Automation of Fabric Pattern Construction using Genetic Algorithms

OMEMA AHMED, Department of Computer Science - Habib University, Pakistan
MUHAMMAD SALMAN ABID, Department of Computer Science - Habib University, Pakistan
AIMAN JUNAID, Department of Computer Science - Habib University, Pakistan
SYEDA SALEHA RAZA, Department of Computer Science - Habib University, Pakistan

This paper introduces the use of Genetic Algorithms to evolve fabric patterns from randomly generated seeds. The patterns are evolved from random, often dull coloring of the image, to bright multi-color patterns that are aesthetically pleasing in nature. The main problem that this paper intends to solve is to introduce complete automation in the design process of patterns, which have historically been dependent upon human arbitrators to judge the quality of intermediate outputs. In its stead, the proposed algorithm evaluates the quality of the image using inherent latent features present in the image itself. Our algorithm takes into account the distribution of color, global contrast, and the overall dullness score of the image to evaluate the quality of the generated patterns. To create diverse patterns that feel more natural, different approaches are experimented with. These include the use of L-systems and image processing techniques, in a bid to construct a pattern which seems more human-like, rather than just rudimentary digital art.

CCS Concepts: • **Theory of computation** → *Theory of randomized search heuristics*; • **Computing methodologies** → **Artificial intelligence**; • **Applied computing** → *Arts and humanities.*

Additional Key Words and Phrases: genetic algorithm, evolutionary algorithm, color palette, textile pattern, fabric pattern, l-systems, image processing

ACM Reference Format:
Omema Ahmed, Muhammad Salman Abid, Aiman Junaid, and Syeda Saleha Raza. 2018. Automation of Fabric Pattern Construction using Genetic Algorithms. 1, 1 (May 2018), 10 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

The process of generating new patterns for fabrics is a crucial one in the textile industry. Creating a pattern from scratch is tedious work, and it takes multiple hours of skilled labor for every pattern that is designed. To help automate this process, this paper presents some rules which would determine what a good pattern would consist of. Art is subjective, and so are patterns by extension. However, using as an inspiration the criteria mentioned in an article by Microsoft India [7], we came close to some characteristics that would be commonplace in popular fabric patterns. Embodying these features, this paper devises a new methodology that aims to eliminate the human feedback element from the design process in order to truly automate the pattern construction.

Authors' addresses: Omema Ahmed, ahmed.omema@hotmail.com, Department of Computer Science - Habib University, Karachi, Pakistan; Muhammad Salman Abid, salman-abid@hotmail.com, Department of Computer Science - Habib University, Karachi, Pakistan; Aiman Junaid, amo.junaid1@gmail.com, Department of Computer Science - Habib University, Karachi, Pakistan; Syeda Saleha Raza, saleha.raza@sse.habib.edu.pk, Department of Computer Science - Habib University, Karachi, Pakistan.

XXXX-XXXX/2018/5-ART $15.00
https://doi.org/XXXXXXX.XXXXXXX

In this paper, a Genetic Algorithm is used to evolve a randomly generated seed according to a set fitness criteria, resulting in a multi-colored fabric pattern. Some post-processing techniques are then applied to the evolved seed, in order to generate a more natural pattern. The details of the algorithm are further described in the following sections.

The rest of the paper is organized as follows: Section 2 provides a background of the techniques used in this paper, and explores the existing work done in this domain. Section 3 presents details of the methodologies employed in this paper. Section 4 and 5 give a summary of our results, showcasing how the pattern has evolved from scratch. They also compare our results with those of similar works that have implemented GAs to generate patterns. Finally, Section 6 concludes the paper and Section 7 mentions the possible future work in this domain.

