An Intelligent Design Explorer for New Violin Shapes

Hao Wang and Lutz Hamel
Department of Computer Science and Statistics, University of Rhode Island, Kingston, U.S.A.
hao_wang@my.uri.edu, lutzhamel@uri.edu

Keywords: Evolutionary Computation, Evolutionary Strategies, Creative Evolutionary Systems, Art and Design.

Abstract: This work focuses on a design tool, the violin design explorer, for classic violin outline design. Our design tool employs an evolution strategies algorithm and can be viewed as a creative evolutionary system. With this system, users without any design knowledge can create their favorite violin outline through a set of simple choices. The underlying violin designs are implemented using the Digital Amati geometry engine. In order to validate the design tool we conducted an anonymous experiment. Four groups of participants were asked to design their favorite violin outline. Design patterns that emerged within these four participant groups during the experiment seem to validate that our design tool works as we envisioned. We view this implementation of the designer as the first stage of a comprehensive violin design tool.

1 INTRODUCTION

The origins of evolutionary computation can be traced back to the late 1950s and started to receive significant attention during the 1980s (Bäck et al., 2018). Pioneers have conducted research using evolutionary computation, especially in the field of design, music, and art. Here we explore whether we can use evolutionary computation in the area of violin outline design. Our design tool, the violin design explorer, employs an evolution strategies algorithm and can be viewed as a creative evolutionary system. The violin design explorer aims to help users without design experience to directly integrate their imagination and creativity into violin outline design and create novel solutions. The explorer abstracts each violin outline into a unique genotype representation suitable for design representation within the Digital Amati geometry engine (Digital Amati, 2018) which also handles violin design parameters and constraints. Instead of using a fitness function the violin design explorer uses human selection to determine the parent populations of the evolutionary computation. Besides allowing users without design background or knowledge to undertake non-trivial design tasks another advantage of creative evolutionary systems is that they allow the user to sidestep the limitations of "conventional wisdom" and "design fixation" (Menges and Ahlquist, 2011).

We validated the design tool by conducting an anonymous experiment. Four groups of participants were asked to design their favorite violin outline. Design patterns that emerged within these four participant groups during the experiment seem to validate that our design tool works as we envisioned.

We view this implementation of the designer as the first stage of a comprehensive violin design tool. A next logical step would be to extend the designer to embody acoustic characteristics besides the pure aesthetic consideration currently implemented through Digital Amati. A good starting point for developing such an extension would be the research by Carleen Hutchins (Hutchins, 1981).

The remainder of the paper is structured as follows. Section 2 discusses related work. In Section 3 we describe the main internal structures of our design tool. Section 4 describes our experiment and we close the paper with Section 5 where we discuss conclusions and further work.

2 RELATED WORK

2.1 Creative Evolutionary Systems

A creative evolutionary system is a computer system that makes use of some aspect of evolutionary computation to explore creative solutions (Bentley, 1999a). This kind of system is designed to:

- Aid our own creative processes, and/or
- to generate results to problems that traditionally

Wang, H. and Hamel, L.
An Intelligent Design Explorer for New Violin Shapes.
In *Proceedings of the 11th International Joint Conference on Computational Intelligence (IJCCI 2019)*, pages 229-236
ISBN: 978-989-758-384-1

required creative people to find solutions (Bentley and Corne, 2002).

There are five elements that make up a creative evolutionary system framework:

- An evolutionary algorithm
- A genetic representation
- An embryogeny
- A phenotype representation
- Fitness function(s) and/or processing of user input (Bentley and Corne, 2002)

A creative evolutionary system uses some kind of evolutionary algorithm to create new offspring. The genotype representation is a series of elements abstracted from real-world objects. The evolutionary algorithm modifies the genotype representation through recombination and mutation processes to generate a new genotype representation. Once the new genotype is generated, an embryogeny process decodes the genotype representation to construct the corresponding phenotype. Finally, a fitness function or human selection process will help users select an optimal solution to create the solutions that make up the next generation. The difference between a creative evolutionary system and standalone evolutionary algorithms is that an evolutionary algorithm is used to search for an optimal solution whereas a creative evolutionary system acts as an explorer to provide inspiration and investigate creative solutions.

