A Genetic Algorithm Approach to Collaborative Music Creation on a Multi-Touch Table

Niklas Klügel
Technische Universität München,
Garching, Germany
kluegel@in.tum.de

Andreas Lindström
KTH Royal Institute of Technology,
Kista, Sweden
andlinds@kth.se

Georg Groh
Technische Universität München,
Garching, Germany
grohg@in.tum.de

ABSTRACT

Multi-touch interfaces provide new opportunities for collaborative music composing. In this report, an approach using genetic algorithms to evolve musical beats in a collaborative setting is presented. A prototype using a multi-touch interface is developed and evaluated.

1. INTRODUCTION

Advances in computer technology in the last decades have provided many new tools for musical tasks and have in many ways changed the way musicians and composers approach music and composing [1]. With cheap personal computers and music software readily available, music making and recording is no longer confined to professional recording studios. There is a wide array of DAW [1] applications on the market today that allow users to produce and record music at home, without requiring expensive equipment. However, most of these are single user applications that still require extensive knowledge on behalf of the user in order to fully facilitate the potential of the program. A typical work-flow when using these applications in a setting with more than one composer is to make local changes to the project files and then send files back and forth between the collaborating composers. This way of working puts limitations on the collaboration process and does not allow parallel inputs from the different composers. It is interesting to study how the music creation process can be made more collaborative, and also how to make it more accessible to users without expert knowledge in techniques for music production. Given the collaborative nature of musical performance and innovation, new technologies and interfaces in collaborative computer systems can be used to explore the possibilities of music composing and performance in a group setting. Naturally, some problems arise when facing the task of composing music together with others. How should the group members' different backgrounds, intentions and musical ideas be mapped onto the resulting singular composition? This is insofar a key issue as tasks involving group creativity benefit from heterogeneous groups [2] [3, p. 450-452], [4]. This also means that novices need to be integrated in the collaborative effort. In [5] it is argued that for social creativity the heterogeneity of individuals, the overall diversities in knowledge, experience and expertise are the key elements to foster creativity and therefore must be integrated. Depending on the definition, creativity relates to both an individual [3, p. 137-144, p. 450-452] - human cognition offers creative capacity as an essential property- and a social context - the assessment of value of an original idea for a certain domain by means of social and cultural processes- [6],[3, p. 313-335]. Especially social creativity has been shown to provide creative and sustainable solutions [5, 2], e.g. in the context of small world networks and its relationship to key innovations in art and science [7]. This effect is further amplified by the phenomenon of *Group Flow* [8, p. 158]. Group Flow is important as it stimulates an implicit learning process and therefore creates a tri-directional link between individual, group and the music created. Besides the feeling of personal engagement leading to the empathic involvement with music [9], it is also bound to the feeling of *social presence*. The experience of Group Flow has been already shown to take place in Computer Supported Collaborative Music making (CSCM) [10]. Furthermore, the empirical study by MacDonald [11] has revealed that groups that reported the experience of Group Flow created compositions that were rated more valuable (by a third party) than groups that did not experience it. In this way Group Flow is a motivator and means for the group to innovate in a creative task. The engagement by intrinsic motivation has furthermore the effect of supporting the learning process for musical expression, thus the mediating interaction with the shared CSCM environment and finally the social interaction with peer members. Thus, we see the integration of these social effects that take place in creative collaboration as highly beneficial for the task of creating music.

In this context, an application that supports social creativity can be seen to have a mediating role, namely to consolidate different intents by the collaborators. Furthermore, especially for novices, such a mediating support has to facilitate the expression of intents by augmenting the users' skills. This can be provided by either making use of compositional rules or generally, by guessing the users' intent function (e.g. give compositional recommendations or generate specific derivations of the shared composition).

In this contribution we will motivate and present an ap-

[1] Digital Audio Workstation

proach for co-located group composition using genetic algorithms. Here, such algorithms will be used to establish a mediating functionality that will adapt to the users' modifications of the shared composition in order to converge towards a collectively agreeable goal.