2 BACKGROUND

Genetic algorithms are a type of optimization algorithm, meaning they are used to find the optimal solution(s) to a given computational problem that maximizes or minimizes a particular function. Genetic algorithms represent one branch of the field of study called evolutionary computation, in which they imitate the biological processes of reproduction and natural selection to solve for the 'fittest' solutions [8]. Like in evolution, many of genetic algorithm processes are random, however, this optimization technique allows one to set the level of randomization and the level of control [8]. To imitate the natural cycle of evolution, the algorithm runs for multiple iterations called 'generations', and each entity to be evolved is called a 'chromosome'. In each generation, three basic genetic operators are applied to each chromosome with certain probabilities, i.e. selection, crossover and mutation [8].

2.1 Procedure

The general procedure for Genetic Algorithms, as outlined by [1] is:

(1) **Start:** Generate random population of n chromosomes (suitable solutions for the problem).
(2) **Fitness:** Evaluate the fitness $f(x)$ of each chromosome x in the population.
(3) **New Population:** Create a new population by repeating the following steps until a new population is complete
   (a) **Selection:** Select two parent chromosomes from a population according to their fitness (the better fitness, the higher the chance to be selected).
   (b) **Crossover:** With a crossover probability, cross over the parents to form a new offspring (child). If no crossover was performed, offspring is an exact copy of parents.
   (c) **Mutation:** With a mutation probability, mutate new offspring at each locus (position in chromosome).
   (d) **Accepting:** Place new offspring in a new population.
(4) **Replace:** Use newly generated population for a further run of algorithm.
(5) **Test:** If the end condition is satisfied, stop, and return the best solution in current population
(6) **Loop:** Go to step 2.

2.2 Related Work

The possibility of pattern generation in textile production processes has previously been investigated [4], although improvement in the quality of the generated results is required. Genetic Algorithm has also been used to generate new carpet patterns [3], and in constructing Jacquard patterns for weft knitted fabrics [5]. However, both of these approaches are based on interactive genetic algorithm, where human feedback is required to select and optimize the generated designs. Zamani [3] uses already existing carpet design as a source to generate new designs which are then selected or discarded based on the observer's evaluation of the design. An attempt to generate art tile patterns using Genetic Algorithm was also made by M. Heidarpour and S. M. Hoseini [2], where patterns were generated independent of human input. A randomly initialised grayscale image block was used

as initial chromosome, which was evolved and combined to generate a complete tile. This approach, although producing new tile patterns, was restricted to grayscale images with low resolution patterns. We aim to improve upon their idea, to generate complete patterns both in the RGB colorspace and with a higher resolution.

3 METHODOLOGY

This section provides an overview of the methodology to generate the patterns. The process compromises of the following four steps:

- Polygon Construction
- Color Palette Generation
- Evolution via GA
- Pattern formation from evolved block

3.1 Construction of Polygons

The covering of an entire plane with one or more geometric shapes which do not overlap, and leave no gaps between them is known as 'tiling'. This paper incorporates tiling to generate fabric patterns consisting of different polygons. The construction of polygons comprises of the following steps:

(1) Create a basic 200 pixels by 200 pixels sized structure of the image block.
(2) Randomly select the type of line formation i.e. horizontal, vertical or slant, and generate two random points on the distinct edges of the image block. For example, to generate a horizontal line, two random vertices will be chosen from two opposing edges of the block with the same y value.
(3) After the line has been drawn, consider the new segment bounded by this line to be a new polygon, and recursively divide this polygon using line partitions until the specified depth limit is reached.
(4) Iterate over the list of polygons and randomly assign colors from the color palette to each polygon.

Examples of polygons generated using recursive division are shown in Fig 2

3.2 Color Palette

Different color palettes are used to fill RGB colors in the generated polygons. To fill the patterns using contrasting colors, this paper makes use of inspiration images. A color palette is generated from the most dominant colors present in the inspiration image. To create the color palette using this approach, KMeans clustering is used. First, the pixel intensities of the RGB images are separated into different clusters based on their colors. Then, the most significant color from each cluster is extracted and added to the color palette. The color palettes in Fig 1 are generated using this approach.