2.2 Evolutionary Computation in Engineering Design

As one of the most well known evolutionary algorithms, the genetic algorithm (Kramer, 2017), is now a well-established technique in engineering design applications. For example, in 1992 John Frazer and his students used a genetic algorithm in yacht hull design (Frazer, 2002). Their program was intended to explore a range of widely different alternative solutions to yacht hull design (Figure 1). The program starts by generating a randomized initial population of yacht hull designs. All designs are evaluated by a fitness function which considers both objective and subjective factors. The objective engineering factors include parameters such as stability, center of buoyancy, wetted surface area, prismatic coefficient, and blocking. The subjective factors (*i.e.* designer criteria) mainly considered aesthetic appearance, historic tradition, and allusion of form.

On the basis of the fitness function, the designs with the highest fitness scores are selected and become the parent population. The crossover and mutation processes are then applied to the parent population and produce new offspring. These processes continue until a satisfactory design emerges. The yacht hulls design program produces a range of choices for clients and provides a series of new ideas in yacht hull designs to explore.

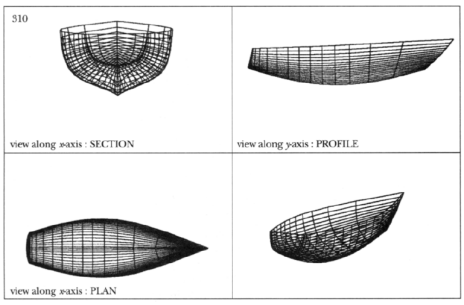

Figure 1: Yacht hull designs (Frazer, 2002).

2.3 Evolutionary Computation in Art

Compared with engineering design, music and art are more subjective, fuzzy and difficult to analyze. But it can still be evolved through evolutionary computing. The algorithmic art assembly (Algorithmic Art Assembly, 2019) hosted in San Francisco in 2019 showcased a diverse range of artists who are using algorithmic tools and processes in their works. Although the algorithmic tools are not necessarily evolutionary in nature it points to a new approach of using computational tools in music and art generation. DarwinTunes (MacCallum et al., 2012) is a computational engine to evolve music. As shown in Figure 2 it uses a genetic algorithm to maintain a population of tree-like digital genomes, each of which encodes a computer program representing a song. DarwinTunes uses recombination and mutation to randomly create songs with novel musical motifs, rhythms, and harmonies. Instead of using a fitness function, user selection via a web interface is the only way to help DarwinTunes select the parent individuals for the next generation.

One of the best known "evolutionary artists", Steven Rooke, uses genetic programming algorithms to evolve some astonishing pieces of art (Rooke, 2002). The image evolution system begins with a series of randomly generated tree structure genomes. Based on these genomes the computer generates corresponding images according to specific mapping rules. In each generation, the image evolution system determines the parent genomes by random selection or user selection. The parent genomes generate new offspring through crossover and mutation steps. Figure 3 shows some of the images generated by the image evolution system.

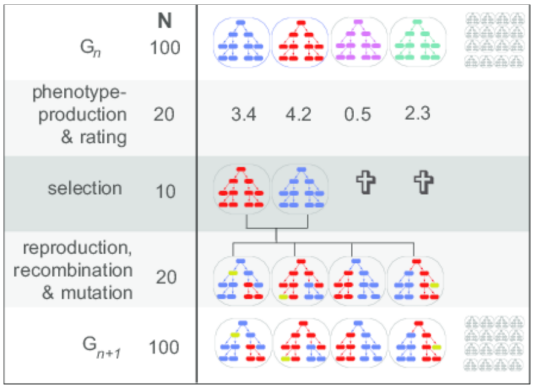

Figure 2: The DarwinTunes program (MacCallum et al., 2012).

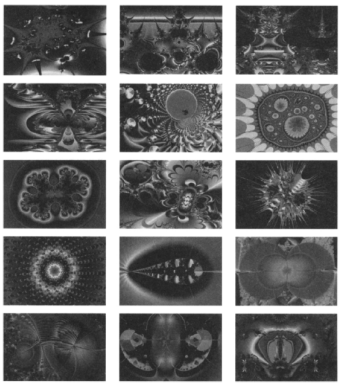

Figure 3: Some creative outcomes of Rooke's image evolution system (Rooke, 2002).