However, the main difficulty in solely evaluating the additional benefit of this mediating role of such an approach is to avoid or at least minimize the bias that the expressivity of direct controls themselves impose upon the application - such as familiar control metaphors for controlling musical events or overtly self-evident gestures encoding musical event mappings. This may otherwise skew results by favouring proficient users for this use case or simplify a compositional process. The reasoning is that musical novices would benefit the most from a mediating functionality given that they presumably have no prior knowledge on how to approach the task of music composition. Hence, they may not be able to *musically* understand the effect of their interactions with the application in view of an intended goal. Thus, regarding the user interface and interaction metaphors, we decidedly put the users into a position where they have to rely on the mediating functionality, therefore enforcing a loss of control at the expense of *perceived* expressivity. We will elaborate this in more detail in section 4.

To summarize, the objective of this contribution is to evaluate, whether active and adaptive mediation, specifically using Genetic Programming, adds additional value for supporting the creative task of collaboratively composing music. For this we will devise a system that is capable of creating a variety of short musical forms ("beats") that include bass and melody lines with accompaniment including chord progressions and drum patterns. We see the music composition system underlying the application as capable to create a realistic subset of music forms common in modern popular music.

Finally, for the prototypical implementation we will utilize a multi-touch table for interaction to further offer means that facilitate social communication protocols [12].

The outline of this paper is as follows: first we will review related work, especially with regard to the application of genetic algorithms. We will then motivate the concept and elaborate the prototypical implementation. This is followed by the presentation of a user study that aimed to evaluate the prototype. The contribution concludes with the discussion of the results thereof and gives starting points for future research. We do not want to evaluate for expressivity but the additional benefit of a mediating role, in our case the GA specifically.

2. RELATED WORK

Tabletop interfaces represent one end of the spectrum of synchronicity where all group interaction is carried out physically and locally using a single shared workspace. Several applications either in the scientific or commercial field exist that facilitate the collaborative composition of music following various metaphors using a multi-touch table. ReacTable [13], for example, allows the creation of looped phrases using step-sequencers that modulate audio generators (modular synthesizer). A previous contribution [14] allowed collaboratively arranging sequences of note events that have been entered using the piano roll notation (non-linear multi-track sequencer). Xenakis, on the other hand, follows a probabilistic approach to compose musical events (algorithmic composition) and Touchtr4ck [15] allows to mix and manipulate recorded sound material (loop/phrase sampling). However, these applications do not *actively* mediate the collaborative composition process by taking part in the decision making.

There have been many attempts to mediate or automate music composition tasks in order to have computers aid composing music that is pleasing to human ears. Different approaches in the general field of algorithmic composition are described in [16]. A prominent one is to use genetic algorithms to generate music. Such algorithms draw inspiration from the evolution in nature in order to solve computational problems and were invented by John Holland in the 1960s [16, p. 2]. No domain specific knowledge about the problem is necessary and they can be applied to a vast array of optimization problems. Given a well-defined problem, candidate solutions to the problem are encoded as chromosomes, usually as bit-strings. In order to evaluate the fitness of a chromosome, a fitness function is used that assigns a score to a given chromosome. In a simple form, a genetic algorithm starts by randomly generating a starting population of chromosomes and assigning scores to each one using the fitness function. New chromosomes are then spawned from the initial population using crossover and mutation operations. Crossover between two "parent" chromosomes is done by concatenating divisions of the bit-strings of the chromosomes in order to create a new one. Mutation of the resulting "child" chromosome is performed by changing the value at each bit position with a small probability. This is done to ensure variance in the population over time. By giving highly fit individuals a bigger probability of being chosen for reproduction by crossover, the aim is to obtain high quality solutions after several generations of evolution.

When applying genetic algorithms in a musical composition context, the search problem can be defined as "from the space of all possible compositions, find one that sounds good" [17]. Two obstacles need to be overcome here in order to be able to apply genetic algorithms: first, an encoding of the musical structure must be defined so that chromosomes can be created and evolved. Second, a fitness function must be defined that can rate the quality musical output. Of these two obstacles, the second one is definitely the hardest one to overcome due to the difficulty of objectively rating the quality of music by automatic means. Nevertheless, quite a few applications have emerged using this approach. An overview of some of these can be found in [18, ch. 7.4]. Many of the applications described usually perform a specific task (such as harmonizing a melody line) with comparison to human compositions as basis for the fitness function. Others use rules grounded in musical theory to rate the chromosomes.