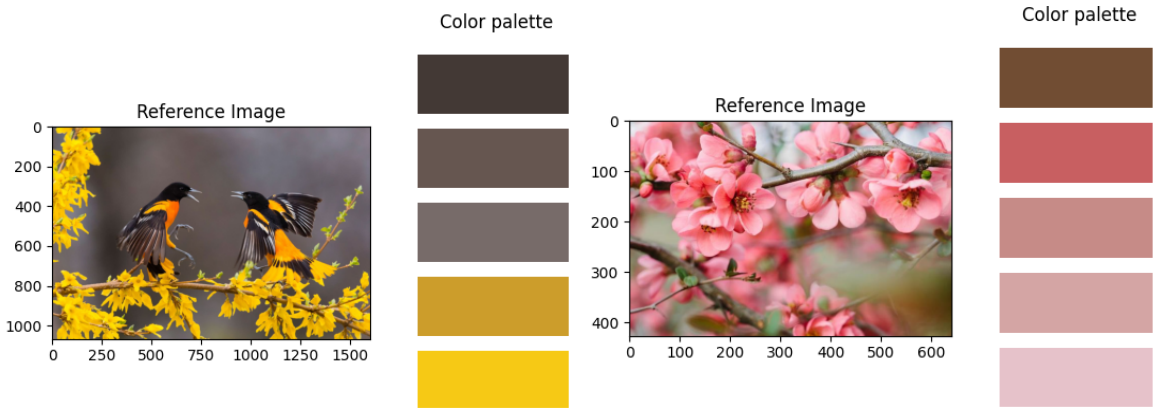


Fig. 1. Color Palettes generated using Reference Images

3.3 Genetic Algorithm

The genetic algorithm as outlined in Section I has been used to evolve fabric patterns. Our population was initialized with 10 randomly generated chromosomes. The following steps explain our formulation of the problem to generate new textile patterns.

3.3.1 Chromosome Structure.

Our population comprises of multiple blocks of 200 by 200 px RGB images, which were randomly initialized with different polygons by joining any two points on the distinct edges of the block. This makes up one chromosome of our population. Each polygon in this chromosome is then assigned a unique color from the selected color palette which is generated using reference images, as outlined earlier. This process is repeated several times to create multiple polygons on this chromosome, as shown in Fig. 2. By this process, we initialised our population with ten such chromosomes.

Fig. 2. Initial Chromosomes

3.3.2 Fitness Function.

Our problem aims to maximize image contrast, distribution of the colors present in the patterns, and the brightness of the image. To compute fitness for each pattern in our population, the individual fitness for each pixel is calculated to form the cumulative fitness, using the formula detailed below:

$$f = \frac{1}{\sigma} \sum_{i=0}^{N} Image_{H_i+S_i+(V_i*contrast)} \tag{1}$$

where σ is the standard deviation between the different colors present in the image. The higher the value of σ, lower the fitness value. A high value of σ indicates that the colours in our image are not evenly distributed. *HSV*

represent Hue, Saturation and Value components of the image and *contrast* is the overall contrast of the image which is calculated using the formula:

$$contrast = \frac{max_{\text{image pixels}} - min_{\text{image pixels}}}{max_{\text{image pixels}} + min_{\text{image pixels}}} \tag{2}$$

where *max* and *min* represent global maximum/minimum brightness of the pixels derived from a gray-scale conversion of the source.

3.3.3 *Crossover.*

Two parents were chosen from our population of 10 chromosomes using random selection. These two parents were crossed over to produce a new offspring. During crossover, each parent was divided into two sections from the middle, either vertically or horizontally from the centre of the image. The first half of parent 1 was used and the polygons lying in this part of the image were copied to the offspring's image, using the same colors as that of parent 1. Similarly, the second half of parent 2 was copied to the offspring.

