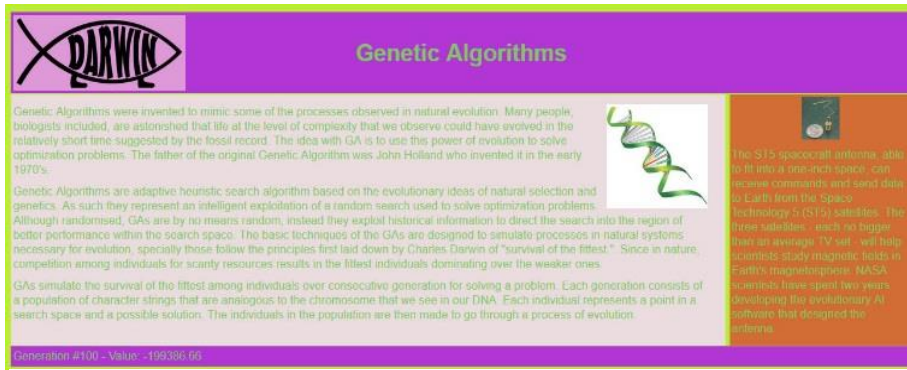
Fig. 6: 100 generations
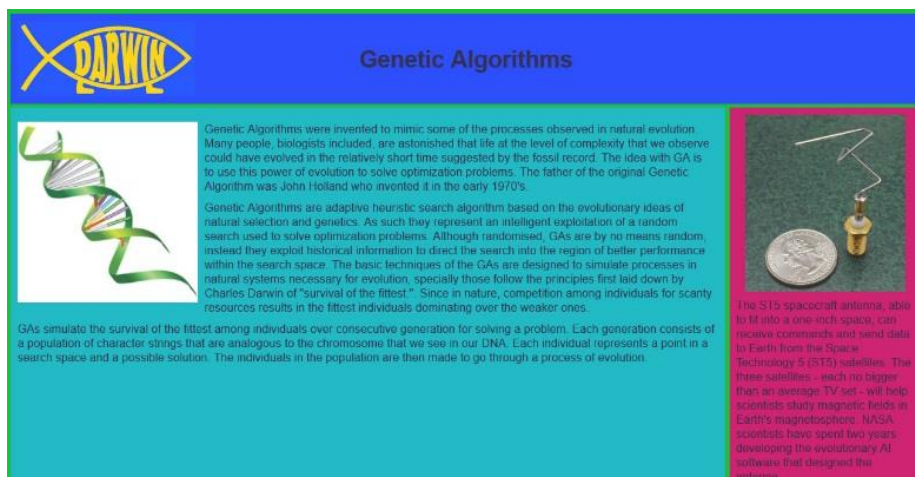

Fig.7: 100 - 20 000 generations

Fig. 8: Reconfigured algorithm, 2 000 generations

In the final version (Fig. 9), the analogous colours are calculated correctly and the

proportions are set appropriately, leading to a more harmonic design. This website took 60,000 generations although an identical layout with a slightly greener main colour was achieved in only 20,000 generations.
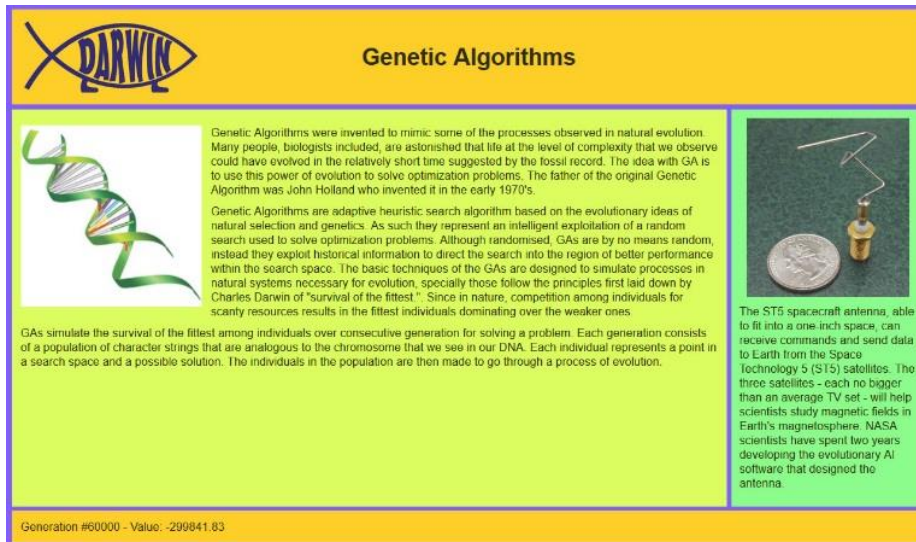

Fig. 9: Reconfigured algorithm, 60 000 generations

6 Future Work

A solution such as the one described here could be used as groundwork for further improvements. For example, an interactive genetic algorithm [25] which evaluates the individuals based on user feedback instead of set rules could take a created solution as starting point. Usually, interactive GAs need a lot of time and user feedback to come up with acceptable solutions, which fatigues the test users and impairs the quality of the outcome. If the first generation was already populated by acceptable solutions, these resources could be used to optimise the results.

With a higher variety in outcomes due to milder restraints, the results could be used as an inspiration for human designers. In that case, the overall appeal of the solution is arguably less important than single interesting aspects about the design which can serve as a source for a new design idea. A further experiment in which designers evaluated the system's results for attributes tested in the fitness function (symmetry, harmony etc.) and also aspects that are not included (e.g. boldness and novelty) might facilitate this.