Another approach is to have a human-based fitness function by letting a human evaluator rate the output of the program. This is used for example in the application GenJAM [19], which can generate jazz solos over a given chord progression. This circumvents the difficult problem of algorithmically rating musical quality. However, as discussed in [18, ch. 7.4.2], when using a human fitness function the output is dependent on the evaluating person who might be exposed to fatigue after evaluating large numbers of musical examples. This could lead to non-consequent rating from the evaluator which in turn results in a poor performance of the algorithm. Therefore such issues have to be remedied for a practical real-time application of a genetic algorithm approach. Furthermore, it is necessary that users have direct influence on the behaviour of the algorithm such that it can can adapt to new composition objectives (e.g. an abrupt change in the group's stylistic preference).

3. CONCEPT

There are some studies and applications that explore the possibility of using genetic algorithms in an interactive setting. One example is described in [20], where a genetic algorithm guided by the position of two users in space generates layered melodies. However, this application lacks a fitness function and only generates new chromosomes based on the users' actions. Studies and applications of this kind are interesting for several reasons. It might be possible to overcome the previously mentioned fatigue problems when using human evaluators by letting the evaluators themselves interact with and affect the chromosomes linked to the music creation. This can be seen as guided mutation operations, as opposed to the random mutations performed by the program. In fact, it is argued in [21, p. 2-3] that development of musical structure can be seen as directed mutation; new musical ideas are generated by mutating other musical ideas.

Such an approach can be implemented using a turn-based composition strategy that comprises of repeated turns of active modification of the composition by the users (e.g. changing the scale or patterns of the shared composition) and evaluation by the users after which the genetic algorithm performs one cycle and then presents the users with an updated version of their previous composition. The evaluation can be implemented using a voting scheme, thus it corresponds directly to a fitness function in generic genetic algorithms.

Splitting the composition process into several cycles of phases furthermore has the not only the advantage of users being able to directly alter the composition in real-time to steer the genetic algorithm but also to hear the results of their interactions immediately (first phase). This immediacy is necessary for any musical engagement [22].

To summarize, our approach implies a direct analogy between the human composition process and the compositional process performed by a genetic algorithm. This also serves as a motivation for the choice of using genetic algorithms with additional guided mutation from the users as an additional element, since the user input can directly be applied to the algorithm at run-time. Therefore, the approach involves the users in the music generation instead of just letting them passively listen and rate the automated music creation, in order to achieve faster convergence of the genetic search and to solve the previously discussed listening fatigue problem. In this way, the mediating role of this creativity support system is in some way to guess the collaborator's musical intent and to converge towards it.

In order to investigate how genetic algorithms can support and enhance the creativity of group composition, an application was developed that realizes the concept of applying user input as directed mutation to a genetic algorithm that generates musical beats. In order to limit the musical space that the users can explore, the application was designed to produce musical beats with a length of four bars. The beats have a granularity corresponding to 16th notes, not unlike that of a 16th-note step sequencer. This approach of course further limits the musical abilities of the application. The application can only produce note lengths that are integer multiples of a 16-th note, which means that for instance triplets can not be achieved. However, for the purpose of this experiment, this simple approach should be sufficient while it can be safely assumed that the application can be generalized to contain more complex rhythmic combinations and to support music that is not beat based, if the basic concept should prove to be successful.

A beat in the application consists of a drum part, a bass line, a melody and a chord sequence that defines the harmonic environment for the bass and melody parts. The drum part in turn consists of hi-hat, bass, and snare drum rhythms that can vary individually over the four bars. The bass and melody are monophonic with notes extracted from the current chord. Each chord consists of four notes that are extracted from a global seven note scale. One chord is applied to each bar, which means that the beat can be seen as a chord sequence of four chords. Apart from controlling the root note of each chord, the users can also control the specific voicing used for each chord as well as the rhythm of the chord playback. Other global parameters apart from the scale used are the tempo for the beat and a binary shuffle switch that applies to all instruments on a 16-th note level. By interacting with the interface of the application, the users can change the parameters of each part individually. An important question in this context is how to map the settings of the interface controls to patterns (rhythm and pitch) for each of the instruments. Also, in order to apply a genetic algorithm to the beat generation, an encoding is needed to represent different beats.