(a) Parent 1 (b) Parent 2 (c) Offspring

Fig. 3. Crossover Result

In the example in Fig. 3, parent chromosomes are horizontally divided from the middle, with top half of Parent 1 (Fig. 3(a)) and bottom half of Parent 2 (Fig. 3(b)) resulting in the offspring shown in Figure 3(c).

3.3.4 *Mutation.*

The chance of mutation of our offspring is based on a random variable in the range $[0, 1]$, where mutation takes place when the value of this random number comes out to be greater than 0.5 i.e. there is a 50% chance of mutation.
In mutation, the colors of the polygons of our offspring are updated using random colors from the chosen color palette.

3.3.5 *Survival Selection.*

In each generation, crossover and mutation is repeated five times. The new size of our population thus becomes 15. From our new population, the top 10 chromosomes (sorted by their fitness as a criteria) are chosen for the next generation using the truncation selection scheme.

3.3.6 *Termination Condition.*

Our fitness function converges after only a few generations. After various experiments, the optimal generation size for our problem came out to be 30.

3.4 Pattern Formation

A final block is generated as a result of the evolutionary process. To form a complete pattern, variations of the final block are combined in different orientations. The final pattern is constructed using a random dimension for

the number of blocks the pattern will be made up of.
The transformation techniques applied on our final constructed pattern include the following:

(1) Horizontal mirroring
(2) Vertical mirroring
(3) Both horizontal & vertical mirroring

A random mix of each of these is used to generate our final pattern. The dimensions, d for our pattern are in the range [5,40], where $d * d$ represents the final size of our pattern.
The final pattern is generated using two different approaches, with each having a significantly different design.

(1) The first approach uses unmodified output from the GA to make use of in the pattern generation. This results in a high contrast pattern with sharp edges and distinct color boundaries.
(2) The second approach applies post-processing to smooth out the sharp edges found in the images, giving it a softer overall theme. This is achieved by applying a Mode Filter ($k = 25$) which is an edge-preserving smoothing filter that computes the mode of the empirical density of the image.
The image is then overlaid with fractals formed using different rules for L-systems. L-systems are a formal grammar that is used to describe the growth of an initial axiom. The system closely follows the behaviour of natural phenomenon such as plant growth. The grammar constructs the string by replacing characters in the string recursively with defined phrases of the alphabet, creating a fractal-like growth of the axiom. This finally results in a more organic image which more closely resembles human-generated patterns with a less overall synthetic theme.

Algorithm 1 Methodology Overview

```
colors = generateColorPalette()
population = [ ]
for 1 to 10 do
   population += constructPolygons(colors)
end for
GA(population)
max_fit = maxFitness(population)
formTile(max_fit)
```

The algorithm detailed above gives a high-level view of the flow of the program, and shows the steps to achieve the final design. Initially, a coloring is generated using the generateColorPalette() method, making use of the clustering technique described previously to attain it.
The following loop runs for ten iterations; it generates instances of initial patterns using the constructPolygon() method, with the coloring as a parameter that will determine the polygon color fills.
The choice of the population size determines the number of iterations of this loop, as the GA() function call uses this collection of image instances as the seed population. A variation in the population parameter will be reflected in the seed population initialization accordingly.
After the evolution process is completed, the most fit chromosome will be chosen as the candidate image block, and a tile will be created with various symmetric repetitions of the block in the formTile() method.

4 EXPERIMENTS AND RESULTS

The results of our Genetic Algorithm consist of patterns generated after setting two different values of the background parameter; choosing to either use a white canvas to draw polygons upon, or choosing a random

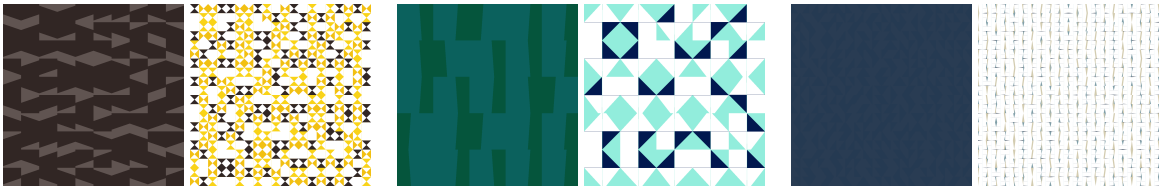
color from the color palette of the reference image. The value of the parameter significantly affects the coloring of the pattern, and examples from both experiments are shown below.