3 THE DESIGNER

Our violin design explorer is conceived as a creative evolutionary system which can help users without formal design background create their favorite violin outline design.

3.1 The Digital Amati Project

Digital Amati (Digital Amati, 2018) developed by Harry Mairson is a software package based on Euclidean geometry and uses a "geometry engine" language (AmatiML) to design classical stringed instruments. The AmatiML language defines a series of geometric elements such as points, line segments, and circles; and uses a series of inscribed circles and reverse curves formed by these geometric elements to draw a violin outline (Figure 4).

In the violin design explorer all violin drawing work is handled by the Digital Amati software.

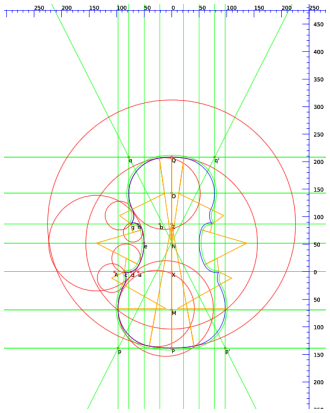

Figure 4: A violin outline drawn by Digital Amati.

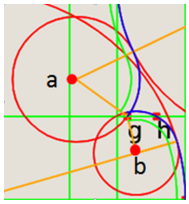

Figure 5: The scope of points.

3.2 Representation

3.2.1 Genetic Representation

In the violin design explorer the genotype representation consists of an array of point locations (details follow below). Each array corresponds to a single violin outline. Each element in the genotype representation array has a scope. For example, consider points a and b that are the centers of two circles (Figure 5). Point g is the intersection of these two circles. Here we can interpret point g as the apex of the violin's upper left sharp corner. As the positions of points a and b change the sharp corner of the violin changes. When the positions of points a and b are far enough apart the intersection (point g) of the two circles disappears. Therefore, point g has a scope attached in order to ensure that the two circles intersect. A scope can be interpreted as a range of values that satisfies certain criteria.

Interestingly, in Digital Amati the scopes for some points can be discontinuous. For example, the scope of point P (see Table 1) is 6-9 and 11-13 written in interval notation as [6,9] and [11,13]. The violin design explorer manages these discontinuities by using five different models or search spaces for violin outlines. Table 1 summarizes the different models and their point scopes used by the violin explorer.

In the following we briefly highlight the function

Table 1: The violin explorer genotype representation models.

Model	Genotype	Scopes
Model 1	[q,O,e,c]	q:[1,8] O:[4,5] e:[6,10] c:[2,6]
Model 2	[q,O,e,c,P]	q:[2,4] O:[3,5] c:[2,6] P:[6,9] e:[8,10]
Model 3	[q,O,e,c,P]	q:[2,4] O:[3,5] c:[2,6] P:[11,13] e:[8,10]
Model 4	[q,O,e,c,P]	q:[5,8] O:[3,5] c:[2,6] P:[6,9] e:[8,10]
Model 5	[q,O,e,c,P]	q:[5,8] O:[3,5] c:[2,6] P:[11,13] e:[8,10]

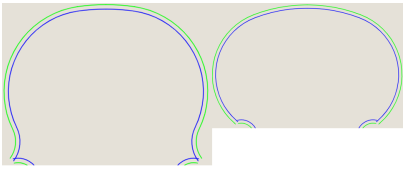

Figure 6: The effect of the point q value.

Figure 7: The effect of the point O value.

of each point in the genotype representation. The value of point q controls the width of the upper bouts of a violin. As the value of point q increases, the upper bouts of the violin design will become wider. As shown in Figure 6, the point q value of the left design is 2 and the right design is 8.

The value of point O controls the smoothness of the upper bouts of a violin. When the value of point O increases, the upper bout of a violin is smoother. For example, there are three upper bouts design in Figure 7. The point O value of the left design is 3. The point O value of the middle design is 5. The point O value of the right design is 6.

The value of point e controls the width of the middle bouts of a violin. As the value of point e increases, the middle bouts of the violin will become narrower. As shown in Figure 8, the point e value of the left design is 6 and the point e value of the right design is 10.