3.1 Pattern Mappings

Regarding the mapping from user input to musical patterns, an initial approach was to allow all possible combinations of beats by representing the rhythm of an instrument by four 16-bit numbers (one for each bar) where the value of the bit in position i decided if the i:th 16-note in that bar should be a hit (1), or a rest (0). By setting the range of the user controls for the rhythm of that instrument in that bar to the interval $[0, 2^{16} - 1]$, all possible 16-th note rhythms cab be achieved. For each note in the rhythms of the melodic instruments (bass and melody), all scale

Instrument / Parameter	# patterns
Hi-Hat	8
Snare drum	4
Kick drum	4
Chord root note	7
Chord voicing	8
Chord rhythm	4
Melody	16
Bass line	16
Scale	8

Table 1. Patterns used for the various instrument types or compositional parameters and their respective number of alternate instances

pitches would be available. However, this approach turned out to be problematic for two main reasons. First, the encoding of a 4 bar beat would need more than 300 bits for its representation which might lead to very long convergence times, especially in an interactive setting with human evaluators. Second, a large subset of the possible beats would probably be rendered non-musical by the users. Thus, this approach needed some modification. To reduce the search space and to render more musical beats, predefined patterns were introduced to each instrument. The rhythmic patterns for each instrument still used the bit-extraction technique described above, but now only a small subset of integers in the interval could be used (between 4-16 rhythmic patterns per instrument). For the melodic instruments, common arpeggiator patterns [2] were used. Eight different scales could be chosen as the harmonic environment for each beat, ranging from common scales such as Ionian (major) and minor pentatonic to more exotic scales such as diminished and whole-tone scales. The tempo, expressed in beats per minute, could be set to any integer in the interval $[60, 300]$ and the shuffle switch was indirectly defined as a modulo-2 operation of the sum of the drum parameters, resulting in a shuffled beat if they add up to an odd number. To imitate the shuffling of a human drummer, the shuffle ratio was decided by the tempo of the beat as described in [23]. The advantages of this pattern-based approach include more musically coherent beats and radically shorter beat encodings. The patterns can be ordered with respect to their rhythmic complexity [24], which allows for more musically sensible mutations. An obvious disadvantage is that the musical space is totally defined by the pre-defined patterns which reduces the generality of the application and the musical freedom of the users.

3.2 BEAT ENCODING

Using the pattern based approach described in the previous section, the chromosome of a beat could be encoded as a string of concatenated integers, with each integer giving the pattern index of a specific instrument. Thus, the set of integers specifying the patterns to apply for each instrument corresponds to a gene in the chromosome. The pattern encoding for each instrument is given in table 1. As an example, the gene integer that corresponds to the melody pattern in a bar takes a value in the interval $[1, 16]$. A beat contains four bars in which the patterns can vary independently (with the exception of the scale, which is global for all four bars). Disregarding global tempo and pitch parameters, the cardinality $|S|$ of the set of unique beats that the application can produce is

$$|S| = 8 \cdot (8 \cdot 4 \cdot 4 \cdot 7 \cdot 8 \cdot 4 \cdot 16 \cdot 16)^4 \approx 2.3 \cdot 10^{28}$$

Half of these beats will be shuffled beats. The tempo and pitch of each beat can be altered for further variation. This result ensures that the musical space available to the users to explore is still very large, despite using the limiting pattern approach.

3.3 Fitness Function

With the encoding of beats defined, all that remains is to define a fitness function that grades the quality of a beat. Due to the difficulty of algorithmically grading musical quality (see section 2) as well as due to having the goal of a highly interactive setting, the choice was made to implement a voting system for the fitness function. After modifying a beat, each user can grade the current beat by supplying a decimal value in the interval $[0, 1]$, where high scores correspond to high musical quality. The average of all votes is then fed to the genetic algorithm and used as fitness. By using this approach, the aim is to map the group's creative will into a singular value, thus allowing the beats to converge to the group's preference over time.

4. INTERFACE & PROTOTYPE

Regarding the emphasis on the mediating functionality presented in the introduction, we decided to use abstract shapes to control the various parameters inherent in the composition which deliberately do not suggest concrete musical meaning. Thus, for a user not knowing the underlying pattern based system, there is no immediate connection between interaction and the *exact* musical result. However, all parameters of the shared composition are represented with a direct one-to-one mapping: each individual instrument or parameter was represented as a shape, changing their size or three dimensional rotation (using multi-touch gestures) corresponded to the pattern selected for that particular instrument of parameter. Furthermore, besides the scale factor of a shape the amount of absolute rotation was indicated by illustrating arcs surrounding it, giving visual feedback that the interaction had been registered by the application. In this way the state of the application is always visible and not artificially obfuscated. Instead, it is expressed in terms that are not making use of established musical vocabulary and that only allow a rough guess of what the parameter values represented by the position of a shape are (e.g. the visualization of a rotated shape does not indicate exactly which pattern has been chosen).