4.1 Patterns Generated on White Canvas

Due to the high degree of stochastic behavior in the algorithm, the generated patterns exhibit unique geometry in their visuals. Some of the patterns using a white canvas are shown below:

4.1.1 Without Post-Processing.

The following patterns do not use post-processing

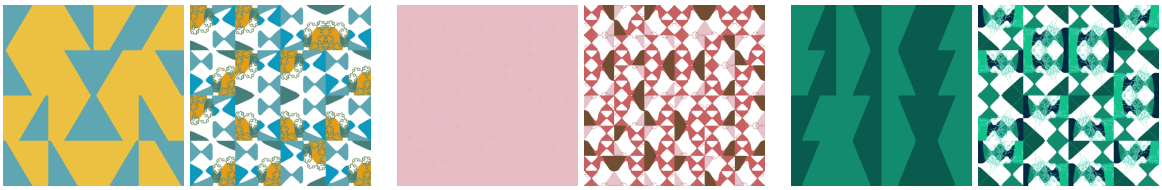


(a) Pattern 1 (b) Pattern 2 (c) Pattern 3

Fig. 4. Resulting pattern at Generation 0 vs Generation 30

4.1.2 With Post-Processing.

The following patterns use post-processing



(a) Pattern 1 (b) Pattern 3 (c) Pattern 3

Fig. 5. Resulting pattern at Generation 0 vs Generation 30

These figures generated their palette using different reference images, and the random generation in the first step yielded low-quality and almost monochromatic results. This pattern was then evolved further, and at the 30th generation we were able to achieve a design that was both high-contrast and low in dullness score.

4.2 Patterns Generated on Colored Canvas

Patterns using a background color chosen from the reference image show softer contrast in comparison to those with the white background. A few of them are given below:

4.2.1 Without Post-Processing.

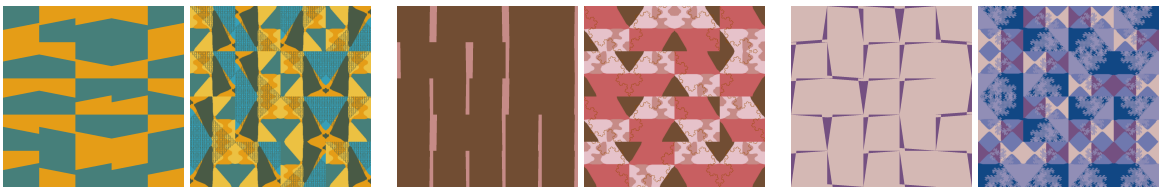
The following patterns are constructed without any filters after generation and evolution:

(a) Pattern 1 (b) Pattern 2 (c) Pattern 3

Fig. 6. Resulting pattern at Generation 0 vs Generation 30

4.2.2 *With Post-Processing.*

The following patterns use post-processing techniques, including smoothing and mode filters:



(a) Pattern 1 (b) Pattern 2 (c) Pattern 3

Fig. 7. Resulting pattern at Generation 0 vs Generation 30

The performance of the genetic algorithm run over multiple epochs is visualized by plotting the average values of best fitness and average fitness so far, with an increasing number of generations on the x-axis.

The trend in the graph on Fig 8 shows a best fitness value of approximately 110 as an average of ten iterations run for the algorithm. The best fitness value shows early convergence, as the algorithm soon achieves a pattern that satisfies the fitness function adequately for the given color choices. In contrast to this, the average-so-far fitness keeps on continually improving, but there is still a stark contrast between average fitness and best fitness values of the algorithm, indicating that there exists a better solution than all the other chromosomes present.