The value of point c controls the shape of the lower bouts. As the value of point c increases, the corner of the violin's lower bouts become much sharper. As shown in Figure 9, the point c value of the left design is 2 and the point c value of the right design is 6.

The value of point P also affects the shape of a violin's lower bouts. As the value of point P increases, the violin's lower bouts become taller and rounder. As shown in Figure 10, the point P value of the left design is 6 and the point P value of the right design is 8.

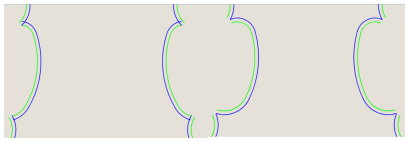

Figure 8: The effect of the point e value.

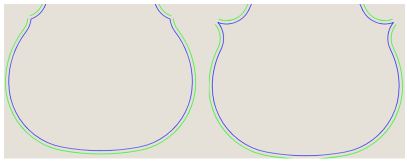

Figure 9: The effect of the point c value.

3.2.2 Embryogeny

An embryogeny is a mapping process from genotype to phenotype (Bentley and Corne, 2002). The embryogeny process of the violin explorer is implemented through the Digital Amati software package. The Digital Amati package uses the AmatiML language to define a series of geometric elements such as points, lines, and circles; and use a series of inscribed circles and reverse curves formed by these geometric elements to draw a violin outline. Each component in the genotype is used to construct part of a particular violin outline as described in in the previous section.

3.2.3 Phenotype Representation

A phenotype representation in the violin design explorer is a violin outline drawn by the Digital Amati software using the Retrofit and Batik libraries (Retrofit 2, 2019; Apache^{TM} Batik SVG Toolkit, 2019), as shown in Figure 11.

These phenotype representations are treated as solutions to violin outline designs and are evaluated by the users of the system.

3.3 The Core Evolutionary Algorithm

The violin design explorer uses a standard evolutionary strategies algorithm (Beyer, 2013; Hansen et al., 2015), as outlined in the following:

Step 1: The violin design explorer is initialized by randomly generating 10 violin outlines. These vi-

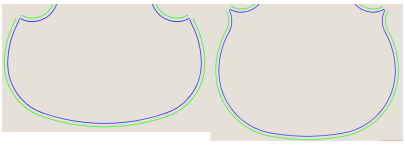

Figure 10: The effect of the point P value.

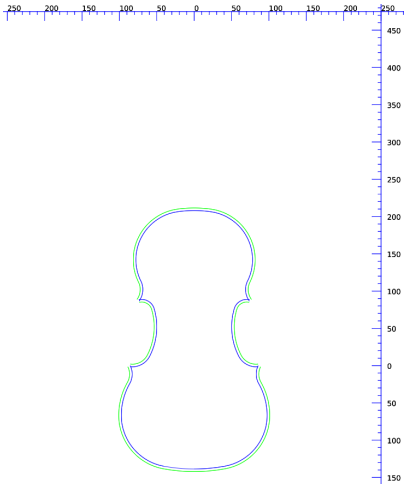

Figure 11: The phenotype representation of the violin explorer.

olin outlines are treated as the original population pool.

Step 2: Users are asked to select two of the most satisfying designs from the population pool.

Step 3: These two designs will become the parent population. Offspring are introduced through recombination. All offspring are placed into the population pool until it reaches the maximum population; in our case 10 violin outlines.

Step 4: Mutation is applied to each single violin outline in the population pool based on values in strategy parameters of the system. The strategy parameters themselves are also mutated within the overall mutation process.

Step 5: Once recombination and mutation have generated enough new individuals to fill the population pool, users will be asked to select two of the most satisfying designs from the population pool. These two designs will be treated as the parent population for the next generation.

Step 6: Repeat Step 3 through Step 5 to generate new generations of violin designs.

This evolutionary process terminates after it reaches a predefined number of generations or when users are satisfied with the current design. In our case

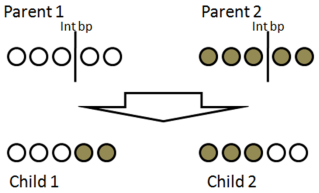

Figure 12: Recombination process.

we limit the number of generations to 50 across all 5 models.