A screen-shot of the interface of the prototype can be seen in figure 1. The four blue and yellow shapes correspond to chord and drum parameters, respectively. The grey pyramid shape at the top of the screen controls the melody and

[2] Ascending, descending, ascending / descending and alternating.

the cube to its right controls the bass parameters. The two red shapes to the right control the global scale, pitch and tempo. The three voting panels can be seen, as well as a centered circular progress bar that completes one lap in the four measures of the beat.

When the application was started, an initial population of six beats was randomly generated and given low random fitness scores. One of these randomly generated beats was chosen as the initial beat for the users to modify. The shapes were automatically updated (resized and rotated) to represent the new state of the application corresponding to the randomly generated patterns. This beat could now be heard by the users and they could modify it by moving and rotating the shapes. The users were free to vote at any time, and when all three users had supplied a vote, the average score was supplied to the genetic algorithm. Two beats were then selected for mating from the entire population. A new offspring was created using crossover of the chromosomes and the resulting child subjected to random mutations (usually by adding or subtracting a small number to the pattern index) and the child beat was given to the users for modifications. The chance for a beat of being selected for mating was proportional to the fitness rating of the beat. To further speed up convergence, we performed additional modifications to the genetic algorithm, namely the reduction of the population size to at most ten chromosomes and a reduction of the fitness score of old beats after each new generation to premier new beats added to the population.

We used a 55 inch multi-touch table to interface with our application. The number of parameters available to modify the beats as well as the dimensions of the multi-touch board made the prototype suitable to use in groups of three, with the intention that one person controls the drum parameters, a second the chord parameters and the third one the melody and bass. The control over the global parameters scale, tempo and pitch could be shared between the users. This can be seen as an analogy to a traditional band environment. However, for the evaluation we explicitly told the participants that they are free to change position as they wish during the testing.

The interface and genetic algorithms were implemented as a Scala program, sending OSC messages to a Pure Data patch that realized the patterns. From this patch, MIDI messages were sent to a running instance of Ableton Live [3] which played back the beat using stock instruments.

5. EVALUATION

18 test subjects were divided into groups of three for the evaluation. The test subjects were engineering students of which 61% played an instrument or sang in their free time. 56% had received musical training in some form. The test subjects' amount of previous experience with multi-touch interfaces varied substantially. The evaluation was divided into two consecutive parts: first, after a brief explanation of the functionality of the application, the test subjects were allowed to create beats with the application for 25 minutes. During this session, the users were allowed to experiment and use the voting system freely. They were instructed that they could use any means of collaboration available to them, verbal as well as non-verbal. They were also told that they could move around the multi-touch table to rotate the control of the beat parameters within the group. In the second part of the evaluation, the test subjects answered a computer-mediated questionnaire containing questions regarding collaboration aspects, expressive capabilities of the prototype and the user experience. Most questions used a 5-level Likert scale.

[3] https://www.ableton.com/

6. RESULTS

As expected, the application was rated low to mediocre regarding control and expressivity. Only 45% of the participants agreed that they found it easy to control specific parts of the composition, 67% to control their parts of the music such as instruments or and only 50% control over the music as a whole. Furthermore, they felt that they could not express their creativity properly (44% agreement), but express themselves musically (63% agreement). Still, and in accordance with our assumption in the introduction that unacquainted users would still be able to make out the effect of their interactions, 71% agreed that their actions affected the music. Additionally, concerning the potential for exploring the musical possibilities, 84% of users thought there was a lot to discover with the application.

Regarding collaboration, 71% saw their collaborators as enrichment for their creative endeavours, 67% agreed that the application supported their collaboration and furthermore 72% felt that the application helped them being creative as a group.

With respect to the most important part of the evaluation, the mediating functionality was voted favourably. The average number of voting rounds was 11, indicating that this part of the system was frequently used during the comparatively short time for evaluation. In accordance to this, only 24% of the participants thought of the voting system to be unnecessary. 83% of users stated that the music got better over time, and 66% stated that after completing a single voting round, the musical quality of the new offspring beat that was presented was higher than the parent beat while still being recognizable (70%). Half of the users said they were helped to new musical ideas and also stated that their group was mostly in agreement about the voting (73%).