5 COMPARISON WITH EXISTING WORK

The patterns shown in Fig 9(a) and Fig 9(b) are results of works that have introduced methods of GA which create patterns with human intervention. Based on their claim that there is no fitness function for art, they use a human arbitrator to choose the most fit individuals from the population in each generation. The paper by Obe et. al [4] presents the user with eight choices, from which they select the best option. On the other hand, the fitness selection in the paper by Zamani et. al [3] is carried out by the human through assigning a fitness value to each chromosome, repeated for each generation until convergence. Contrary to their claims, we have devised rules, based on dullness score and color contrast present in the image, that make it possible to quantify the visual appeal of a chromosome to some extent. Our results are a testament to their validity and show that these do help create bright and appealing patterns.

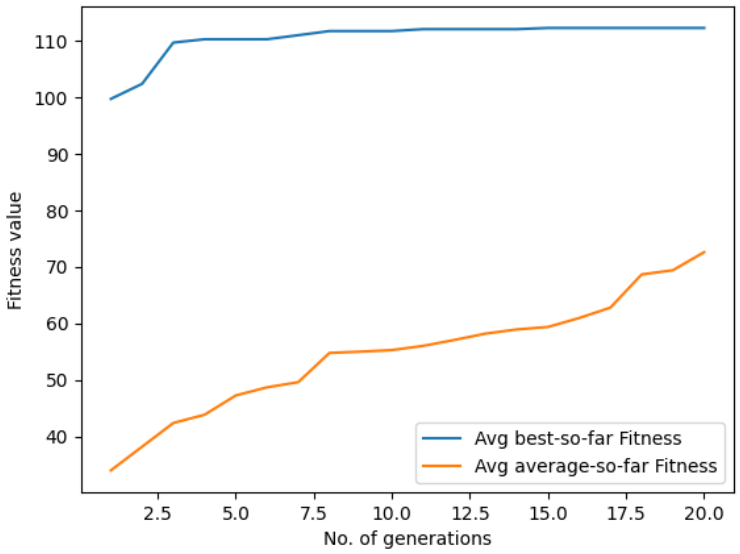
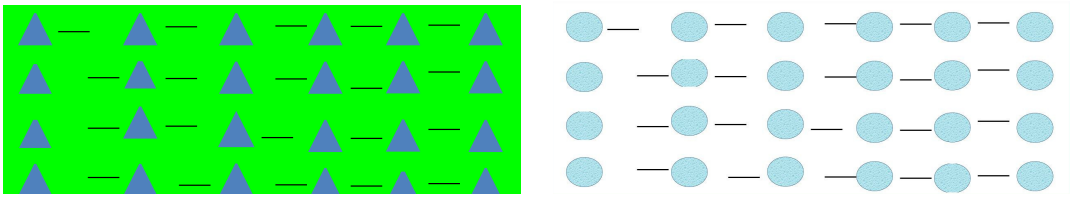
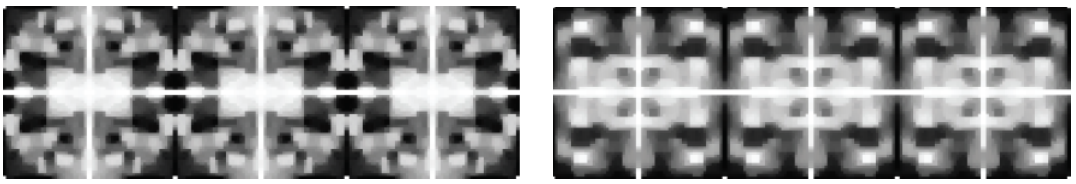


Fig. 8. Fitness Trend Analysis



(a) Patterns generated by Obe. et al [4]

(b) Carpet design patterns generated by Zamani et. al [3]



(c) Tile patterns generated by M. Heidarpour et. al [3]