3.3.1 Evolutionary Strategies Scheme

As we have seen, the violin design explorer uses a (μ, λ)-ES scheme to create new individuals in each generation. Here μ is the number of parents and λ is the number of offspring (Beyer, 2013). We chose the values $\mu = 2$ and $\lambda = 10$ according to the recommendations by Hanson *et al.* (Hansen et al., 2015). According to Hanson *et al.* the optimal truncation ratio is $\mu/\lambda = 0.27$ with a corresponding maximized progress per offspring of 0.202. Our choice of the values for $\mu = 2$ and $\lambda = 10$ gets us close to the optimal truncation ratio, keeps things straight forward in terms of computation, and yet still maintain a manageable number of designs for the user to process.

3.3.2 Recombination

Recombination is applied at each generation cycle which recombines parent individuals in different ways to create new offspring. The violin explorer defines a breakpoint integer bp which is randomly generated and designed to determine the position to break a genotype into two pieces. For example, when $bp = 3$ both parents are divided into two pieces from the 3rd position (Figure 12) and the resulting pieces are recombined to form offspring.

According to our overall algorithm the recombination process will repeat until sufficient offspring are generated.

3.3.3 Mutation

Mutation plays an important role in evolutionary strategy in order to enhance the overall search for possible solutions (Bentley, 1999b). The mutation process in the violin design explorer involves making small changes to each genotype element. The value of change is determined by the mutation strength parameter. The mutation strength parameter itself is an array that is randomly generated when the original population is initialized. This array has the same length as

the genotype representation array and each array element within the mutation strength array corresponds to a position in the genotype representation. The mutation strength of each element is used to produce random changes in the corresponding genotype element according to a Gaussian distribution with a mean of zero. During the evolutionary process the mutation strength is reduced in order to foster a better convergence behavior.

3.3.4 Solution Evaluation

In our currently implementation violin outline design is considered an aesthetic problem. It is almost an impossible task to establish a fitness function to capture aesthetics. Many factors such as a user's age and background, as well as current fashion trends may become key factors influencing violin design. Therefore the violin design explorer uses human interaction processes instead of using a fitness function to evaluate violin outline designs.

4 EXPERIMENT

The primary objective of our experiment is to validate the violin design explorer. Four distinct groups of participants were asked to design their favorite violin outline. We were working under the hypothesis that if we see distinct design patterns in the four participant groups then our violin design explorer works as envisioned. The design patterns that emerged within these four participant groups during the experiment seem to validate that our design tool works as we envisioned.

4.1 Experiment Setup

Our experiment population consisted of 100 volunteers of different ages and backgrounds. The volunteers were asked two questions to determine their experimental grouping:

- Which age group are you in? "18 to 40" or "older than 40"?
- Do you have an artistic background such as painting or playing musical instruments?

Based on the participants' answers to these two questions they were divided into four experimental groups each with 25 participants as shown in Table 2.

During the experiment volunteers used the violin design explorer to design their favorite violin outlines. Our analysis of the data often makes reference to the "classic" or "original" violin design. By this we mean a design as established by the old Italian masters in the 1600's. Figure 11 shows an image of this reference design.

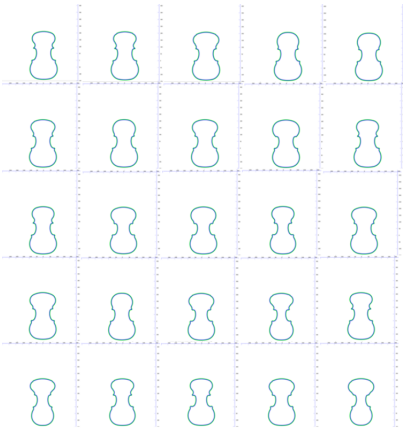

Figure 13: The designs of group 1.