Given the low ratings in regards to control, it is surprising that only 38% percent of the users stated that they got frustrated using the application. Accordingly to the favourable reviews of the collaborative and mediating aspects 80% of the participants felt that it was fun to make music with the application and, equally, that they enjoyed using it. Especially since the results were mixed with respect to liking the music that has been produced.

Regarding verbal communication, it was noted that groups where at least one member did not know the other members used a lot less verbal communication in the form of directives or discussions as opposed to groups where all members knew each other from before. Furthermore, some groups frequently changed positions around the table, which

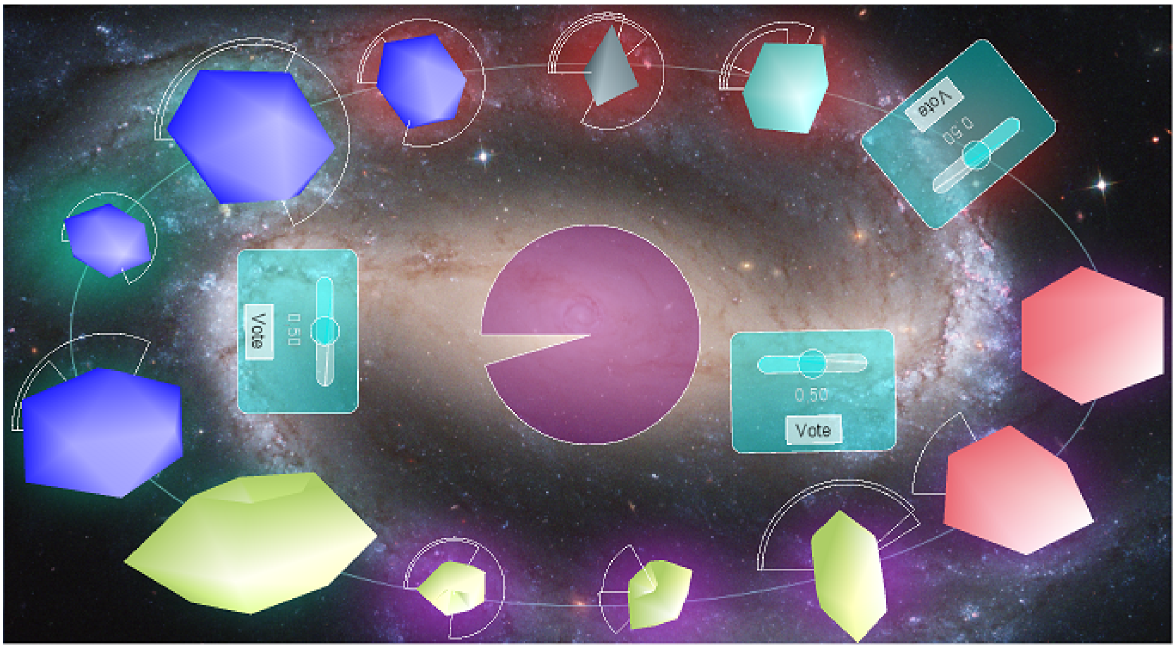

Figure 1. A screenshot showing the interface of the prototypical implementation

may hint at their more democratic approach to using the application.

7. DISCUSSION & CONCLUSION

The problems revealed in the evaluation are related to the usability of the prototype. The difficulty for users to understand the functionality of the controls and inability to modify specific parts of the beat to their liking can be traced to the abstract layout of the interface. Given this contrast to the fact that the majority of users thought the quality of the resulting beat improved (after a single or multiple voting rounds) compared to the original beat indicates that the genetic algorithm approach may indeed fulfil its role in supporting the users in their creative endeavours. Furthermore, the aim of having software inspiring new musical ideas is rather ambitious and the fact that half of the users got new ideas from the genetic approach can therefore be seen as positive. However this result should be regarded as preliminary and therefore taken with caution, since additional experiments are necessary to differentiate between beneficial social effects of the collaboration, the efficiency of the genetic approach, or possible placebo effects. Therefore we argue that a feasible approach would be to evaluate the application with two modifications: completely randomized offspring and offspring that are not altered at all. With these results it should be possible to remove the bias from the evaluations. We credit the mixed results regarding the produced music to both the instrumentation (stock synthesizer patches) and the use of patterns. As the space of possible compositions is still confined to these and that some users may have preferred other musical styles.