Fig. 9. Results of Existing Work of Research in this Domain

The tile patterns created by M. Heidarpour [2] in Fig 9(c) were one of the first attempts at creating patterns completely autonomously using GA, and hence have low fidelity and gray-scale images. Their algorithm consisted of an entirely randomized seed, with each pixel of the image having a luminance value between $[0 - 255]$. In contrast, our paper proposes a seed consisting of polygons generated via recursive division where each polygon is randomly assigned a color value using the methodology explained in Section 3.2 and Section 3.3.1. This helps for the initial seed to have a semblance of a pattern from the start, that creates patterns both autonomously and of high quality (high fidelity and RGB images).

6 CONCLUSION

Most algorithms that deal with generating artistic results using genetic algorithms are interactive; incorporating user feedback as part of the decision making process. This paper showcases a non-interactive method, that is able to both accept a reference input and generate a bright and appealing pattern.
The paper addresses a major bottleneck in the textile industry; the cumbersome process of designing a pattern before a new product can be launched. By employing this solution, vendors can reliably generate patterns that follow color rules in their generation, and are therefore highly likely to be popular with consumers. It may be a cost-effective and efficient solution to supplement existing manufacturing processes.

7 FUTURE WORK

Although aesthetically pleasing, there are still some aspects of the paper that can be built upon for further research. Since our patterns are generated using only polygons, further variation in patterns can be achieved by incorporating circular shapes in the construction process.
Moreover, random point crossover and different mutation techniques can also be looked into to explore different avenues of evolution in the GA algorithm.
To improve upon the fractals used in the paper, custom grammar could be developed for new and unique fractals that more closely resemble floral designs created by designers in the textile industry.

REFERENCES

[1] L. Haldurai, T. Madhubala2 and R. Rajalakshmi (2016) *A Study on Genetic Algorithm and its Applications*, Department of Computer Science (PG), Kongunadu Arts and Science College, Coimbatore, India
[2] M. Heidarpour and S. M. Hoseini, *Generating art tile patterns using genetic algorithm*, 2015 4th Iranian Joint Congress on Fuzzy and Intelligent Systems (CFIS), Zahedan, Iran, 2015, pp. 1-4, doi: 10.1109/CFIS.2015.7391652.
[3] F. Zamani , M. Amani-Tehran & M. Latifi (2009) *Interactive genetic algorithm aided generation of carpet pattern*, The Journal of The Textile Institute, 100:6, 556-564, DOI: 10.1080/00405000802125055
[4] Obe, Olumide & Egwuche, Ojonukpe. (2018). *Genetic Algorithm for fabric pattern generation in textile industries*, 16. 86-91.
[5] Semnani, Dariush & Hadjianfar, Mehdi & Aziminia, Hamed & Sheikhzadeh, Mohammad. (2014). *Jacquard pattern optimizing in weft knitted fabrics via interactive genetic algorithm*, Fashion and Textiles. 1. 1-9. 10.1186/s40691-014-0017-2.
[6] Hee-Su Kim, Sung-Bae Cho, Application of interactive genetic algorithm to fashion design, Engineering Applications of Artificial Intelligence, Volume 13, Issue 6, 2000.
[7] N. R. U. G. A. C. M. J. P. A. Sonam Damani, *Transforming traditional Block Prints with AI*, Microsoft India Technology Blog, 9 September 2019. [Online]. Available: https://www.microsoft.com/en-in/msidc/blogs/Articles/Transforming-traditional-Block-Prints-with-AI/52718cb6-a066-4de3-bb25-96cf9c8569e.
[8] Goldberg, D. E. (1989). Genetic Algorithms in Search, Optimization, and Machine Learning. Reading: Addison-Wesley.
[9] Aupetit, Sébastien, et al. Interactive evolution of ant paintings. Evolutionary Computation, 2003. CEC'03.
[10] Lewis, Matthew. Evolutionary visual art and design. The art of artificial evolution. Springer Berlin Heidelberg, 2008. 3-37.