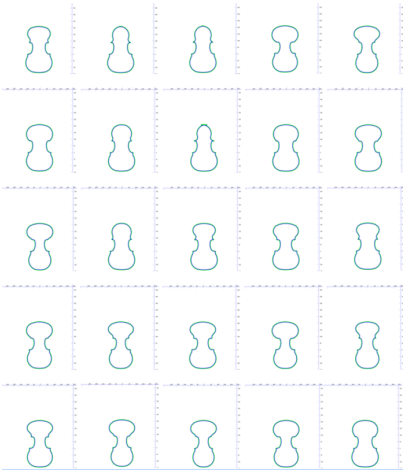

Figure 14: The designs of group 2.

4.2 Results

The volunteers in group 1 were between 18 and 40 years old and had an artistic background. The 25 violin designs of that group are shown in Figure 13.

The volunteers in group 2 were between 18 and 40 years old and had no artistic background. The 25 violin designs in that group are shown in Figure 14.

The volunteers in group 3 were older than 40 years and had an artistic background. The 25 violin designs in that group are shown in Figure 15.

The volunteers in group 4 were older than 40 years and had no artistic background. The 25 violin designs in that group are shown in Figure 16.

We averaged the genotype representations within

Table 2: Four experimental groups.

Group 1	Participants are between the ages of 18 to 40 and have an artistic background.
Group 2	Participants are between the ages of 18 to 40 but do not have any artistic background.
Group 3	Participants are older than 40 and have an artistic background.
Group 4	Participants are older than 40 but do not have any artistic background.

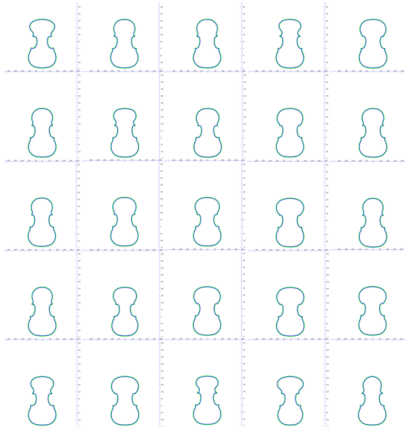

Figure 15: The designs of group 3.

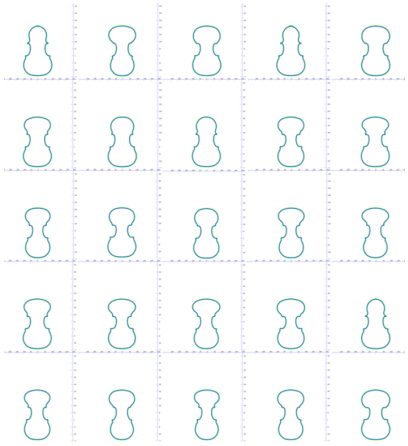

Figure 16: The designs of group 4.

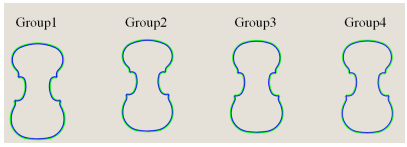

Figure 17: The average designs for the four groups.

each experimental group in order to compute the "average" design for each group and these designs are shown in Figure 17. A visual inspection of these average designs reveals that the average design of group 3 (older than 40 with an artistic background) is most similar to the classic violin design of the old Italian masters in Figure 11.

Table 3: The design variance in each group.

Group	Design Variance
1	1.93
2	2.29
3	1.41
4	1.94

Table 4: Design deviation from the classic design.

Group	Design Deviation
1	1.94
2	2.37
3	1.35
4	2.20

An interesting metric to consider is the design variance in each group. The design variance is computed by computing the parameter variances in each group and then averaging those variances for each group. The design variance for each group is shown in Table 3. We observe that groups 1 and 3 have the lowest design variances indicating that users with an artistic background have similar notions of optimal designs. The lowest design variance is in group 3 indicating the older users with an artistic background tend to have very similar ideas about violin designs. The highest design variance is in group 2 indicating that younger users without any artistic background are the most likely to experiment with violin designs.

The design deviation from the classic design in Figure 11 in each group can also be considered as a metric. We computed the design deviation by computing the standard deviation of the designs in each group from the classic design. Table 4 shows the design deviations in each group. Perhaps not unexpected we find that groups 1 and 3 have the lowest deviation from the classic design. We also find that group 3 (older users with an artistic background) have the overall lowest deviation from the classic design validating our casual observation from earlier that the designs in this group are most similar to the classic violin design. Finally, we find that users in group 2 (younger users without an artistic background) have the highest design deviation further validating the fact that users in this group are the most likely to experiment with violin designs.