Another application area might be dynamically adjusting websites to new content. This can be used if the content is uploaded or created by users rather than the owners of the page. It would enable the page to be adjusted in almost real-time.

Furthermore, the fitness function of this project can be used as a quick evaluation tool for existing websites. This might be a useful source of data for web crawlers.

Different fitness functions for other website types besides informational ones could be developed. Layout options that could be introduced would be:

- determining an appropriate font and size for the text;
- automatically dividing tabs;
- dynamic content placement.

However, adding more parameters greatly increases the time and generations needed for a satisfying solution so there may be practical limitations on the features that can be taken into account.

Technical changes that could be introduced include:

- mutation via bit flipping;
- dynamic mutation rates: If progress stagnates, a higher mutation rate might help to overcome local maxima, while the elite population prevents the solution from losing achievements by an extensive amount of random search.

Acknowledgements

References

[1] T. Lavie and N. Tractinsky, "Assessing dimensions of perceived visual aesthetics of websites,"International Journal of Human-Computer Studies, 60(3), 269-298, 2004.

[2] K. Oyibo and J. Vassileva, "What drives perceived usability in mobile Web design: classical or expressive aesthetics?" In: A. Marcus and W. Wang (Eds.), Design, User Experience, and Usability: Theory, Methodology, and Management. Cham, Switzerland: Springer, 2017.

[3] L.A. Peracchio and D. Luna, "The role of thin-slice judgments in consumer psychology.," Journal of Consumer Psychology, 16, 25–32, 2006.

[4] C. Thompson, "Search engines invite new problems," Marketing Management, 13, 52–53, 2004.

[5] P.B. Lowry, D.W. Wilson and W.L. Haig, "A picture is worth a thousand words: Source Credibility theory applied to logo and website design for heightened credibility and consumer trust," International Journal of Human–Computer Interaction, 30 (1), 63-93, 2014.

[6] S. Park, "Webpage design optimuization using genetic algorithm driven CSS". Retrospective Theses and Dissertations, 14548, Iowa State University. 2007. https://lib.dr.iastate.edu/14548

[7] E.P. Mensch, "Optimizing Website Design Through the Application of an Interactive Genetic Algorithm". Senior Projects Spring 2016. http://digitalcommons.bard.edu/senproj_s2016/313

[8] A. Olliver, N. Monmarchë and G. Venturini, "Interactiove Design of Web Sites with a Genetic Algorithm", In Proceedings of tne IADIS Internatiinal Conference WWW/Internet, 2002, 355-362.

[9] M. Mitchell, An introduction to genetic algorithms, London, MIT Press, 1996.

[10] P. Kora and P. Yadlapalli, "Crossover operators in genetic algorithms: a review," International Journal of Computer Applications, 162(10), 34-36, 2017.

[11] P.K. Chawdhry, R. Roy amd R.K. Pant (Eds.), Soft Computing in Engineering Design and Manufacturing. London: Springer, 1998.

[12] T. Bäck, Evolutionary Algorithms in Theory and Practice: Evolution strategies, Evolutionary programming, Genetic algorithms, Oxford, Oxford University Press, 1996.

[13] S. Baluja and R. Caruana, "Removing the genetics from the standard genetic algorithm", in Proceedings of the Twelfth International Conference on Machine Learning, Lake Tahoe, CA. July, 1995.

[14] C.W. Ahn and R.S. Ramakrishna, "Elitism-based compact genetic algorithms," IEEE Transactions on Evolutionary Computation, 7(4), 367-385, 2003.

[15] A. Hussam, "The Gestalt principle: Design theory for Web designers," Web Design Theory Blog, https://webdesign.tutsplus.com/articles/the-gestalt-principle-design-theory-for-web-designers--webdesign-1756, 2011, January 12.

[16] R. Arnheim, Kunst und Sehen: Eine Psychologie des schöpferischen Auges, (3rd ed.). Berlin: De Gruyter, 2000.

[17] A.N. Tuch, J.A. Bargas-Avila and K. Opwis, "Symmetry and aesthetics in website design: it's a man's business," Computers in Human Behavior, 26(6) 1831-1837, 2010.

[18] A. Altaboli, and Y. Lin, "Effects of unity of form and symmetry on visual aesthetics of website interface design," Proceedings of the Human Factors and Ergonomics Society Annual Meeting, 56(1), 728–732, 2012.

[19] P. Hemenway and A. Ray, Divine Proportion: Phi in Art, Nature, and Science, New York, Sterling, 2005.

[20] M. Lieberman, "How important are images in your website design? Inbound – the Blog, https://www.square2marketing.com/blog/bid/ 121405/how-important-are-images-in-your-website-design, 2012, March 22.

[21] S. Djamasbi, S., Siegel M. and T. Tullis, "Generation Y, web design, and eye tracking.," International Journal of Human-Computer Studies, 68(5), 307-323, 2010.

[22] T.S. Tullis and C.M. Tullis, "Statistical analyses of E-commerce websites: can a site be usable and beautiful?" In 12th International Conference on Human-Computer Interaction, Beijing, China, 2007.

[23] B. Jones, "Color theory: the importance of color in Web design," Design and Promote Blog, https://www.designandpromote.com/color-theory-the-importance-of-color-in-web-design/, 2014, June 27.

[24] F. Alabsi and R. Naoum, "Comparison of selection methods and crossover operations using steady state genetic based intrusion detection system," Journal of Emerging Trends in Computing and Information Science, 3(7), 1053-1058, 2012.

[25] H. Takagi, "Interactive evolutionary computation: fusion of the capacities of EC optimization and human evaluation," In: Proceedings of the IEEE 89 (9): 1275–1296, 200.