To conclude, we nevertheless see these results as satisfactory, especially regarding the favourable feedback to the collaborative aspects in general, indicating that there is additional value in supporting collaborative creativity actively by taking part of the decision process is a viable option. For more complete applications, however, the aim should be not only to add expressive or musically meaningful controls for the composition but also to the synthesis of sound for shaping timbre.

The source code for our application and the related libraries are available online [4] .

8. REFERENCES

[1] P. Ramshaw, “Is music production now a composition process?” in *Proc. Int. Conf. First Annual Conference on the Art of Record Production*, London, UK, 2006.

[2] G. Fischer, “Distributed intelligence: Extending the power of the unaided, individual human mind,” *Proc. Work. Conf. Adv. Vis. Interfaces*.

[3] R. J. Sternberg, *Handbook of creativity*, ser. Perspectives on individual differences, R. J. Sternberg, Ed. Cambridge University Press.

[4] G. Morgan, “Paradigms, Metaphors, and Puzzle Solving in Organization Theory,” *Adm. Sci. Q.*, no. 4, p. 605.

[5] G. Fischer, “Distances and diversity: sources for social creativity,” in *Proc. 5th Conf. Creat. Cogn.* ACM, pp. 128–136.

[6] M. Csikszentmihalyi, *Creativity : Flow and the Psychology of Discovery and Invention*. Perennial, Jun.

[7] B. Uzzi and J. Spiro, “Collaboration and Creativity: The Small World Problem,” *Am. J. Sociol.*, no. 2, pp. 447–504, Sep.

[4] `https://github.com/lodsb/UltraCom`

[8] R. K. Sawyer, "Group creativity: musical performance and collaboration," *Psychol. Music*, no. 2, pp. 148–165, Apr.

[9] L. Nijs, *The musical instrument of natural extension of the musician*. Witwatersrand University Press.

[10] B. Swift and A. Riddell, "Engagement Networks in Social Music-making," *Interfaces (Providence).*, 2004.

[11] R. MacDonald, "Creativity and flow in musical composition: an empirical investigation," *Psychol. Music*, no. 3, pp. 292–306, Jul.

[12] D. Wigdor and D. Wixon, *Brave NUI World: Designing Natural User Interfaces for Touch and Gesture*. Morgan Kaufmann.

[13] S. Jordà, M. Kaltenbrunner, G. Geiger, and M. Alonso, "The reacTable: a tangible tabletop musical instrument and collaborative workbench," in *SIGGRAPH '06*. New York, USA: ACM, 2006, p. 91.

[14] N. Klügel, M. R. Frieß, G. Groh, and F. Echtler, "An approach to collaborative music composition," in *Proc. of Int. Conf. New Interfaces for Musical Expression*, 2011, pp. 32–35.

[15] A. Xambó, R. Laney, and C. Dobbyn, "Touchtr4ck: democratic collaborative music," in *Proceedings of the fifth international conference on Tangible, embedded, and embodied interaction*. ACM, 2011, pp. 309–312.

[16] M. Mitchell, *An introduction to genetic algorithms*. MIT press, 1998.

[17] B. Jacob, "Composing with genetic algorithms," 1995.

[18] G. Nierhaus, *Algorithmic composition: paradigms of automated music generation*. Springer, 2009.

[19] J. Biles, "Genjam: A genetic algorithm for generating jazz solos," in *Proceedings of the International Computer Music Conference*. INTERNATIONAL COMPUTER MUSIC ACCOCIATION, 1994, pp. 131–131.

[20] S. Fels and J. Manzolli, "Interactive, evolutionary textured sound composition," in *Multimedia 2001*. Springer, 2002, pp. 153–164.

[21] A. Gartland-Jones, "Can a genetic algorithm think like a composer?" in *Generative Art*, 2002.

[22] T. Blaine and S. Fels, "Contexts of collaborative musical experiences," in *Proceedings of the 2003 conference on New interfaces for musical expression*. National University of Singapore, 2003, pp. 129–134.

[23] A. Friberg and A. Sundström, "Swing ratios and ensemble timing in jazz performance: Evidence for a common rhythmic pattern," *Music Perception*, vol. 19, no. 3, pp. 333–349, 2002.

[24] E. Thul, "Measuring the complexity of musical rhythm," Ph.D. dissertation, McGill University, 2008.