Given these not unexpected design patterns within

each of the groups we feel that this validates the workings of our violin design explorer.

5 CONCLUSIONS AND FURTHER WORK

Here we discussed an evolutionary design tool, the violin design explorer, for violin outline design. With this system, users without any formal design background can create their favorite violin outline through a set of simple choices. Our design tool employs an evolution strategies algorithm and can be viewed as a creative evolutionary system. The genotype representation of the system directly supports violin outline manifestation through the Digital Amati software library. In order to validate the design tool we conducted an anonymous experiment. Four groups of participants were asked to design their favorite violin outline. Design patterns that emerged within each of these four participant groups during the experiment seem to validate that our design tool works as we envisioned.

Currently we partition the violin outline search space into five parts via the five models driven by parameter scopes. We would like to explore other ways of structuring the search space and therefore provide potential solutions to the users in different ways. Furthermore, in our current implementation we use the default violin template and its associated parameters provided in Digital Amati as a starting point for the evolution of new outlines. This completely constrains the user to stay within the somewhat classical look of a violin with upper, lower, and "C" bouts, *etc*. It would be interesting to experiment with different kinds of templates as starting points providing the user with a larger range of possible outline designs.

Here we only explored one aspect of violin design – the aesthetics of the violin outline. Our goal is to provide comprehensive interactive design tool for a fully functional violin. In order to accomplish this we will need to address body cavity design as well as neck, sound hole and bridge designs. This means we have to develop models for these violin parts in order to provide constrained parameter spaces that make sense for violin design similar to the scopes we saw in the outline design. A good starting point for developing these kind of models is the research by Carleen Hutchins (Hutchins, 1981).

Ultimately we would like to be able to generate designs that are sufficiently detailed so that they can be directly submitted to a CNC (Xu and Newman, 2006) or other milling machine for construction.

REFERENCES

Algorithmic Art Assembly (2019). https://aaassembly.org.

Apache^{TM} Batik SVG Toolkit (2019). https://xmlgraphics.apache.org/batik.

Bäck, T., Fogel, D. B., and Michalewicz, Z., editors (2018). *Evolutionary computation 1: Basic algorithms and operators*. CRC Press.

Bentley, P. (1999a). *Evolutionary design by computers*. Morgan Kaufmann.

Bentley, P. (1999b). An introduction to evolutionary design by computers. *Evolutionary design by computers*, pages 1–73.

Bentley, P. J. and Corne, D. W. (2002). An introduction to creative evolutionary systems. In *Creative evolutionary systems*, pages 1–75. Elsevier.

Beyer, H.-G. (2013). *The theory of evolution strategies*. Springer Science & Business Media.

Digital Amati (2018). http://www.digitalamati.org.

Frazer, J. (2002). Creative design and the generative evolutionary paradigm. In *Creative evolutionary systems*, pages 253–274. Elsevier.

Hansen, N., Arnold, D. V., and Auger, A. (2015). Evolution strategies. In *Springer handbook of computational intelligence*, pages 871–898. Springer.

Hutchins, C. M. (1981). The acoustics of violin plates. *Scientific American*, 245(4):170–187.

Kramer, O. (2017). *Genetic algorithm essentials*, volume 679. Springer.

MacCallum, R. M., Mauch, M., Burt, A., and Leroi, A. M. (2012). Evolution of music by public choice. *Proceedings of the National Academy of Sciences*, 109(30):12081–12086.

Menges, A. and Ahlquist, S., editors (2011). *Computational design thinking: computation design thinking*. John Wiley & Sons.

Retrofit 2 (2019). https://github.com/square/retrofit.

Rooke, S. (2002). Eons of genetically evolved algorithmic images. In *Creative evolutionary systems*, pages 339–365. Elsevier.

Xu, X. W. and Newman, S. T. (2006). Making cnc machine tools more open, interoperable and intelligent—a review of the technologies. *Computers in Industry*, 57(2):141–152.